MODEL-BASED REINFORCEMENT LEARNING FOR ONLINE APPROXIMATE OPTIMAL CONTROL

By

RUSHIKESH LAMBODAR KAMALAPURKAR

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

2014

To my parents Arun and Sarita Kamalapurkar for their invaluable support

ACKNOWLEDGMENTS

I would like to express sincere gratitude towards Dr. Warren E. Dixon, whose constant encouragement and support have been instrumental in my academic success. As my academic advisor, he has provided me with valuable advice regarding research. As a mentor, he has played a central role in preparing me for my academic career by inspiring me to do independent research, providing me valuable insights into the nitty-gritties of an academic career, and helping me hone my grant writing skills. I would also like to extend my gratitude towards my committee members Dr. Prabir Barooah, Dr. Mrinal Kumar, Dr. Anil Rao, and Dr. Sean Meyn, and my professors Dr. Paul Robinson and Dr. Michael Jury for their time, the valuable recommendations they provided, and for being excellent teaches from whom I have drawn a lot of knowledge and inspiration. I would also like to thank my colleagues at the University of Florida Nonlinear Controls and Robotics laboratory for countless fruitful discussions that have helped shape the ideas in this dissertation. I acknowledge that this dissertation would not have been possible without the support and encouragement provided by my family and my friends and without the financial support provided by the National Science Foundation and the Office of Naval Research.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table page

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACI	actor-critic-identifier
ADP	adaptive dynamic programming
ARE	algebraic Riccati equation
BE	Bellman error
CL	concurrent learning
DP	dynamic programming
DRE	differential Riccati equation
GHJB	generalized Hamlton-Jacobi-Bellman
HJ	Hamilton-Jacobi
HJB	Hamilton-Jacobi-Bellman
LIP	linear-in-the-parameters
LP	linearly parameterizable / linearly parameterized
MPC	model predictive control
NN	neural network
PE	persistence of excitation / persistently exciting
PI	policy iteration
RL	reinforcement learning
SDRE	state dependent Riccati equations
TD	temporal-difference
UB	ultimately bounded
VI	value iteration

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

MODEL-BASED REINFORCEMENT LEARNING FOR ONLINE APPROXIMATE OPTIMAL CONTROL

By

Rushikesh Lambodar Kamalapurkar

August 2014

Chair: Warren E. Dixon
Major: Mechanical Engineering

The objective of an optimal control synthesis problem is to compute the policy that an agent should follow in order to maximize the accumulated reward. Analytical solution of optimal control problems is often impossible when the system dynamics are nonlinear. Many numerical solution techniques are available to solve optimal control problems; however, such methods generally require perfect model knowledge and may not be implementable in real-time.

Inroads to solve optimal control problems for nonlinear systems can be made through insights gained from examining the value function. Under a given policy, the value function provides a map from the state space to the set of real numbers that measures the value of a state, generally defined as the total accumulated reward starting from that state. If the value function is known, a reasonable strategy is to apply control to drive the states towards increasing value. If the value function is unknown, a reasonable strategy is to use input-output data to estimate the value function online, and use the estimate to compute the control input. Reinforcement learning (RL)-based optimal control synthesis techniques implement the aforementioned strategy by approximating the value function using a parametric approximation scheme. The approximate optimal policy is then computed based on the approximate value function.

RL-based techniques are valuable not only as online optimization tools but also as control synthesis tools. In discrete-time stochastic systems with countable state

and action spaces RL-based techniques have demonstrated the ability to synthesize stabilizing policies with minimal knowledge of the structure of the system. Techniques such as Q-learning have shown to be effective tools to generate stabilizing policies based on input-output data without any other information about the system. RL thus offers a potential alternative to traditional control design techniques. However, the extensions of RL techniques to continuous-time systems that evolve on a continuous state-space are scarce, and often require more information about the system than just input-output data.

This dissertation investigates extending the applicability of RL-based techniques in a continuous-time deterministic setting to generate approximate optimal policies online by relaxing some of the limitations imposed by the continuous-time nature of the problem. State-of-the-art implementations of RL in continuous-time systems require a restrictive PE condition for convergence to optimality. In this dissertation, model-based RL is implemented via simulation of experience to relax the restrictive persistence of excitation condition. The RL-based approach is also extended to obtain approximate feedback-Nash equilibrium solutions to N-player nonzero-sum games.

In trajectory tracking problems, since the error dynamics are nonautonomous, the value function depends explicitly on time. Since universal function approximators can approximate functions with arbitrary accuracy only on compact domains, value functions for infinite-horizon optimal tracking problems cannot be approximated with arbitrary accuracy using universal function approximators. Hence, the extension of RL-based techniques to optimal tracking problems for continuous-time nonlinear systems has remained a non-trivial open problem. In this dissertation, RL-based approaches are extended to solve trajectory tracking problems by using the desired trajectory, in addition to the tracking error, as an input to learn the value function.

Distributed control of groups of multiple interacting agents is a challenging problem with multiple practical applications. When the agents possess cooperative or competitive objectives, the trajectory and the decisions of each agent are affected by the trajectories and decisions of the neighboring agents. The external influence renders the dynamics of each agent nonautonomous; hence, optimization in a network of agents presents challenges similar to the optimal tracking problem. The interaction between the agents in a network is often modeled as a differential game on a graph, defined by coupled dynamics and coupled cost functions. Using insights gained from the tracking problem, this dissertation extends the model-based RL technique to generate feedback-Nash equilibrium optimal policies online, for agents in a network with cooperative or competitive objectives. In particular, the network of agents is separated into autonomous subgraphs, and the differential game is solved separately on each subgraph.

The applicability of the developed methods is demonstrated through simulations, and to illustrate their effectiveness, comparative simulations are presented wherever alternate methods exist to solve the problem under consideration. The dissertation concludes with a discussion about the limitations of the developed technique, and further extensions of the technique are proposed along with the possible approaches to achieve them.

CHAPTER 1
INTRODUCTION

1.1 Motivation

The ability to learn the correct behavior from interactions with the environment is a highly desirable characteristic of a cognitive agent. Typical interactions between an agent and its environment can be described in terms of actions, states, and rewards (or penalties). The actions executed by the agent affect the state of the system (i.e., the agent and the environment), and the agent is presented with a reward (or a penalty). Assuming that the agent chooses an action based on the state of the system, the behavior (or the policy) of the agent can be described as a map from the state space to the action space.

To learn the correct policy, it is crucial to establish a measure of correctness. The correctness of a policy can be quantified in numerous ways depending on the objectives of the agent-environment interaction. For guidance and control applications, the correctness of a policy is often quantified in terms of a Lagrange cost and a Meyer cost. The Lagrange cost is the cumulative penalty accumulated along a path traversed by the agent and the Meyer cost is the penalty at the boundary. Policies with lower total cost are considered better and policies that minimize the total cost are considered optimal. The problem of finding the optimal policy that minimizes the total Lagrange and Meyer cost is known as the Bolza optimal control problem.

Obtaining an analytical solution to the Bolza problem is often infeasible if the system dynamics are nonlinear. Many numerical solution techniques are available to solve Bolza problems; however, numerical solution techniques require exact model knowledge and are realized via open-loop implementation of offline solutions. Open-loop implementations are sensitive to disturbances, changes in objectives, and changes in the system dynamics; hence, online closed-loop solutions of optimal control problems are sought-after. Inroads to solve an optimal control problem online can be made by looking at the so-called value function. Under a given policy, the value function provides a map from the state space to the set of real numbers that measures the quality of a state. In other words, under a given policy, the value function evaluated at a given state is the cost accumulated when starting in the given state and following the given policy. Under general conditions, the policy that drives the system state along the steepest negative gradient of the optimal value function turns out to be the optimal policy; hence, online optimal control design relies on computation of the optimal value function.

For systems with finite state and action spaces, value function-based dynamic programming (DP) techniques such as policy iteration (PI) and value iteration (VI) are established as effective tools for optimal control synthesis. However, both PI and VI suffer from Bellman's curse of dimensionality, i.e., they become computationally intractable as the size of the state space grows. Furthermore, both PI and VI require exact knowledge of the system dynamics. The need for excessive computation can be realistically sidestepped if one seeks to obtain an approximation to the optimal value function instead of the exact optimal value function (i.e., approximate dynamic programming). The need for exact model knowledge can be eliminated by using a simulation-based approach where the goal is to learn the optimal value function using state-action-reward triplets observed along the state trajectory (i.e., reinforcement learning (RL)).

Approximate dynamic programming algorithms approximate the classical PI and VI algorithms by using a parametric approximation of the policy or the value function. The central idea is that if the policy or the value function can be parameterized with sufficient accuracy using a small number of parameters, the optimal control problem reduces to an approximation problem in the parameter space. Furthermore, this formulation lends itself to an online solution approach using RL where the parameters are adjusted on-the-fly using input-output data. However, sufficient exploration of the state-action space is required for convergence, and the optimality of the obtained policy depends heavily on the accuracy of the parameterization scheme; the formulation of which requires some insight into the dynamics of the system. Despite the aforementioned drawbacks, RL has given rise to effective techniques that can synthesize nearly optimal policies to control nonlinear systems that have large state and action spaces and unknown or partially known dynamics. As a result, RL has been a growing area of research in the past two decades.

In recent years, RL techniques have been extended to autonomous continuous-time deterministic systems. In online implementations of RL, the control policy derived from the approximate value function is used to control the system; hence, obtaining a good approximation of the value function is critical to the stability of the closed-loop system. Obtaining a good approximation of the value function online requires convergence of the unknown parameters to their ideal values. Hence, similar to adaptive control, the sufficient exploration condition manifests itself as a persistence of excitation (PE) condition when RL is implemented online. In general, it is impossible to guarantee PE a priori; hence, a probing signal designed using trial and error is added to the controller to ensure PE. The probing signal is not considered in the stability analysis; hence, stability of the closed-loop implementation cannot be guaranteed. In this dissertation, a model-based RL scheme is developed to relax the PE condition. Model-based RL is implemented using a concurrent learning (CL)-based system identifier to simulate experience by evaluating the Bellman error (BE) over unexplored areas of the state space.

A multitude of relevant control problems can be modeled as multi-input systems, where each input is computed by a player, and each player attempts to influence the system state to minimize its own cost function. In this case, the optimization problem for each player is coupled with the optimization problem for other players; hence, in general, an optimal solution in the usual sense does not exist, motivating the formulation of alternative solution concepts. The most popular solution concept is a Nash equilibrium solution which finds applications in optimal disturbance rejection, i.e., H_∞ control, where the disturbance is modeled as a player in a two-player nonzero-sum differential game. A set of policies is called a Nash equilibrium solution to a multi-objective optimization problem if none of the players can improve their outcome by changing their policy while all the other players abide by the Nash equilibrium policies. Thus, Nash equilibrium solutions provide a secure set of strategies, in the sense that none of the players have an incentive to diverge from their equilibrium policy. Motivated by the widespread applications of differential games, this dissertation extends the model-based RL techniques to obtain feedback-Nash equilibrium solutions to $N-$player nonzero-sum differential games.

Extension of RL to trajectory tracking problems is not trivial because the error dynamics are nonautonomous, resulting in time-varying value functions. Since universal function approximators can approximate functions with arbitrary accuracy only on compact domains, value functions for infinite-horizon optimal tracking problems cannot be approximated with arbitrary accuracy using universal function approximators. The results in this dissertation extend RL-based approaches to trajectory tracking problems by using the desired trajectory, in addition to the tracking error, as an input to learn the value function.

The fact that the value function depends on the desired trajectory results in a challenge in establishing system stability during the learning phase. Stability during the learning phase is often established using Lyapunov-based stability analysis methods, which are motivated by the fact that under general conditions, the optimal value function is a Lyapunov function for the closed-loop system under the optimal policy. In tracking problems, the value function, as a function of the tracking error and the desired trajectory is not a Lyapunov function for the closed-loop system under the optimal policy. In this dissertation, the aforementioned challenge is addressed by proving that the value function, as a time-varying function of the tracking error can be used as a Lyapunov function.

RL techniques are valuable not only for optimization but also for control synthesis in complex systems such as a distributed network of cognitive agents. Combined efforts from multiple autonomous agents can yield tactical advantages including: improved munitions effects; distributed sensing, detection, and threat response; and distributed communication pipelines. While coordinating behaviors among autonomous agents is a challenging problem that has received mainstream focus, unique challenges arise when seeking autonomous collaborative behaviors in low bandwidth communication environments. For example, most collaborative control literature focuses on centralized approaches that require all nodes to continuously communicate with a central agent, yielding a heavy communication demand that is subject to failure due to delays, and missing information. Furthermore, the central agent is required to carry enough on-board computational resources to process the data and to generate command signals. These challenges motivate the need for a decentralized approach where the nodes only need to communicate with their neighbors for guidance, navigation and control tasks. Furthermore, when the agents posses cooperative or competitive objectives, the trajectory and the decisions of each agent are affected by the trajectories and decisions of the neighboring agents. The external influence renders the dynamics of each agent nonautonomous, and hence, optimization in a network of agents presents challenges similar to the optimal tracking problem.

The interaction between the agents in a network is often modeled as a differential game on a graph, defined by coupled dynamics and coupled cost functions. Using insights gained from the tracking problem, this dissertation extends the model-based RL technique to generate feedback-Nash equilibrium optimal policies online, for agents in a network with cooperative or competitive objectives. In particular, the network of agents is separated into autonomous subgraphs, and the differential game is solved separately on each subgraph.

The applicability of the developed methods is demonstrated through simulations, and to illustrate their effectiveness, comparative simulations are presented wherever alternate methods exist to solve the problem under consideration. The dissertation concludes with a discussion about the limitations of the developed technique, and further extensions of the technique are proposed along with the possible approaches to achieve them.

1.2 Literature Review

One way to develop optimal controllers for general nonlinear systems is to use numerical methods [1]. A common approach is to formulate the optimal control problem in terms of a Hamiltonian and then to numerically solve a two point boundary value problem for the state and co-state equations [2, 3]. Another approach is to cast the optimal control problem as a nonlinear programming problem via direct transcription and then solve the resulting nonlinear program [4–9]. Numerical methods are offline, do not generally guarantee stability, or optimality, and are often open-loop. These issues motivate the desire to find an analytical solution. Developing analytical solutions to optimal control problems for linear systems is complicated by the need to solve an algebraic Riccati equation (ARE) or a differential Riccati equation (DRE). Developing analytical solutions for nonlinear systems is even further complicated by the sufficient condition of solving a Hamilton-Jacobi-Bellman (HJB) partial differential equation, where an analytical solution may not exist in general. If the nonlinear dynamics are exactly known, then the problem can be simplified at the expense of optimality by solving an ARE through feedback-linearization methods (cf. [10–14]).

Alternatively, some investigators temporarily assume that the uncertain system could be feedback-linearized, solve the resulting optimal control problem, and then use adaptive/learning methods to asymptotically learn the uncertainty, i.e., asymptotically converge to the optimal controller [15–18]. Inverse optimal control [19–24] is also an alternative method to solve the nonlinear optimal control problem by circumventing the need to solve the HJB equation. By finding a control Lyapunov function, which can be shown to also be a value function, an optimal controller can be developed that optimizes a derived cost. However, since the cost is derived rather than specified by mission/task objectives, this approach is not explored in this dissertation. Optimal control-based algorithms such as state dependent Riccati equations (SDRE) [25–28] and model-predictive control (MPC) [29–35] have been widely utilized for control of nonlinear systems. However, both SDRE and MPC are inherently model-based. Furthermore, due to nonuniqueness of state dependent linear factorization in SDRE-based techniques, and since the control problem is solved over a small prediction horizon in MPC, SDRE and MPC generally result in suboptimal policies. Furthermore, MPC-based approaches are computationally intensive, and closed-loop stability of SDRE-based methods is generally impossible to establish a priori and has to be established through extensive simulation. Owing to the aforementioned drawbacks, SDRE and MPC approaches are not explored in this dissertation. This dissertation focuses on DP-based techniques.

The fundamental idea in all DP techniques is the principle of optimality, due to Bellman [36]. DP techniques based on the principle of optimality have been extensively studied in literature (cf. [37–42]). The applicability of classical DP techniques like PI and VI is limited by the curse of dimensionality and the need for model knowledge. Simulation-based reinforcement learning (RL) techniques such as Q-learning [40] and temporal-difference (TD)-learning [38, 43] avoid the curse of dimensionality and the need for exact model knowledge. However, these techniques require the states and the actions to be on finite sets. Even though the theory is developed for finite state spaces of any size, the implementation of simulation-based RL techniques is feasible only if the size of the state space is small. Extensions of simulation-based RL techniques to general state spaces or very large finite state spaces involve parametric approximation of the policy. Such algorithms have been studied in depth for systems with countable state and action spaces under the name of neuro-DP (cf. [42, 44–48] and the references therein). The extension of these techniques to general state spaces and continuous time-domains is challenging and only a small number of results are available in the literature [49].

For deterministic systems, RL algorithms have been extended to a solve finite and infinite-horizon discounted and total cost optimal regulation problems (cf. [50–59]) under names such as adaptive dynamic programming (ADP) or adaptive critic algorithms. The discrete/iterative nature of the approximate dynamic programming formulation lends itself naturally to the design of discrete-time optimal controllers [50, 53, 55, 60–67], and the convergence of algorithms for DP-based RL controllers is studied in results such as [61, 68–70]. Most prior work has focused on convergence analysis for discrete-time systems, but some continuous examples are available [52, 54, 57, 70–79]. For example, in [72] Advantage Updating was proposed as an extension of the Q-learning algorithm which could be implemented in continuous time and provided faster convergence. The result in [74] used a HJB-based framework to derive algorithms for value function approximation and policy improvement, based on a continuous version of the TD error. An HJB framework was also used in [70] to develop a stepwise stable iterative approximate dynamic programming algorithm for continuous input-affine systems with an input-quadratic performance measure. Based on the successive approximation method first proposed in [71], an adaptive optimal control solution is provided in [73], where a Galerkin's spectral method is used to approximate the solution to the generalized HJB (GHJB). A least-squares-based successive approximation solution to the GHJB is provided in [52], where an NN is trained offline to learn the GHJB solution. Another continuous formulation is proposed in [75].

In online real-time applications, DP-based techniques generally require a restrictive PE condition to establish stability and convergence. However, recent research indicates that data-driven learning based on recorded experience can improve the efficiency of information utilization, thereby mollifying the PE requirements. Experience replay techniques have been studied in RL literature to circumvent the PE requirement, which is analogous to the requirement of sufficient exploration. Experience replay techniques involve repeated processing of recorded input-output data in order to improve efficiency of information utilization [80–85].

ADP-based methods that seek an online solution to the optimal control problem, (cf., [53, 57, 59, 63, 86, 87] and the references therein) are structurally similar to adaptive control schemes. In adaptive control, the estimates for the uncertain parameters in the plant model are updated using the current tracking error as the performance metric, whereas, in online RL-based techniques, estimates for the uncertain parameters in the value function are updated using a continuous-time counterpart of the TD error, called the BE, as the performance metric. Convergence of online RL-based techniques to the optimal solution is analogous to parameter convergence in adaptive control.

Parameter convergence has been a focus of research in adaptive control for several decades. It is common knowledge that the least squares and gradient descent-based update laws generally require PE in the system state for convergence of the parameter estimates. Modification schemes such as projection algorithms, $\sigma-$modification, and $e-$modification are used to guarantee boundedness of parameter estimates and overall system stability; however, these modifications do not guarantee parameter convergence unless the PE condition, which is often impossible to verify online, is satisfied [88–91].

As recently shown in results such as [92] and [93], CL-based methods can be used to guarantee parameter convergence in adaptive control without relying on the PE condition. Concurrent learning relies on recorded state information along with current state measurements to update the parameter estimates. Learning from recorded data is effective since it is based on the model error, which is closely related to the parameter estimation error. The key concept that enables the computation of the model error from past recorded data is that the model error can be computed if the state derivative is known, and the state derivative can be accurately computed at a past recorded data point using numerical smoothing techniques [92, 93]. Similar techniques have been recently shown to be effective for online real-time optimal control [94, 95]. In particular, the results in [95] indicate that recorded values of the BE can be used to solve the online real-time optimal control problem without the need of PE. However, a finite amount of added probing noise is required for the recorded data to be rich enough. Inspired by the results in [96] and [97], which suggest that simulated experience based on a system model can be more effective than recorded experience, the efforts in this dissertation focus on the development of online real-time optimal control techniques based on model learning and BE extrapolation.

A multitude of relevant control problems can be modeled as multi-input systems, where each input is computed by a player, and each player attempts to influence the system state to minimize its own cost function. In this case, the optimization problem for each player is coupled with the optimization problem for other players, and hence, in general, an optimal solution in the usual sense does not exist, motivating the formulation of alternative optimality criteria.

Differential game theory provides solution concepts for many multi-player, multi-objective optimization problems [98–100]. For example, a set of policies is called a Nash equilibrium solution to a multi-objective optimization problem if none of the players can improve their outcome by changing their policy while all the other players abide by the Nash equilibrium policies [101]. Thus, Nash equilibrium solutions provide a secure set of strategies, in the sense that none of the players have an incentive to diverge from their equilibrium policy. Hence, Nash equilibrium has been a widely used solution concept in differential game-based control techniques.

In general, Nash equilibria are not unique. For a closed-loop differential game (i.e., the control is a function of the state and time) with perfect information (i.e. all the players know the complete state history), there can be infinitely many Nash equilibria. If the policies are constrained to be feedback policies, the resulting equilibria are called (sub)game perfect Nash equilibria or feedback-Nash equilibria. The value functions corresponding to feedback-Nash equilibria satisfy a coupled system of Hamilton-Jacobi (HJ) equations [102–107].

If the system dynamics are nonlinear and uncertain, an analytical solution of the coupled HJ equations is generally infeasible; hence, dynamic programming-based approximate solutions are sought [56, 58, 87, 108–112]. In this dissertation, a simulation-based actor-critic-identifier (ACI) architecture is developed to obtain an approximate feedback-Nash equilibrium solution to an infinite horizon N-player nonzero-sum differential game online, without requiring PE, for a nonlinear control-affine system with uncertain linearly parameterized drift dynamics.

For trajectory tracking problems in discrete-time systems, several approaches have been developed to address the nonautonomous nature of the open-loop system. Park et.al. [113] use generalized backpropagation through time to solve a finite horizon tracking problem that involves offline training of neural networks (NNs). An ADP-based approach is presented in [114] to solve an infinite-horizon optimal tracking problem where the desired trajectory is assumed to depend on the system states. A greedy heuristic dynamic programming based algorithm is presented in [86] which uses a system transformation to express a nonautonomous system as an autonomous system. However, this result lacks an accompanying stability analysis. ADP-based approaches are presented in [115, 116] for tracking in continuous-time systems. In both the results, the value function (i.e. the critic) and the controller (i.e. the actor) presented are time-varying functions of the tracking error. However, since the problem is an infinite-horizon optimal control problem, time does not lie on a compact set. NNs can only approximate functions on a compact domain. Thus, it is unclear how a NN with time invariant basis functions can approximate the time-varying value function and the policy.

For problems with multiple agents, as the desired action by an individual agent depends on the actions and the resulting trajectories of its neighbors, the error system for each agent becomes a complex nonautonomous dynamical system. Nonautonomous systems, in general, have non-stationary value functions. Since non-stationary functions are difficult to approximate using parameterized function approximation schemes such as NNs, designing optimal policies for nonautonomous systems is not trivial. To address this challenge, differential game theory is often employed in multi-agent optimal control, where solutions to coupled Hamilton-Jacobi (HJ) equations (c.f. [112]) are sought. Since the coupled HJ equations are difficult to solve, some form of RL is often employed to get an approximate solution. Results such as [58, 112, 117–120] indicate that ADP can be used to generate approximate optimal policies online for multi-agent systems. Since the HJ equations are coupled, all of these results have a centralized control architecture.

Decentralized control techniques focus on finding control policies based on local data for individual agents that collectively achieve the desired goal, which, for the problem considered in this effort, is tracking a desired trajectory while maintaining a desired formation. Various methods have been developed to solve formation tracking problems for linear systems (cf. [121–125] and the references therein). For nonlinear systems, MPC-based approaches ([126, 127]) and ADP-based approaches ([128, 129]) have been proposed. The MPC-based controllers require extensive numerical computations and lack stability and optimality guarantees. The ADP-based approaches either require offline computations, or are suboptimal because not all the inter-agent interactions are considered in the value function. In this dissertation, a simulation-based ACI architecture is developed to cooperatively control a group of agents to track a trajectory while maintaining a desired formation.

1.3 Outline of the Dissertation

Chapter 1 serves as the introduction. Motivation behind the results in the dissertation is presented along with a detailed review of the state of the art.

Chapter 2 contains a brief review of available techniques used in the application of RL to deterministic continuous-time systems. This chapter also highlights the problems and the limitations of existing techniques, thereby motivating the development in the dissertation.

Chapter 3 implements model-based RL to solve approximate optimal regulation problems online with a relaxed PE-like condition using a simulation-based ACI architecture. The development is based on the observation that, given a model of the system, model-based RL can be implemented by evaluating the Bellman error at any number of desired points in the state space. In this result, a parametric system model is considered, and a CL-based parameter identifier is developed to compensate for uncertainty in the parameters. Ultimately bounded (UB) regulation of the system states to a neighborhood of the origin, and convergence of the developed policy to a neighborhood of the optimal policy are established using a Lyapunov-based analysis, and simulations are presented to demonstrate the performance of the developed controller.

Chapter 4 extends the results of Chapter 3 to obtain an approximate feedback-Nash equilibrium solution to an infinite-horizon N-player nonzero-sum differential game online, without requiring PE, for a nonlinear control-affine system with uncertain linearly parameterized drift dynamics. It is shown that under a condition milder than PE, uniformly ultimately bounded convergence of the developed control policies to the feedback-Nash equilibrium policies can be established. Simulation results are presented to demonstrate the performance of the developed technique without an added excitation signal.

Chapter 5 presents an ADP-based approach using the policy evaluation (Critic) and policy improvement (Actor) architecture to approximately solve the infinite-horizon optimal tracking problem for control-affine nonlinear systems with quadratic cost. The problem is solved by transforming the system to convert the tracking problem that has a non-stationary value function, into a stationary optimal control problem. The ultimately bounded UB tracking and estimation result is established using Lyapunov analysis for nonautonomous systems. Simulations are performed to demonstrate the applicability and the effectiveness of the developed method.

Chapter 6 utilizes model-based reinforcement learning to extend the results of Chapter 5 to systems with uncertainties in drift dynamics. A system identifier is used for approximate model inversion to facilitate the formulation of a feasible optimal control problem. Model-based reinforcement learning is implemented using a concurrent learning-based system identifier to simulate experience by evaluating the Bellman error over unexplored areas of the state space. Tracking of the desired trajectory and convergence of the developed policy to a neighborhood of the optimal policy is established via Lyapunov-based stability analysis. Simulation results demonstrate the effectiveness of the developed technique.

Chapter 7 combines graph theory and differential game theory with the actor-critic-identifier architecture in ADP to synthesize approximate online feedback-Nash equilibrium control policies for agents on a communication network with a spanning tree. NNs are used to approximate the policy, the value function, and the system dynamics. UB convergence of the agents to the desired formation, UB convergence of the agent trajectories to the desired trajectories, and UB convergence of the agent controllers to their respective feedback-Nash equilibrium policies is established through a Lyapunov-based stability analysis. Simulations are presented to demonstrate the applicability of the proposed technique to cooperatively control a group of five agents.

Chapter 8 concludes the dissertation. A summary of the dissertation is provided along with a discussion on open problems and future research directions.

1.4 Contributions

This section details the contributions of this dissertation over the state-of-the-art.

1.4.1 Approximate Optimal Regulation

In RL-based approximate online optimal control, the HJB equation along with an estimate of the state derivative (cf. [49, 59]), or an integral form of the HJB equation (cf. [130]) is utilized to approximately evaluate the BE at each visited state along the system trajectory. The BE provides an indirect measure of the quality of the current estimate of the value function at each visited state along the system trajectory. Hence, the unknown value function parameters are updated based on the BE along the system trajectory. Such weight update strategies create two challenges for analyzing convergence. The system states need to be PE, and the system trajectory needs to visit enough points in the state space to generate a good approximation to the value function over the entire operating domain. These challenges are typically addressed by adding an exploration signal to the control input (cf. [43, 49, 130]) to ensure sufficient exploration in the desired region of the state space. However, no analytical methods exist to compute the appropriate exploration signal when the system dynamics are nonlinear.

In this dissertation, the aforementioned challenges are addressed by observing that the restriction that the BE can only be evaluated along the system trajectories is a consequence of the model-free nature of RL-based approximate online optimal control. In particular, the integral BE is only meaningful as a measure of quality of the value function if evaluated along the system trajectories, and state derivative estimators can only generate estimates of the state derivative along the system trajectories using numerical smoothing. However, if the system dynamics are known, the state derivative, and hence, the BE can be evaluated at any desired point in the state space. Unknown parameters in the value function can therefore be adjusted based on least square minimization of the BE evaluated at any number of desired points in the state space. For example, in an infinite-horizon regulation problem, the BE can be computed at sampled points uniformly distributed in a neighborhood around the origin of the state space. The results of this dissertation indicate that convergence of the unknown parameters in the value function is guaranteed provided the selected points satisfy a rank condition. Since the BE can be evaluated at any desired point in the state space, sufficient exploration can be achieved by appropriately selecting the points in a desired neighborhood.

If the system dynamics are partially unknown, an approximation to the BE can be evaluated at any desired point in the state space based on an estimate of the system dynamics. If each new evaluation of the BE along the system trajectory is interpreted as gaining experience via exploration, an evaluation of the BE at an unexplored point in the state space can be interpreted as a simulated experience. Learning based on simulation of experience has been investigated in results such as [131–136] for stochastic model-based RL; however, these results solve the optimal control problem offline in the sense that repeated learning trials need to be performed before the algorithm learns the controller, and system stability during the learning phase is not analyzed. This dissertation furthers the state of the art for nonlinear, control-affine plants with linearly parameterizable (LP) uncertainties in the drift dynamics by providing an online solution to deterministic infinite-horizon optimal regulation problems. In this dissertation, a CL-based parameter estimator is developed to exponentially identify the unknown parameters in the system model, and the parameter estimates are used to implement simulated experience by extrapolating the BE. The main contributions of this chapter include:

- Novel implementation of simulated experience in deterministic nonlinear systems using CL-based system identification.
- Detailed stability analysis to establish simultaneous online identification of system dynamics and online approximate learning of the optimal controller, while maintaining system stability. The stability analysis shows that provided the system dynamics can be approximated fast enough, and with sufficient accuracy, simulation of experience based on the estimated model implemented via approximate BE extrapolation can be utilized to approximately solve an infinite-horizon optimal regulation problem online are provided.
- For the first time ever, simulation results that demonstrate the approximate solution of an infinite-horizon optimal regulation problem online for an inherently unstable

control-affine nonlinear system with uncertain drift dynamics without the addition of an external ad-hoc probing signal.

1.4.2 N-player Nonzero-sum Differential Games

In [58], a PE-based integral reinforcement learning algorithm is presented to solve nonzero-sum differential games in linear systems without the knowledge of the drift matrix. In [112], a PE-based dynamic programming technique is developed to find an approximate feedback-Nash equilibrium solution to an infinite-horizon N-player nonzero-sum differential game online for nonlinear control-affine systems with known dynamics. In [119], a PE-based ADP method is used to solve a two-player zero-sum game online for nonlinear control-affine systems without the knowledge of drift dynamics. In this dissertation, a simulation-based ACI architecture (cf. [59]) is used to obtain an approximate feedback-Nash equilibrium solution to an infinite-horizon N-player nonzero-sum differential game online, without requiring PE, for a nonlinear control-affine system with uncertain LP drift dynamics. The contribution of this result is that it extends the development in Chapter 3 to the more general $N-$player nonzero-sum differential game framework.

1.4.3 Approximate Optimal Tracking

Approximation techniques like NNs are commonly used in ADP literature for value function approximation. ADP-based approaches are presented in results such as [115, 116] to address the tracking problem for continuous time systems, where the value function, and the controller presented are time-varying functions of the tracking error. However, for an infinite-horizon optimal control problem, the domain of the value function is not compact. NNs can only approximate functions on a compact domain. Thus, it is unclear how a NN with the tracking error as an input can approximate the time-varying value function and controller.

For discrete time systems, several approaches have been developed to address the tracking problem. Park et.al. [113] use generalized back-propagation through time to solve a finite horizon tracking problem that involves offline training of NNs. An ADP-based approach is presented in [114] to solve an infinite-horizon optimal tracking problem where the desired trajectory is assumed to depend on the system states. Greedy heuristic dynamic programming based algorithms are presented in results such as [86, 137, 138] which transform the nonautonomous system into an autonomous system, and approximate convergence of the sequence of value functions to the optimal value function is established. However, these results lack an accompanying stability analysis. In this result, the tracking error and the desired trajectory both serve as inputs to the NN for value function approximation. Effectiveness of the developed technique is demonstrated via numerical simulations. The main contributions of this result include:

- Formulation of a stationary optimal control problem for infinite-horizon total-cost optimal tracking control.
- Formulation and proof of the hypothesis that the optimal value function is a valid candidate Lyapunov function when interpreted as a time-varying function of the tracking error.
- New Lyapunov-like stability analysis to establish ultimate boundedness under sufficient persistent excitation.

1.4.4 Model-based Reinforcement Learning for Approximate Optimal Tracking

This chapter extends the actor-critic method developed in the previous chapter to solve an infinite-horizon optimal tracking problem for systems with unknown drift dynamics using model-based RL. The development in the previous chapter relies on minimizing the difference between the implemented controller and the steady-state controller. The computation of the steady-state controller requires exact model knowledge. In this chapter, a CL-based system identifier is developed generate an online approximation to the steady-state controller. Furthermore, the CL-based system identifier is also used to implement model-based RL to simulate experience by evaluating the BE over unexplored areas of the state space. Effectiveness of the developed technique is demonstrated via numerical simulations. The main contributions of this result include:

- Extension of tracking technique to systems with uncertain drift dynamics via the use of a CL-based system identification for approximate model inversion.
- Lyapunov-based stability analysis to show simultaneous system identification and ultimately bounded tracking in the presence of uncertainties.

1.4.5 Differential Graphical Games

Various methods have been developed to solve formation tracking problems for linear systems. An optimal control approach is used in [139] to achieve consensus while avoiding obstacles. In [140], an optimal controller is developed for agents with known dynamics to cooperatively track a desired trajectory. In [141] an inverse optimal controller is developed for unmanned aerial vehicles to cooperatively track a desired trajectory while maintaining a desired formation. In [142] a differential game-based approach is developed for unmanned aerial vehicles to achieve distributed Nash strategies. In [143], an optimal consensus algorithm is developed for a cooperative team of agents with linear dynamics using only partial information. A value function approximation based approach is presented in [128] for cooperative synchronization in a strongly connected network of agents with known linear dynamics.

For nonlinear systems, an MPC-based approach is presented in [126], however, no stability or convergence analysis is presented. A stable distributed MPC-based approach is presented in [127] for nonlinear discrete-time systems with known nominal dynamics. Asymptotic stability is proved without any interaction between the nodes, however, a nonlinear optimal control problem need to be solved at every iteration to implement the controller. An optimal tracking approach for formation control is presented in [129] using single network adaptive critics where the value function is learned offline. Online feedback-Nash equilibrium solution of differential graphical games in a topological network of agents with continuous-time uncertain nonlinear dynamics has remained an open problem. The contributions of this chapter are the following:

- Introduction of relative control error minimization technique to facilitate the formulation of a feasible infinite-horizon total-cost differential graphical game.

- Development a set of coupled HJ equations corresponding to feedback-Nash equilibrium solutions of differential graphical games.
- Lyapunov-based stability analysis to show ultimately bounded formation tracking in the presence of uncertainties.

CHAPTER 2
PRELIMINARIES

2.1 Notation

Throughout the dissertation, $\mathbb{R}^n$ denotes $n-$dimensional Euclidean space, $\mathbb{R}_{>a}$ denotes the set of real numbers strictly greater than $a \in \mathbb{R}$, and $\mathbb{R}_{\geq a}$ denotes the set of real numbers greater than or equal to $a \in \mathbb{R}$. Unless otherwise specified, the domain of all the functions is assumed to be $\mathbb{R}_{\geq 0}$. Functions with domain $\mathbb{R}_{\geq 0}$ are defined by abuse of notation using only their image. For example, the function $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is defined by abuse of notation as $x \in \mathbb{R}^n$, and referred to as x instead of $x(t)$. By abuse of notation, the state variables are also used to denote state trajectories. For example, the state variable x in the equation $\dot{x} = f(x) + u$ is also used as $x(t)$ to denote the state trajectory i.e., the general solution $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ to $\dot{x} = f(x) + u$ evaluated at time t. Unless otherwise specified, all the mathematical quantities are assumed to be time-varying. Unless otherwise specified, an equation of the form $g(x) = f + h(y,t)$ is interpreted as $g(x(t)) = f(t) + h(y(t), t)$ for all $t \in \mathbb{R}_{\geq 0}$, and a definition of the form $g(x,y) \triangleq f(y) + h(x)$ for functions $g : A \times B \to C$, $f : B \to C$ and $h : A \to C$ is interpreted as $g(x,y) \triangleq f(y) + h(x)\,, \forall (x,y) \in A \times B$. The only exception to the aforementioned equation and definition notation is the definitions of cost functionals, where the arguments to the cost functional are functions. The total derivative $\frac{\partial f(x)}{\partial x}$ is denoted by ∇f and the partial derivative $\frac{\partial f(x,y)}{\partial x}$ is denoted by $\nabla_x f(x,y)$. An $n \times n$ identity matrix is denoted by I_n, $n \times m$ matrices of zeros and ones are denoted by $\mathbf{0}_{n\times m}$ and $\mathbf{1}_{n\times m}$, respectively, and $\mathbf{1}_S$ denotes the indicator function of the set S.

2.2 Problem Formulation

The focus of this dissertation is to obtain online approximate solutions to infinite-horizon total-cost optimal control problems. To facilitate the formulation of the optimal control problem, Consider a control-affine nonlinear dynamical system

$$\dot{x} = f(x) + g(x)u, \tag{2–1}$$

where $x \in \mathbb{R}^n$ denotes the system state, $u \in \mathbb{R}^m$ denotes the control input, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ denotes the control effectiveness matrix. The functions f and g are assumed to be locally Lipschitz continuous functions such that $f(0) = 0$ and $\nabla f(x)$ is continuous and bounded for every bounded $x \in \mathbb{R}^n$. In the following, the notation $\phi^u(t; t_0, x_0)$ denotes a trajectory of the system in (2–1) under the control signal u with the initial condition $x_0 \in \mathbb{R}^n$ and initial time $t_0 \in \mathbb{R}_{\geq 0}$.

The control objective is to solve the infinite-horizon optimal regulation problem online, i.e., to simultaneously design and utilize a control signal u online to minimize the cost functional

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

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

$$r(x, u) \triangleq Q(x) + u^T R u, \tag{2–3}$$

where $Q : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a positive definite function and $R \in \mathbb{R}^{m \times m}$ is a constant positive definite symmetric matrix.

2.3 Exact Solution

It is well known that if the functions f, g, and Q are stationary (time-invariant) and the time-horizon is infinite, then the optimal control input is a stationary state-feedback policy $u(t) = \xi(x(t))$ for some function $\xi : \mathbb{R}^n \to \mathbb{R}^m$. Furthermore, the function that maps each state to the total accumulated cost starting from that state and following a stationary state-feedback policy, i.e., the value function, is also a stationary function. Hence, the optimal value function $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ can be expressed as

$$V^*(x) \triangleq \inf_{u(\tau) | \tau \in \mathbb{R}_{\geq t}} \int\limits_{t}^{\infty} r(\phi^u(\tau; t, x), u(\tau))\, d\tau, \tag{2–4}$$

for all $x \in \mathbb{R}^n$, where $U \subset \mathbb{R}^m$ is the action space. Assuming an optimal controller exists, the optimal value function can be expressed as

$$V^*(x) \triangleq \min_{u(\tau)|\tau\in\mathbb{R}_{\geq t}} \int_t^\infty r\left(\phi^u(\tau;t,x), u(\tau)\right) d\tau. \tag{2–5}$$

The optimal value function is characterized by the corresponding HJB equation [1]

$$0 = \min_{u \in U} \left(\nabla V(x^o)\left(f(x^o) + g(x^o)u\right) + r(x^o, u)\right), \tag{2–6}$$

for all $x^o \in \mathbb{R}^n$, with the boundary condition $V(0) = 0$. Provided the HJB in (2–6) admits a continuously differentiable solution, it constitutes a necessary and sufficient condition for optimality, i.e., if the optimal value function in (2–5) is continuously differentiable, then it is the unique solution to the HJB in (2–6) [144]. The optimal control policy $u^* : \mathbb{R}^n \to \mathbb{R}^m$ can be determined from (2–6) as [1]

$$u^*(x^o) = -\frac{1}{2}R^{-1}g^T(x^o)\left(\nabla V^*(x^o)\right)^T, \ \forall x^o \in \mathbb{R}^n. \tag{2–7}$$

The HJB in (2–6) can be expressed in the open-loop form

$$\nabla V^*(x^o)\left(f(x^o) + g(x^o)u^*(x^o)\right) + r\left(x^o, u^*(x^o)\right) = 0, \tag{2–8}$$

for all $x^o \in \mathbb{R}^n$, and using (2–7), the HJB in (2–8) can be expressed in the closed-loop form

$$\nabla V^*(x^o)f(x^o) - \frac{1}{4}\nabla V^*(x^o)g(x^o)R^{-1}g^T(x^o)\left(\nabla V^*(x^o)\right)^T + Q(x^o) = 0. \tag{2–9}$$

for all $x^o \in \mathbb{R}^n$. The optimal policy can now be obtained using (2–7) if the HJB in (2–9) can be solved for the optimal value function V^*.

2.4 Value Function Approximation

An analytical solution of the HJB equation is generally infeasible; hence, an approximate solution is sought. In an approximate actor-critic-based solution, the optimal value function V^* is replaced by a parametric estimate $\hat{V}\left(x, \hat{W}_c\right)$ and the optimal policy u^* by a parametric estimate $\hat{u}\left(x,\hat{W}_a\right)$ where $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ denote vectors of estimates of the ideal parameters. The objective of the critic is to learn the parameters $\hat{W}_c$, and the objective of the actor is to learn the parameters $\hat{W}_a$. Substituting the estimates $\hat{V}$ and $\hat{u}$ for V^* and u^* in (2–8), respectively, a residual error $\delta : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$, called the BE, is defined as

$$\delta\left(x,\hat{W}_c,\hat{W}_a\right) \triangleq \nabla_x \hat{V}\left(x,\hat{W}_c\right)\left(f\left(x\right)+g\left(x\right)\hat{u}\left(x,\hat{W}_a\right)\right)+r\left(x,\hat{u}\left(x,\hat{W}_a\right)\right). \tag{2–10}$$

To solve the optimal control problem, the critic aims to find a set of parameters $\hat{W}_c$ and the actor aims to find a set of parameters $\hat{W}_a$ such that $\delta\left(x^o,\hat{W}_c,\hat{W}_a\right) = 0$, and $\hat{u}\left(x^o,\hat{W}_a\right) = -\frac{1}{2}R^{-1}g^T\left(x^o\right)\left(\nabla\hat{V}\left(x^o,\hat{W}_c\right)\right)^T$ $\forall x^o \in \mathbb{R}^n$. Since an exact basis for value function approximation is generally not available, an approximate set of parameters that minimizes the BE is sought. In particular, to ensure uniform approximation of the value function and the policy over an operating domain $\mathcal{D} \subset \mathbb{R}^n$, it is desirable to find parameters that minimize the integral error $E_s : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$ defined as

$$E_s\left(\hat{W}_c,\hat{W}_a\right) \triangleq \int\limits_{x\in\mathcal{D}} \delta^2\left(x,\hat{W}_c,\hat{W}_a\right)dx.$$

Hence, in an online implementation of the deterministic actor-critic method, it is desirable to update the parameter estimates $\hat{W}_c$ and $\hat{W}_a$ online to minimize the instantaneous error $E_s\left(\hat{W}_c\left(t\right),\hat{W}_a\left(t\right)\right)$ or the cumulative instantaneous error

$$E\left(t\right) \triangleq \int\limits_0^t E_s\left(\hat{W}_c\left(\tau\right),\hat{W}_a\left(\tau\right)\right)d\tau, \tag{2–11}$$

while the system in (2–1) is being controlled using the control law $u\left(t\right) = \hat{u}\left(x\left(t\right),\hat{W}_a\left(t\right)\right)$.

2.5 RL-based Online Implementation

Computation of the BE in (2–10) and the integral error in (2–11) requires exact model knowledge. Furthermore, computation of the integral error in (2–11) is generally infeasible. Two prevalent approaches employed to render the control design robust to uncertainties in the system drift dynamics are integral RL (cf. [95] and [145]) and state derivative estimation (cf. [59] and [146]).

Integral RL exploits the fact that for all $T > 0$ and $t > t_0 + T$, the BE in (6–2) has an equivalent integral form $\delta_{int}(t) = \hat{V}\left(x(t-T), \hat{W}_c(t)\right) - \hat{V}\left(x(t), \hat{W}_c(t)\right) - \int_{t-T}^{t} r(x(\tau), u(\tau))\, d\tau$, where $u(t) = \hat{u}\left(x(t), \hat{W}_a(t)\right)$, $\forall t \in \mathbb{R}_{\geq t_0}$. Since the integral form does not require model knowledge, policies designed based on δ_{int} can be implemented without knowledge of f.

State derivative estimation-based techniques exploit the fact that if the system model is uncertain, the critic can compute the BE at each time instance t using the state-derivative $\dot{x}(t)$ as

$$\delta_t(t) \triangleq \nabla_x \hat{V}\left(x(t), \hat{W}_c(t)\right)\dot{x}(t) + r\left(x(t), \hat{u}\left(x(t), \hat{W}_a(t)\right)\right). \tag{2–12}$$

If the state-derivative is not directly measurable, an approximation of the BE can be computed using a dynamically generated estimate of the state-derivative. Note that the integral form of the BE is inherently dependent on the state trajectory, and since adaptive derivative estimators estimate the derivative only along the trajectory, the derivative estimation-based techniques are also dependent on the state trajectory. Hence, in techniques such as [59, 95, 145, 146] the BE can only be evaluated along the system trajectory.

Since (2–8) constitutes a necessary and sufficient condition for optimality, the BE serves as an indirect measure of how close the critic parameter estimates $\hat{W}_c$ are to their ideal values; hence, in RL literature, each evaluation of the BE is interpreted as gained experience. In particular, the critic receives state-derivative-action-reward tuples $(x(t), \dot{x}(t), u(t), r(x(t), u(t)))$ and computes the BE using (2–12). The critic then performs a one-step update to the parameter estimates $\hat{W}_c$ based on either the instantaneous experience, quantified by the squared error $\delta_t^2(t)$, or the cumulative experience, quantified by the integral squared error

$$E_t\left(t\right) \triangleq \int\limits_0^t \delta_t^2\left(\tau\right) d\tau, \tag{2–13}$$

using a steepest descent update law. The use of the cumulative squared error is motivated by the fact that in the presence of uncertainties, BE can only be evaluated along the system trajectory; hence, $E_t\left(t\right)$ is the closest approximation to $E\left(t\right)$ in (2–11) that can be computed using the available information.

Intuitively, for $E_t\left(t\right)$ to approximate $E\left(t\right)$ over an operating domain, the state trajectory $x\left(t\right)$ needs to visit as many points in the operating domain as possible. This intuition is formalized by the fact that the use of the approximation $E_t\left(t\right)$ to update the critic parameter estimates is valid provided certain exploration conditions[1] are met. In RL terms, the exploration conditions translate to the need for the critic to gain enough experience in order to learn the value function. The exploration conditions can be relaxed using experience replay, where each evaluation of the BE δ_{int} is interpreted as gained experience, and these experiences are stored in a history stack and are repeatedly used in the learning algorithm to improve data efficiency, however, a finite amount of exploration is still required since the values stored in the history stack are also constrained to the system trajectory.

While the estimates $\hat{W}_c$ are being updated by the critic, the actor simultaneously updates the parameter estimates $\hat{W}_a$ using a gradient-based approach so that the quantity $\hat{u}\left(x, \hat{W}_a\right) + \frac{1}{2}R^{-1}g^T\left(x\right)\left(\nabla\hat{V}\left(x, \hat{W}_c\right)\right)^T$ decreases. The weight updates are performed online in real-time while the system is being controlled using the control law $u\left(t\right) = \hat{u}\left(x\left(t\right), \hat{W}_a\left(t\right)\right)$. Naturally, it is difficult to guarantee stability during the learning phase. In fact, the use of two different sets parameters to approximate the value function

[1] The exploration conditions are detailed in the next section for a linear-in-the-parameters (LIP) approximation of the value function.

and the policy is motivated by the stability analysis. In particular, to date, the author is unaware of any results that can guarantee stability during learning phase in an online continuous-time deterministic implementation of RL-based actor-critic technique in which only the the value function is approximated, and based on (2–7), the system is controlled using the control law $u = -\frac{1}{2}R^{-1}g^{T}\left(x\right)\left(\nabla\hat{V}\left(x,\hat{W}_{c}\right)\right)^{T}$.

2.6 LIP Approximation of the Value Function

For feasibility of analysis, the optimal value function is approximated using a LIP approximation

$$\hat{V}\left(x,\hat{W}_{c}\right)\triangleq\hat{W}_{c}^{T}\sigma\left(x\right), \tag{2–14}$$

where $\sigma:\mathbb{R}^{n}\to\mathbb{R}^{L}$ is a continuously differentiable nonlinear activation function such that $\sigma\left(0\right)=0$ and $\sigma'\left(0\right)=0$, and $\hat{W}_{c}\in\mathbb{R}^{L}$, where L denotes the number of unknown parameters in the approximation of the value function. Based on (2–7), the optimal policy is approximated using the LIP approximation

$$\hat{u}\left(x,\hat{W}_{a}\right)\triangleq-\frac{1}{2}R^{-1}g\left(x\right)^{T}\nabla\sigma^{T}\left(x\right)\hat{W}_{a}. \tag{2–15}$$

The update law used by the critic to update the weight estimates is given by

$$\begin{aligned}\dot{\hat{W}}_{c}&=-\eta_{c}\Gamma\frac{\omega}{\rho}\delta_{t},\\ \dot{\Gamma}&=\left(\beta\Gamma-\eta_{c}\Gamma\frac{\omega\omega^{T}}{\rho^{2}}\Gamma\right)\mathbf{1}_{\left\{\|\Gamma\|\leq\overline{\Gamma}\right\}},\ \left\|\Gamma\left(t_{0}\right)\right\|\leq\overline{\Gamma},\end{aligned} \tag{2–16}$$

where $\omega\triangleq\nabla\sigma\left(x\right)\dot{x}\in\mathbb{R}^{L}$ denotes the regressor vector, $\rho\triangleq1+\nu\omega^{T}\Gamma\omega\in\mathbb{R}$, $\eta_{c},\beta,\nu\in\mathbb{R}_{>0}$ are constant learning gains, $\overline{\Gamma}\in\mathbb{R}_{>0}$ is a constant saturation constant, and Γ is the least squares gain matrix. The update law used by the actor to update the weight estimates is derived using a Lyapunov-based stability analysis, and is given by

$$\dot{\hat{W}}_{a}=-\eta_{a1}\left(\hat{W}_{a}-\hat{W}_{c}\right)-\eta_{a2}\hat{W}_{a}+\frac{\eta_{c}\nabla\sigma\left(x\right)g\left(x\right)R^{-1}g^{T}\left(x\right)\nabla\sigma^{T}\left(x\right)\hat{W}_{a}\omega^{T}}{4\rho}, \tag{2–17}$$

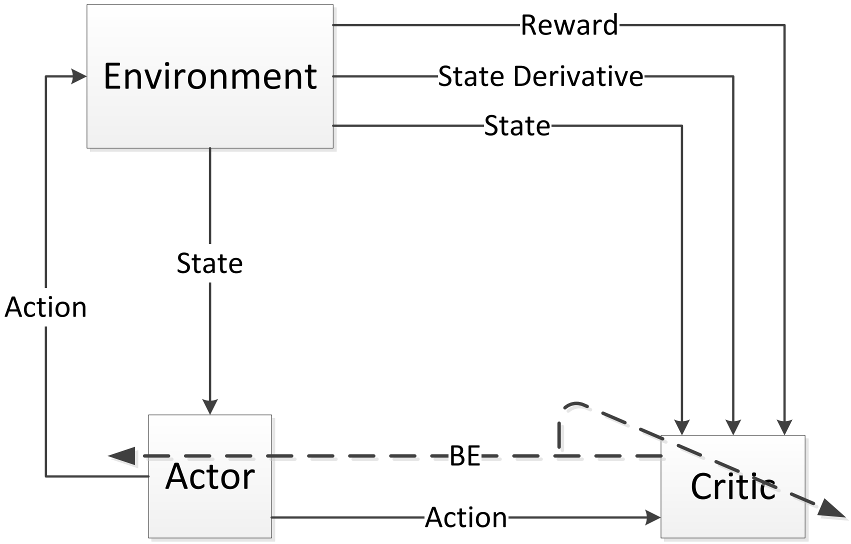

Figure 2-1. Actor-critic architecture

where $\eta_{a1}, \eta_{a2} \in \mathbb{R}_{>0}$ are constant learning gains. A block diagram of the resulting control architecture is presented in Figure 2-1.

The stability analysis indicates that the sufficient exploration condition takes the form of a PE condition that requires the existence of positive constants $\underline{\psi}$ and T such that the regressor vector satisfies

$$\underline{\psi} I_L \leq \int_{t}^{t+T} \frac{\omega(\tau)\,\omega^T(\tau)}{\rho(\tau)} d\tau, \qquad (2\text{–}18)$$

for all $t \in \mathbb{R}_{\geq t_0}$.

Let $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$ denote the vectors of parameter estimation errors, where $W \in \mathbb{R}^L$ denotes the constant vector of ideal parameters. Provided (2–18) is satisfied, and under sufficient conditions on the learning gains and the constants $\underline{\psi}$ and T, the candidate Lyapunov function

$$V_L\left(x, \tilde{W}_c, \tilde{W}_a\right) \triangleq V^*(x) + \frac{1}{2}\tilde{W}_c^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \tilde{W}_a$$

can be used to establish convergence of $x(t)$, $\tilde{W}_c(t)$, and $\tilde{W}_a(t)$ to a neighborhood of zero as $t \to \infty$, when the system in (2–1) is controlled using the control law

$$u(t) = \hat{u}\left(x(t), \hat{W}_a(t)\right), \tag{2–19}$$

and the parameter estimates $\hat{W}_c$ and $\hat{W}_a$ are updated using the update laws in (2–16) and (2–17), respectively.

2.7 Uncertainties in System Dynamics

The use of the state derivative to compute the BE in (2–12) is advantageous because it is easier to obtain a dynamic estimate of the state derivative than it is to identify the system dynamics. For example, consider the high-gain dynamic state derivative estimator

$$\begin{aligned}\dot{\hat{x}} &= g(x)u + k\tilde{x} + \mu, \\ \dot{\mu} &= (k\alpha + 1)\tilde{x},\end{aligned} \tag{2–20}$$

where $\dot{\hat{x}} \in \mathbb{R}^n$ is an estimate of the state derivative, $\tilde{x} \triangleq x - \hat{x}$ is the state estimation error, and $k, \alpha \in \mathbb{R}_{>0}$ are identification gains. Using (2–20), the BE in (2–12) can be approximated by $\hat{\delta}_t$ as

$$\hat{\delta}_t(t) = \nabla_x \hat{V}\left(x(t), \hat{W}_c(t)\right)\dot{\hat{x}}(t) + r(x(t), u(t)).$$

The critic can then learn the value function weights by using an approximation of cumulative experience, quantified by the integral error

$$\hat{E}_t(t) = \int_0^t \hat{\delta}_t^2(\tau)\, d\tau, \tag{2–21}$$

by using $\hat{\delta}_t$ instead of δ_t in (2–16). Under additional sufficient conditions on the gains k and α, the candidate Lyapunov function

$$V_L\left(x, \tilde{W}_c, \tilde{W}_a, \tilde{x}, x_f\right) \triangleq V^*(x) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a + \frac{1}{2}\tilde{x}^T\tilde{x} + \frac{1}{2}x_f^T x_f,$$

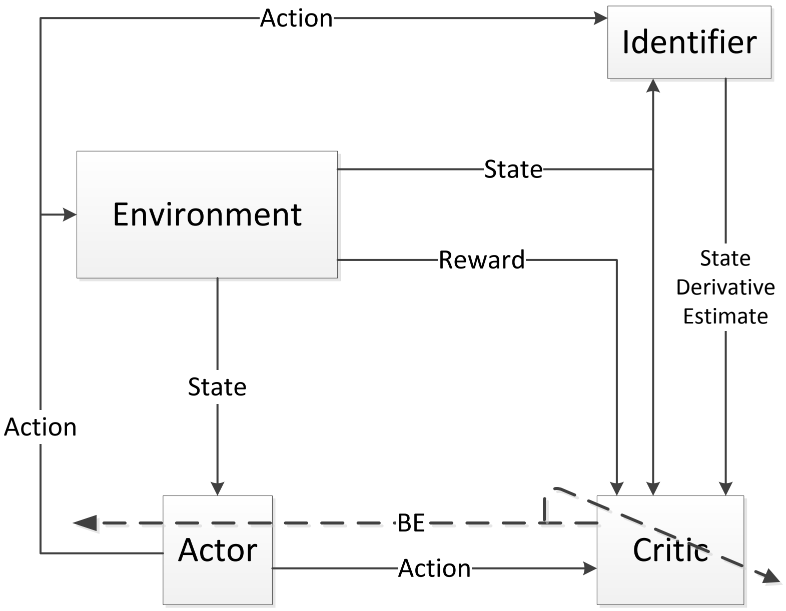

Figure 2-2. Actor-critic-identifier architecture

where $x_f \triangleq \dot{\tilde{x}} + \alpha\tilde{x}$, can be used to establish convergence of $x(t)$, $\tilde{W}_c(t)$, $\tilde{W}_a(t)$, $\tilde{x}$, and x_f to a neighborhood of zero, when the system in (2–1) is controlled using the control law (2–19). This extension of the actor-critic method to handle uncertainties in the system dynamics using derivative estimation is known as the ACI architecture. A block diagram of the ACI architecture is presented in Figure 2-2.

In general, the controller in (2–19) does not ensure the PE condition in (2–18). Thus, in an online implementation, an ad-hoc exploration signal is often added to the controller (cf. [43, 49, 54]). Since the exploration signal is not considered in the the stability analysis, it is difficult the ensure stability of the online implementation. Moreover, the added probing signal causes large control effort expenditure and there is no means to know when it is sufficient to remove the probing signal. The following chapter addresses the challenges associated with the satisfaction of the condition in (2–18) by using simulated experience along with the cumulative experience collected along the system trajectory.

CHAPTER 3
MODEL-BASED REINFORCEMENT LEARNING FOR APPROXIMATE OPTIMAL REGULATION

In this chapter, a CL-based implementation of model-based RL is developed to solve approximate optimal regulation problems online with a relaxed PE-like condition. The development is based on the observation that, given a model of the system, model-based RL can be implemented by evaluating the BE at any number of desired points in the state space. In this result, a parametric system model is considered, and a CL-based parameter identifier is developed to compensate for uncertainty in the parameters. UB regulation of the system states to a neighborhood of the origin, and convergence of the developed policy to a neighborhood of the optimal policy are established using a Lyapunov-based analysis, and simulations are presented to demonstrate the performance of the developed controller.

3.1 Motivation

An ACI architecture to solve optimal regulation problems was presented in Chapter 2, under the restrictive PE requirement in (2–18). The PE requirement is a consequence of the attempt to achieve uniform approximation using information obtained along one system trajectory. In particular, in order to approximate the value function, the critic in the ACI method utilizes experience gained along the system trajectory, quantified by the cumulative observed error in (2–13), instead of the total error in (2–11). The critic in the ACI architecture is restricted to the use of experience gained along the system trajectory because evaluation of the BE requires state derivatives, and the dynamic state-derivative estimator can only estimate state derivatives along the system trajectory.

If the system dynamics are known, or if a system identifier can be developed to estimate the state derivative uniformly over the entire operating domain, then the critic can utilize simulated experience along with gained experience to learn the value function. In particular, the BE in (2–10) can be approximated as

$$\hat{\delta}_X\left(x,\hat{W}_c,\hat{W}_a\right)\triangleq\nabla_x\hat{V}\left(x,\hat{W}_c\right)\dot{X}\left(x,\hat{u}\left(x,\hat{W}_a\right)\right)+r\left(x,\hat{u}\left(x,\hat{W}_a\right)\right),$$

where $\dot{X}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ denotes the estimated dynamics that map the state action pair $\left(x,\hat{u}\left(x,\hat{W}_a\right)\right)$ to the corresponding state derivative. Since the control effectiveness and the control signal in (2–1) are known, a uniform parametric approximation $\hat{f}\left(x,\hat{\theta}\right)$ of the function f, where $\hat{\theta}$ denotes the matrix of parameter estimates, is sufficient to generate a uniform estimate of the system dynamics. In particular, using $\hat{f}$, the BE in (2–10) can be approximated as

$$\hat{\delta}\left(x,\hat{W}_c,\hat{W}_a,\hat{\theta}\right)\triangleq\nabla_x\hat{V}\left(x,\hat{W}_c\right)\left(\hat{f}\left(x,\hat{\theta}\right)+g\left(x\right)\hat{u}\left(x,\hat{W}_a\right)\right)+r\left(x,\hat{u}\left(x,\hat{W}_a\right)\right). \tag{3–1}$$

Similar to Section 2.6, the cumulative gained experience can be quantified using the integral error in (2–21), where $\hat{\delta}_t\left(\tau\right)=\hat{\delta}\left(x\left(\tau\right),\hat{W}_c\left(\tau\right),\hat{W}_a\left(\tau\right),\hat{\theta}\left(\tau\right)\right)$.

Given current parameter estimates $\hat{W}_c\left(t\right)$, $\hat{W}_a\left(t\right)$ and $\hat{\theta}\left(t\right)$, the approximate BE in (3–1) can be evaluated at any point $x_i\in\mathbb{R}^n$. That is, the critic can gain experience on how well the value function is estimated an any arbitrary point x_i in the state space without actually visiting x_i. In other words, given a fixed state x_i and a corresponding planned action $\hat{u}\left(x_i,\hat{W}_a\right)$, the critic can use the estimated drift dynamics $\hat{f}\left(x_i,\hat{W}_a\right)$ to simulate a visit to x_i by computing an estimate of the state derivative at x_i, resulting in simulated experience quantified by the BE $\hat{\delta}_{ti}\left(t\right)=\hat{\delta}\left(x_i,\hat{W}_c\left(t\right),\hat{W}_a\left(t\right),\hat{\theta}\left(t\right)\right)$. The simulated experience can then be used along with gained experience by the critic to learn the value function. The motivation behind using simulated experience is that via selection of multiple (say N) points, the error signal in (2–21) can be augmented to yield a heuristically better approximation $\hat{E}_{ti}\left(t\right)$, given by

$$\hat{E}_{ti}\left(t\right)\triangleq\int\limits_0^t\left(\hat{\delta}_t^2\left(\tau\right)+\sum_{i=1}^N\hat{\delta}_{ti}^2\left(\tau\right)\right)d\tau,$$

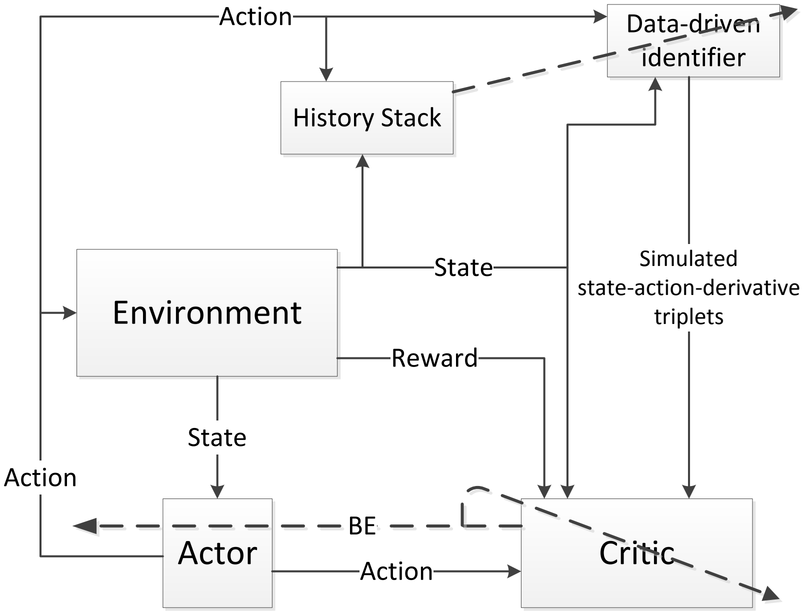

Figure 3-1. Simulation-based actor-critic-identifier architecture

to the desired error signal in (2–11). A block diagram of the simulation-based ACI architecture is presented in Figure (2-2).

Online implementation of simulation of experience requires uniform online estimation of the function f using the parametric approximation $\hat{f}\left(x,\hat{\theta}\right)$, i.e., the parameter estimates $\hat{\theta}$ need to converge to their true values θ. In the following, a system identifier that achieves uniform approximation of f is developed based on recent ideas on data-driven parameter convergence in adaptive control (cf. [92, 93, 147]).

3.2 System Identification

Let $f\left(x^{o}\right)=Y\left(x^{o}\right)\theta$, for all $x^{o}\in\mathbb{R}^{n}$, be a linear parameterization of the function f, where $Y:\mathbb{R}^{n}\to\mathbb{R}^{n\times p}$ is the regression matrix, and $\theta\in\mathbb{R}^{p}$ is the vector of constant unknown parameters.[1] Let $\hat{\theta} \in \mathbb{R}^p$ be an estimate of the unknown parameter vector θ. To estimate the drift dynamics, an identifier is designed as

$$\dot{\hat{x}} = Y(x)\hat{\theta} + g(x)\hat{u} + k_x\tilde{x}, \tag{3–2}$$

where the measurable state estimation error $\tilde{x}$ is defined as $\tilde{x} \triangleq x - \hat{x}$, and $k_x \in \mathbb{R}^{n\times n}$ is a positive definite, constant diagonal observer gain matrix. From (2–1) and (3–2) the identification error dynamics can be derived as

$$\dot{\tilde{x}} = Y(x)\tilde{\theta} - k_x\tilde{x}, \tag{3–3}$$

where $\tilde{\theta}$ is the parameter identification error defined as $\tilde{\theta} \triangleq \theta - \hat{\theta}$.

3.2.1 CL-based Parameter Update

In traditional adaptive control, convergence of the estimates $\hat{\theta}$ to their true values θ is ensured by assuming that a PE condition is satisfied [89–91]. To ensure convergence without the PE condition, this result employs a CL-based approach to update the parameter estimates using recorded input-output data [92, 93, 147].

For ease of exposition, the following system identifier development is based on the assumption that the data required to perform CL-based system identification is available a priori in a history stack. For example, data recorded in a previous run of the system can be utilized, or the history stack can be recorded by running the system using a different known stabilizing controller for a finite amount of time until the recorded data satisfies the rank condition (3–4) detailed in the following assumption.

From a practical perspective, a recorded history stack is unlikely to be available a priori. For such applications, the history stack can be recorded online. Provided

[1] The function f is assumed to be LP for ease of exposition. The system identifier can also be developed using multi-layer NNs for non-LP functions. For example, a system identifier developed using single-layer NNs is presented in Chapter 6.

the system states are exciting over a finite time interval $t \in \left[t_0, t_0 + \bar{t}\right]$ (versus $t \in [t_0, \infty)$ as in traditional PE-based approaches) until the history stack satisfies (3–4), then a modified form of the controller developed in Section 3.3 can be used over the time interval $t \in \left[t_0, t_0 + \bar{t}\right]$, and the controller developed in Section 3.3 can be used thereafter. The required modifications to the controller, and the resulting modifications to the stability analysis are provided in Appendix A.

Assumption 3.1. [92, 93] A history stack $\mathcal{H}_{id}$ containing recorded state-action pairs $\{(x_j, \hat{u}_j) \mid j = 1, \cdots, M\}$, and corresponding numerically computed estimates $\{\dot{\bar{x}}_j \mid j = 1, \cdots, M\}$ of the state derivative $\dot{x}_j \triangleq f(x_j) + g(x_j)\hat{u}_j$ that satisfies

$$\text{rank}\left(\sum_{j=1}^{M} Y_j^T Y_j\right) = p,$$

$$\left\|\dot{\bar{x}}_j - \dot{x}_j\right\| < \bar{d},\ \forall j \tag{3–4}$$

is available a priori, where $Y_j = Y(x_j)$, and $\bar{d} \in \mathbb{R}_{\geq 0}$ is a positive constant.

Based on Assumption 3.1, the update law for the parameter estimates in (3–2) is designed as

$$\dot{\hat{\theta}} = \Gamma_\theta Y(x)^T \tilde{x} + \Gamma_\theta k_\theta \sum_{j=1}^{M} Y_j^T \left(\dot{\bar{x}}_j - g_j\hat{u}_j - Y_j\hat{\theta}\right), \tag{3–5}$$

where $g_j \triangleq g(x_j)$, $\Gamma_\theta \in \mathbb{R}^{p\times p}$ is a constant positive definite adaptation gain matrix, and $k_\theta \in \mathbb{R}$ is a constant positive CL gain. From (2–1) and the definition of $\tilde{\theta}$, the bracketed term in (3–5), can be expressed as $\dot{\bar{x}}_j - g_j\hat{u}_j - Y_j\hat{\theta} = Y_j\tilde{\theta} + d_j$, where $d_j \triangleq \dot{\bar{x}}_j - \dot{x}_j \in \mathbb{R}^n$, and the parameter update law in (3–5) can be expressed in the advantageous form

$$\dot{\hat{\theta}} = \Gamma_\theta Y(x)^T \tilde{x} + \Gamma_\theta k_\theta \left(\sum_{j=1}^{M} Y_j^T Y_j\right)\tilde{\theta} + \Gamma_\theta k_\theta \sum_{j=1}^{M} Y_j^T d_j. \tag{3–6}$$

Even if a history stack is available a priori, the performance of the estimator may be improved by replacing old data with new data. The stability analysis in Section 3.4

allows for a changing history stack through the use of a singular value maximizing algorithm (cf. [93, 147]).

3.2.2 Convergence Analysis

Let $V_0 : \mathbb{R}^{n+p} \to \mathbb{R}_{\geq 0}$ be a positive definite continuously differentiable candidate Lyapunov function defined as

$$V_0(z) \triangleq \frac{1}{2}\tilde{x}^T\tilde{x} + \frac{1}{2}\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}, \tag{3–7}$$

where $z \triangleq \left[\tilde{x}^T, \ \tilde{\theta}^T\right]^T \in \mathbb{R}^{n+p}$. The following bounds on the Lyapunov function can be established:

$$\frac{1}{2}\min\left(1, \underline{\gamma}\right)\|z\|^2 \leq V_0(z) \leq \frac{1}{2}\max\left(1, \overline{\gamma}\right)\|z\|^2, \tag{3–8}$$

where $\underline{\gamma}, \overline{\gamma} \in \mathbb{R}$ denote the minimum and the maximum eigenvalues of the matrix Γ_θ^{-1}.

Using (3–3) and (3–6), the Lyapunov derivative can be expressed as

$$\dot{V}_0 = -\tilde{x}^T k_x \tilde{x} - \tilde{\theta}^T k_\theta \left(\sum_{j=1}^{M} Y_j^T Y_j\right)\tilde{\theta} - k_\theta \tilde{\theta}^T \sum_{j=1}^{M} Y_j^T d_j. \tag{3–9}$$

Let $\underline{y} \in \mathbb{R}$ be the minimum eigenvalue of $\left(\sum_{j=1}^{M} Y_j^T Y_j\right)$. Since $\left(\sum_{j=1}^{M} Y_j^T Y_j\right)$ is symmetric and positive semi-definite, (3–4) can be used to conclude that it is also positive definite, and hence $\underline{y} > 0$. Using (3–8), the Lyapunov derivative in (3–9) can be bounded as

$$\dot{V}_0 \leq -\underline{k_x}\|\tilde{x}\|^2 - \underline{y}k_\theta\left\|\tilde{\theta}\right\|^2 + k_\theta d_\theta\left\|\tilde{\theta}\right\|. \tag{3–10}$$

In (3–10), $d_\theta = \bar{d}\sum_{j=1}^{M}\|Y_j\|$, and $\underline{k_x} \in \mathbb{R}$ denotes the minimum eigenvalue of the matrix k_x. The inequalities in (3–8) and (3–10) can be used to conclude that $\left\|\tilde{\theta}\right\|$ and $\|\tilde{x}\|$ exponentially decay to an ultimate bound as $t \to \infty$.

The CL-based observer results in exponential regulation of the parameter and the state derivative estimation errors to a neighborhood around the origin. In the following, the parameter and state derivative estimates are used to approximately solve the HJB equation without the knowledge of the drift dynamics.

3.3 Approximate Optimal Control

3.3.1 Value Function Approximation

Approximations to the optimal value function V^* and the optimal policy u^* are designed based on NN-based representations. A single layer NN can be used to represent the optimal value function V^* as

$$V^*\left(x^o\right) = W^T\sigma\left(x^o\right) + \epsilon\left(x^o\right), \tag{3–11}$$

for all $x^o \in \mathbb{R}^n$, where $W \in \mathbb{R}^L$ is the ideal weight matrix and $\sigma : \mathbb{R}^n \to \mathbb{R}^L$ and $\epsilon : \mathbb{R}^n \to \mathbb{R}$ are introduced in (2–14).

Based on (3–11) a NN-based representation of the optimal controller is derived as

$$u^*\left(x^o\right) = -\frac{1}{2}R^{-1}g^T\left(x^o\right)\left(\nabla\sigma^T\left(x^o\right)W + \nabla\epsilon^T\left(x^o\right)\right), \tag{3–12}$$

for all $x^o \in \mathbb{R}^n$. The NN-based approximations $\hat{V} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$ of the optimal value function in (3–11) and $\hat{u} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$ of the optimal policy in (3–12) are given by (2–14) and (2–15), respectively, where $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ are estimates of the ideal weights W. The use of two sets of weights to estimate the same set of ideal weights is motivated by the stability analysis and the fact that it enables a formulation of the BE that is linear in the value function weight estimates $\hat{W}_c$, enabling a least squares-based adaptive update law. Using the parametric estimates $\hat{V}$ and $\hat{u}$ of the value function and the policy from (2–14) and (2–15), respectively, and using the system identifier developed in Section 3.2, the BE in (3–1) can be expressed as

$$\hat{\delta}_t = \omega^T\hat{W}_c + x^TQx + \hat{u}^T\left(x, \hat{W}_a\right)R\hat{u}\left(x, \hat{W}_a\right),$$

where $\omega \in \mathbb{R}^L$ is the regressor vector defined as $\omega \triangleq \nabla\sigma\left(x\right)\left(Y\left(x\right)\hat{\theta} + g\left(x\right)\hat{u}\left(x, \hat{W}_a\right)\right)$.

3.3.2 Simulation of Experience via BE Extrapolation

In traditional RL-based algorithms, the value function estimate and the policy estimate are updated based on observed data. The use of observed data to learn the value function naturally leads to a sufficient exploration condition which demands sufficient richness in the observed data. In stochastic systems, this is achieved using a randomized stationary policy (cf. [43, 48, 49]), whereas in deterministic systems, a probing noise is added to the derived control law (cf. [56, 57, 59, 114, 115]). The technique developed in this result implements simulation of experience in a model-based RL scheme by using $Y\hat{\theta}$ as an estimate of the uncertain drift dynamics f to extrapolate the approximate BE to unexplored areas of the state space. The following rank condition enables the extrapolation of the approximate BE to a predefined set of points $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ in the state space.

Assumption 3.2. There exists a finite set of points $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ such that

$$0 < \underline{c} \triangleq \frac{1}{N}\left(\inf_{t\in\mathbb{R}_{\geq t_0}}\left(\lambda_{\min}\left\{\sum_{i=1}^{N}\frac{\omega_i\omega_i^T}{\rho_i}\right\}\right)\right), \tag{3–13}$$

where $\lambda_{min}\{\cdot\}$ denotes the minimum eigenvalue. In (3–13), $\rho_i \triangleq 1 + \nu\omega_i^T\Gamma\omega_i \in \mathbb{R}$ are normalization terms, where $\nu \in \mathbb{R}$ is a constant positive normalization gain, $\Gamma \in \mathbb{R}^{L\times L}$ is a time-varying least-squares gain matrix, $\mathbb{R}_{\geq t_0} \triangleq [t_0, \infty)$, and $\omega_i \triangleq \nabla\sigma(x_i)\left(Y(x_i)\hat{\theta} + g(x_i)\hat{u}\left(x_i, \hat{W}_a\right)\right).$

The rank condition in (3–13) depends on the estimates $\hat{\theta}$ and $\hat{W}_a$; hence, in general, it is impossible to guarantee a priori. However, unlike the PE condition in previous results such as [56, 57, 59, 114, 115], the condition in (3–13) can be verified online at each time t. Furthermore, the condition in (3–13) can be heuristically met by collecting redundant data, i.e., by selecting more points than the number of neurons by choosing $N \gg L$.

To simulate experience, the approximate BE is evaluated at the sampled points $\{x_i \mid i = 1, \cdots, N\}$ as

$$\hat{\delta}_{ti} = \omega_i^T \hat{W}_c + x_i^T Q x_i + \hat{u}^T \left(x_i, \hat{W}_a\right) R \hat{u} \left(x_i, \hat{W}_a\right).$$

For notational brevity, the dependence of the functions f, Y, g, σ, ϵ, $\hat{u}$, $\hat{u}_i$, $\hat{\delta}_t$, and $\hat{\delta}_{ti}$, on the state, time, and the weights is suppressed hereafter. A CL-based least-squares update law for the value function weights is designed based on the subsequent stability analysis as

$$\begin{aligned} \dot{\hat{W}}_c &= -\eta_{c1} \Gamma \frac{\omega}{\rho} \hat{\delta}_t - \frac{\eta_{c2}}{N} \Gamma \sum_{i=1}^{N} \frac{\omega_i}{\rho_i} \hat{\delta}_{ti}, \\ \dot{\Gamma} &= \left(\beta \Gamma - \eta_{c1} \Gamma \frac{\omega \omega^T}{\rho^2} \Gamma\right) \mathbf{1}_{\left\{\|\Gamma\| \leq \overline{\Gamma}\right\}}, \ \|\Gamma\left(t_0\right)\| \leq \overline{\Gamma}, \end{aligned} \tag{3–14}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function, $\overline{\Gamma} \in \mathbb{R}_{>0}$ is the saturation constant, $\beta \in \mathbb{R}_{>0}$ is the forgetting factor, and $\eta_{c1}, \eta_{c2} \in \mathbb{R}_{>0}$ are constant adaptation gains. The update law in (3–14) ensures that the adaptation gain matrix is bounded such that

$$\underline{\Gamma} \leq \|\Gamma\left(t\right)\| \leq \overline{\Gamma}, \ \forall t \in \mathbb{R}_{\geq t_0}, \tag{3–15}$$

where $\underline{\Gamma} \in \mathbb{R}_{>0}$ is a constant. The policy weights are then updated to follow the value function weights as[2]

$$\dot{\hat{W}}_a = -\eta_{a1} \left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2} \hat{W}_a + \left(\frac{\eta_{c1} G_\sigma^T \hat{W}_a \omega^T}{4\rho} + \sum_{i=1}^{N} \frac{\eta_{c2} G_{\sigma i}^T \hat{W}_a \omega_i^T}{4N\rho_i}\right) \hat{W}_c, \tag{3–16}$$

[2] Using the fact that the ideal weights are bounded, a projection-based (cf. [148]) update law $\dot{\hat{W}}_a = proj \left\{-\eta_{a1} \left(\hat{W}_a - \hat{W}_c\right)\right\}$ can be utilized to update the policy weights. Since the policy weights are bounded a priori by the projection algorithm, a less complex stability analysis can be used to establish the result in Theorem 3.1.

where $\eta_{a1}, \eta_{a2} \in \mathbb{R}$ are positive constant adaptation gains, and $G_\sigma \triangleq \nabla\sigma g R^{-1} g^T \nabla\sigma^T \in \mathbb{R}^{L\times L}$.

The update law in (3–14) is fundamentally different from the CL-based adaptive update in results such as [92, 93], in the sense that the points $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$ are selected a priori based on prior information about the desired behavior of the system, and using an estimate of the system dynamics, the approximate BE is evaluated at $\{x_i \in \mathbb{R}^n \mid i = 1, \cdots, N\}$. In the CL-based adaptive update in results such as [92, 93], the prediction error is used as a metric for learning. The prediction error depends on measured or numerically computed values of the state derivative; hence, the prediction error can only be evaluated at observed data points along the state trajectory.

3.4 Stability Analysis

To facilitate the subsequent stability analysis, the approximate BE is expressed in terms of the weight estimation errors $\tilde{W}_c$ and $\tilde{W}_a$ as

$$\hat{\delta}_t = -\omega^T \tilde{W}_c - W^T \nabla\sigma Y \tilde{\theta} + \frac{1}{4}\tilde{W}_a^T G_\sigma \tilde{W}_a + \frac{1}{4} G_\epsilon - \nabla\epsilon f + \frac{1}{2} W^T \nabla\sigma G \nabla\epsilon^T, \tag{3–17}$$

where $G \triangleq g R^{-1} g^T \in \mathbb{R}^{n\times n}$ and $G_\epsilon \triangleq \nabla\epsilon G \nabla\epsilon^T \in \mathbb{R}$. Similarly, the approximate BE evaluated at the sampled states $\{x_i \mid i = 1, \cdots, N\}$ can be expressed as

$$\hat{\delta}_{ti} = -\omega_i^T \tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i} \tilde{W}_a - W^T \nabla\sigma_i Y_i \tilde{\theta} + \Delta_i, \tag{3–18}$$

where $Y_i = Y(x_i)$, and $\Delta_i \triangleq \frac{1}{2} W^T \nabla\sigma_i G_i \nabla\epsilon_i^T + \frac{1}{4} G_{\epsilon i} - \nabla\epsilon_i f_i \in \mathbb{R}$ is a constant.

Let $\mathcal{Z} \subset \mathbb{R}^{2n+2L+p}$ denote a compact set, and let $\chi \triangleq \mathcal{Z} \cap \mathbb{R}^n$. On the compact set $\chi \subset \mathbb{R}^n$ the function Y is Lipschitz continuous; hence, there exists a positive constant

$L_Y \in \mathbb{R}$ such that[3]

$$\left\| Y\left(x\right)\right\| \leq L_Y \left\| x\right\|, \forall x \in \chi. \tag{3–19}$$

Furthermore, using the universal function approximation property, the ideal weight matrix $W \in \mathbb{R}^L$, is bounded above by a known positive constant $\overline{W}$ in the sense that $\left\| W\right\| \leq \overline{W}$ and the function reconstruction error $\epsilon : \mathbb{R}^n \to \mathbb{R}$ is uniformly bounded over χ such that $\sup_{x^o \in \chi} \left|\epsilon\left(x^o\right)\right| \leq \bar{\epsilon}$ and $\sup_{x^o \in \chi} \left|\nabla\epsilon\left(x^o\right)\right| \leq \overline{\nabla\epsilon}$. Using (3–15), the normalized regressor $\frac{\omega}{\rho}$ can be bounded as

$$\sup_{t \in \mathbb{R}_{\geq t_0}} \left\| \frac{\omega\left(t\right)}{\rho\left(t\right)} \right\| \leq \frac{1}{2\sqrt{\nu\underline{\Gamma}}}. \tag{3–20}$$

For brevity of notation, for a function $\xi : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, define the operator $\overline{\left(\cdot\right)} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as $\overline{\xi} \triangleq \sup_{x^o \in \chi} \xi\left(x^o\right)$, and the following positive constants:

$$\vartheta_1 \triangleq \frac{\eta_{c1} L_Y \left\|\theta\right\| \overline{\nabla\epsilon}}{4\sqrt{\nu\underline{\Gamma}}}, \quad \vartheta_2 \triangleq \sum_{i=1}^{N} \left(\frac{\eta_{c2} \left\| \nabla\sigma_i Y_i \right\| \overline{W}}{4N\sqrt{\nu\underline{\Gamma}}} \right), \quad \vartheta_3 \triangleq \frac{L_Y \eta_{c1} \overline{W} \overline{\left\|\nabla\sigma\right\|}}{4\sqrt{\nu\underline{\Gamma}}}, \quad \vartheta_4 \triangleq \overline{\left\| \frac{1}{4} G_\epsilon \right\|},$$

$$\vartheta_5 \triangleq \frac{\eta_{c1} \overline{\left\| 2W^T \nabla\sigma G \nabla\epsilon^T + G_\epsilon \right\|}}{8\sqrt{\nu\underline{\Gamma}}} + \left\| \sum_{i=1}^{N} \frac{\eta_{c2} \omega_i \Delta_i}{N\rho_i} \right\|, \quad \vartheta_7 \triangleq \frac{\eta_{c1} \overline{\left\| G_\sigma \right\|}}{8\sqrt{\nu\underline{\Gamma}}} + \sum_{i=1}^{N} \left(\frac{\eta_{c2} \left\| G_{\sigma i} \right\|}{8N\sqrt{\nu\underline{\Gamma}}} \right),$$

$$\vartheta_6 \triangleq \overline{\left\| \frac{1}{2} W^T G_\sigma + \frac{1}{2} \nabla\epsilon G^T \nabla\sigma^T \right\|} + \vartheta_7 \overline{W}^2 + \eta_{a2} \overline{W}, \quad \underline{q} \triangleq \lambda_{min}\{Q\},$$

$$v_l = \frac{1}{2} \min\left(\frac{\underline{q}}{2}, \frac{\eta_{c2}\underline{c}}{3}, \frac{\eta_{a1} + 2\eta_{a2}}{6}, \underline{k_x}, \frac{k_\theta \underline{y}}{4} \right), \quad \iota = \frac{3\vartheta_5^2}{4\eta_{c2}\underline{c}} + \frac{3\vartheta_6^2}{2\left(\eta_{a1} + 2\eta_{a2}\right)} + \frac{k_\theta d_\theta^2}{2\underline{y}} + \vartheta_4. \tag{3–21}$$

To facilitate the stability analysis, let $V_L : \mathbb{R}^{2n+2L+p} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a continuously differentiable positive definite candidate Lyapunov function defined as

$$V_L\left(Z, t\right) \triangleq V^*\left(x\right) + \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2} \tilde{W}_a^T \tilde{W}_a + V_0\left(z\right), \tag{3–22}$$

[3] The Lipschitz property is exploited here for clarity of exposition. The bound in (3–19) can be easily generalized to $\left\| Y\left(x\right)\right\| \leq L_Y\left(\left\|x\right\|\right)\left\|x\right\|$, where $L_Y : \mathbb{R} \to \mathbb{R}$ is a positive, non-decreasing function.

where V^{*} is the optimal value function, V_0 was introduced in (3–7) and

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

Using the fact that V^{*} is positive definite, (3–8), (3–15) and Lemma 4.3 from [149] yield

$$\underline{v}\left(\|Z^{o}\|\right)\leq V_{L}\left(Z^{o},t\right)\leq\overline{v}\left(\|Z^{o}\|\right), \tag{3–23}$$

for all $t\in\mathbb{R}_{\geq t_0}$ and for all $Z^{o}\in\mathbb{R}^{2n+2L+p}$. In (3–23), $\underline{v},\overline{v}:\mathbb{R}_{\geq0}\to\mathbb{R}_{\geq0}$ are class $\mathcal{K}$ functions.

The sufficient conditions for UB convergence are derived based on the subsequent stability analysis as

$$\frac{\eta_{a1}+2\eta_{a2}}{6}>\vartheta_{7}\overline{W}\left(\frac{2\zeta_{2}+1}{2\zeta_{2}}\right),\quad\frac{k_{\theta}}{4}>\frac{\vartheta_{2}+\zeta_{1}\zeta_{3}\vartheta_{3}\overline{Z}}{\underline{y}\zeta_{1}},\quad\frac{\underline{q}}{2}>\vartheta_{1},$$
$$\frac{\eta_{c2}}{3}>\frac{\zeta_{2}\vartheta_{7}\overline{W}+\eta_{a1}+2\left(\vartheta_{1}+\zeta_{1}\vartheta_{2}+\left(\vartheta_{3}/\zeta_{3}\right)\overline{Z}\right)}{2\underline{c}}, \tag{3–24}$$

$$\sqrt{\frac{\iota}{v_{l}}}\leq r, \tag{3–25}$$

where $\overline{Z}\triangleq\underline{v}^{-1}\left(\overline{v}\left(\max\left(\|Z\left(t_{0}\right)\|,\sqrt{\frac{\iota}{v_{l}}}\right)\right)\right)$, $r\in\mathbb{R}_{\geq0}$ denotes the radius of the set $\mathcal{Z}$ defined as $r\triangleq\frac{1}{2}\sup\left\{\|x-y\|\mid x,y\in\mathcal{Z}\right\}$, and $\zeta_{1},\zeta_{2},\zeta_{3}\in\mathbb{R}$ are known positive adjustable constants. The Lipschitz constants in (3–19) and the NN function approximation errors in (3–11) depend on the underlying compact set; hence, given a bound on the initial condition $Z\left(t_{0}\right)$ for the concatenated state Z, a compact set that contains the concatenated state trajectory needs to be established before adaptation gains satisfying the conditions in (3–24) can be selected. In the following, based on the subsequent stability analysis, an algorithm is developed to compute the required compact set, denoted by $\mathcal{Z}\subset\mathbb{R}^{2n+2L+p}$. In Algorithm 3.1, the notation $\left\{(\cdot)\right\}_{i}$ denotes the value of $(\cdot)$ computed in the i^{th} iteration. Since the constants ι and v_{l} depend on L_{Y} only through the products $L_{Y}\overline{\nabla\epsilon}$ and $\frac{L_{Y}}{\zeta_{3}}$, Algorithm 3.1 ensures the satisfaction of the sufficient condition in (3–25). The main result of this chapter can now be stated as follows.

Algorithm 3.1 Gain Selection

First iteration:

Given $\overline{z} \in \mathbb{R}_{\geq 0}$ such that $\|Z(t_0)\| < \overline{z}$, let $\mathcal{Z}_1 \triangleq \left\{\xi \in \mathbb{R}^{2n+2L+p} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(\overline{z}\right)\right)\right\}$. Using $\mathcal{Z}_1$, compute the bounds in (3–21) and select the gains according to (3–24). If $\left\{\sqrt{\frac{\iota}{v_l}}\right\}_1 \leq \overline{z}$, set $\mathcal{Z} = \mathcal{Z}_1$ and terminate.

Second iteration:

If $\overline{z} < \left\{\sqrt{\frac{\iota}{v_l}}\right\}_1$, let $\mathcal{Z}_2 \triangleq \left\{\xi \in \mathbb{R}^{2n+2L+p} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(\left\{\sqrt{\frac{\iota}{v_l}}\right\}_1\right)\right)\right\}$. Using $\mathcal{Z}_2$, compute the bounds in (3–21) and select the gains according to (3–24). If $\left\{\frac{\iota}{v_l}\right\}_2 \leq \left\{\frac{\iota}{v_l}\right\}_1$, set $\mathcal{Z} = \mathcal{Z}_2$ and terminate.

Third iteration:

If $\left\{\frac{\iota}{v_l}\right\}_2 > \left\{\frac{\iota}{v_l}\right\}_1$, increase the number of NN neurons to $\{L\}_3$ to ensure $\{L_Y\}_2\left\{\overline{\nabla\epsilon}\right\}_3 \leq \{L_Y\}_2\left\{\overline{\nabla\epsilon}\right\}_2, \forall i = 1,..,N$, increase the constant ζ_3 to ensure $\frac{\{L_Y\}_2}{\{\zeta_3\}_3} \leq \frac{\{L_Y\}_2}{\{\zeta_3\}_2}$, and increase the gain k_θ to satisfy the gain conditions in (3–24). These adjustments ensure $\{\iota\}_3 \leq \{\iota\}_2$. Set $\mathcal{Z} = \left\{\xi \in \mathbb{R}^{2n+2L+p} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(\left\{\sqrt{\frac{\iota}{v_l}}\right\}_2\right)\right)\right\}$ and terminate.

Theorem 3.1. *Provided Assumptions* (3.1) - (3.2) *hold and gains $\underline{q}$, η_{c2}, η_{a2}, and k_θ are selected large enough using Algorithm 3.1, the observer in* (3–2) *along with the adaptive update law in* (3–5) *and the controller in* (2–15) *along with the adaptive update laws in* (3–14) *and* (3–16) *ensure that the state x, the state estimation error $\tilde{x}$, the value function weight estimation error $\tilde{W}_c$ and the policy weight estimation error $\tilde{W}_a$ are UB.*

Proof. The time derivative of (3–22) along the trajectories of (2–1), (3–3), (3–6), (3–14), and (3–16) is given by

$$\begin{aligned}\dot{V}_L = {} & \nabla V\left(f + g\hat{u}\right) - \tilde{W}_c^T\left(-\eta_{c1}\frac{\omega}{\rho}\hat{\delta}_t - \frac{\eta_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i}{\rho_i}\hat{\delta}_{ti}\right) - \tilde{W}_a^T\left(-\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a\right) \\ & - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\left(\beta\Gamma - \eta_{c1}\left(\Gamma\frac{\omega\omega^T}{\rho^2}\Gamma\right)\right)\Gamma^{-1}\tilde{W}_c - \tilde{x}^Tk_x\tilde{x} - k_\theta\tilde{\theta}^T\left(\sum_{j=1}^{N}Y_j^TY_j\right)\tilde{\theta} - k_\theta\tilde{\theta}^T\sum_{j=1}^{M}Y_j^Td_j \\ & - \tilde{W}_a^T\left(\frac{\eta_{c1}G_\sigma^T\hat{W}_a\omega^T}{4\rho} + \sum_{i=1}^{N}\frac{\eta_{c2}G_{\sigma i}^T\hat{W}_a\omega_i^T}{4N\rho_i}\right)\hat{W}_c, \end{aligned} \quad (3\text{–}26)$$

Substituting for the approximate BEs from (3–17) and (3–18), using the bounds in (3–19) and (3–20), and using Young's inequality, the Lyapunov derivative in (3–26) can be upper-bounded as

$$\begin{aligned}\dot{V}_L \leq & -\frac{\underline{q}}{2}\left\|x\right\|^2 - \frac{\eta_{c2}\underline{c}}{3}\left\|\tilde{W}_c\right\|^2 - \frac{\eta_{a1}+2\eta_{a2}}{6}\left\|\tilde{W}_a\right\|^2 - \underline{k_x}\left\|\tilde{x}\right\|^2 - \frac{k_\theta \underline{y}}{4}\left\|\tilde{\theta}\right\|^2 - \left(\frac{\underline{q}}{2} - \vartheta_1\right)\left\|x\right\|^2 \\ & - \left(\frac{\eta_{c2}\underline{c}}{3} - \vartheta_1 + \zeta_1\vartheta_2 + \frac{\zeta_2\vartheta_7\overline{W} + \eta_{a1}}{2} - \frac{\vartheta_3\left\|x\right\|}{\zeta_3}\right)\left\|\tilde{W}_c\right\|^2 - \left(\left(\frac{k_\theta \underline{y}}{4} - \frac{\vartheta_2}{\zeta_1}\right) - \vartheta_3\zeta_3\left\|x\right\|\right)\left\|\tilde{\theta}\right\|^2 \\ & - \left(\frac{\eta_{a1}+2\eta_{a2}}{6} - \vartheta_7\left\|W\right\| - \frac{\vartheta_7\left\|W\right\|}{2\zeta_2}\right)\left\|\tilde{W}_a\right\|^2 + \frac{3\vartheta_5^2}{4\eta_{c2}\underline{c}} + \frac{3\vartheta_6^2}{2\left(\eta_{a1}+2\eta_{a2}\right)} + \frac{k_\theta d_\theta^2}{2\underline{y}} + \frac{1}{4}G_\epsilon.\end{aligned} \quad (3–27)$$

Provided the gains are selected based using Algorithm 3.1, the Lyapunov derivative in (3–27) can be upper-bounded as

$$\dot{V}_L \leq -v_l\left\|Z\right\|^2, \quad \forall\left\|Z\right\| \geq \sqrt{\frac{\iota}{v_l}} > 0, \quad (3–28)$$

for all $t \geq 0$ and $\forall Z \in \mathcal{Z}$. Using (3–23), (3–25) and (3–28), Theorem 4.18 in [149] can now be invoked to conclude that Z is UB in the sense that $\limsup_{t\to\infty}\left\|Z\left(t\right)\right\| \leq \underline{v}^{-1}\left(\overline{v}\left(\sqrt{\frac{\iota}{v_l}}\right)\right)$. Furthermore, the concatenated state trajectories are bounded such that $\left\|Z\left(t\right)\right\| \leq \overline{Z}$ for all $t \in \mathbb{R}_{\geq t_0}$. Since the estimates $\hat{W}_a$ approximate the ideal weights W, the definitions in (3–12) and (2–15) can be used to conclude that the policy $\hat{u}$ approximates the optimal policy u^*.[4] □

3.5 Simulation

This section presents two simulations to demonstrate the performance and the applicability of the developed technique. First, the performance of the developed controller is demonstrated through approximate solution of an optimal control problem that has a known analytical solution. Based on the known solution, an exact polynomial basis is used for value function approximation. The second simulation demonstrates

[4] If $\mathcal{H}_{id}$ is updated with new data, (3–3) and (3–6) form a switched system. Provided $\mathcal{H}_{id}$ is updated using a singular value maximizing algorithm, (3–28) can be used to establish that V_L is a common Lyapunov function for the switched system (cf. [93]).

the applicability of the developed technique in the case where the analytical solution, and hence, the basis for value function approximation is unknown. In this case, since the optimal solution is unknown, the optimal trajectories obtained using the developed technique are compared with optimal trajectories obtained through a numerical optimal control technique.

3.5.1 Problem with a Known Basis

The performance of the developed controller is demonstrated by simulating a nonlinear, control-affine system with a two dimensional state $x = [x_1, \ x_2]^T$. The system dynamics are described by (2–1), where [57]

$$f = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1 & x_2\left(1-(cos\left(2x_1\right)+2)^2\right) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ cos\left(2x_1\right)+2 \end{bmatrix} \tag{3–29}$$

where $a, b, c, d \in \mathbb{R}$ are positive unknown parameters. The parameters are selected as[5] $a = -1$, $b = 1$, $c = -0.5$, and $d = -0.5$. The control objective is to minimize the cost in (2–4), where $Q = I_{2\times 2}$ and $R = 1$. The optimal value function and optimal control for the system in (3–29) are given by $V^*(x) = \frac{1}{2}x_1^2 + x_2^2$, and $u^*(x) = -(cos(2x_1)+2)x_2$ (cf. [57]).

To facilitate the identifier design, thirty data points are recorded using a singular value maximizing algorithm (cf. [93]) for the CL-based adaptive update law in (3–5). The state derivative at the recorded data points is computed using a fifth order Savitzky-Golay smoothing filter (cf. [150]).

To facilitate the ADP-based controller, the basis function $\sigma : \mathbb{R}^2 \to \mathbb{R}^3$ for value function approximation is selected as $\sigma = \begin{bmatrix} x_1^2, & x_1x_2, & x_2^2 \end{bmatrix}$. Based on the analytical solution, the ideal weights are $W = [0.5, \ 0, \ 1]^T$. The data points for the CL-based update

[5] The origin is an unstable equilibrium point of the unforced system $\dot{x} = f\left(x\right)$.

law in (3–14) are selected to be on a 5×5 grid on a 2×2 square around the origin. The learning gains are selected as $\eta_{c1} = 1$, $\eta_{c2} = 15$, $\eta_{a1} = 100$, $\eta_{a2} = 0.1$, $\nu = 0.005$, $k_x = 10I_{2\times2}$, $\Gamma_\theta = 20I_{4\times4}$, and $k_\theta = 30$. The policy and the value function weight estimates are initialized using a stabilizing set of initial weights as $\hat{W}_c(0) = \hat{W}_a(0) = [1,\ 1,\ 1]^T$ and the least squares gain is initialized as $\Gamma(0) = 100I_{3\times3}$. The initial condition for the system state is selected as $x(0) = [-1,\ -1]^T$, the state estimates $\hat{x}$ are initialized to be zero, the parameter estimates $\hat{\theta}$ are initialized to be one , and the history stack for CL is recorded online.

Figures 3-2 - 3-4 demonstrates that the system state is regulated to the origin, the unknown parameters in the drift dynamics are identified, and the value function and the policy weights converge to their true values. Furthermore, unlike previous results, an ad-hoc probing signal to ensure PE is not required.

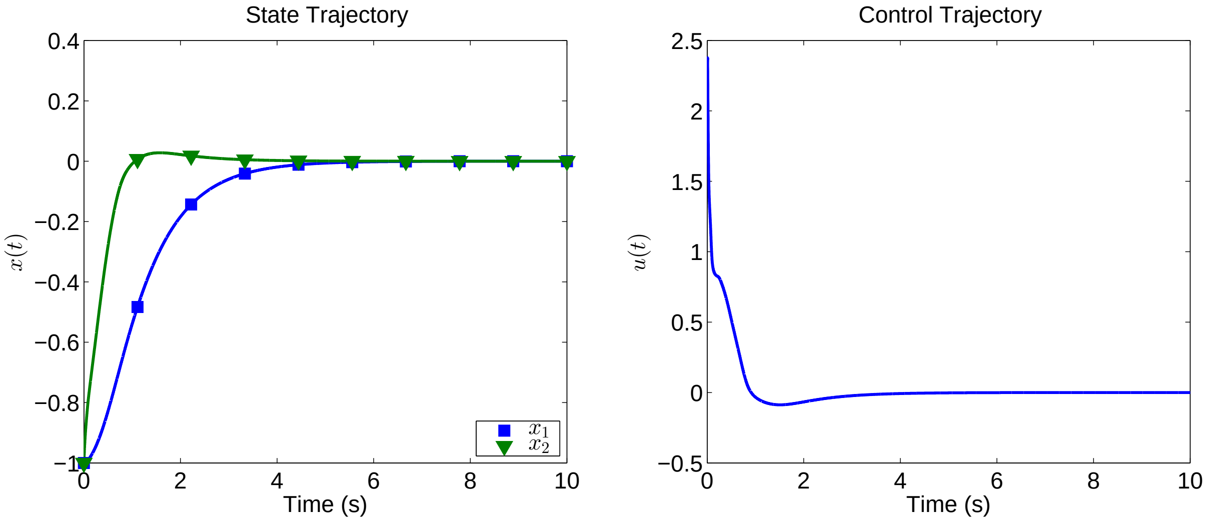

Figure 3-2. System state and control trajectories generated using the developed method for the system in Section 3.5.1.

3.5.2 Problem with an Unknown Basis

To demonstrate the applicability of the developed controller, a nonlinear, control-affine system with a four dimensional state $x = [x_1,\ x_2,\ x_3,\ x_4]^T$ is simulated. The system

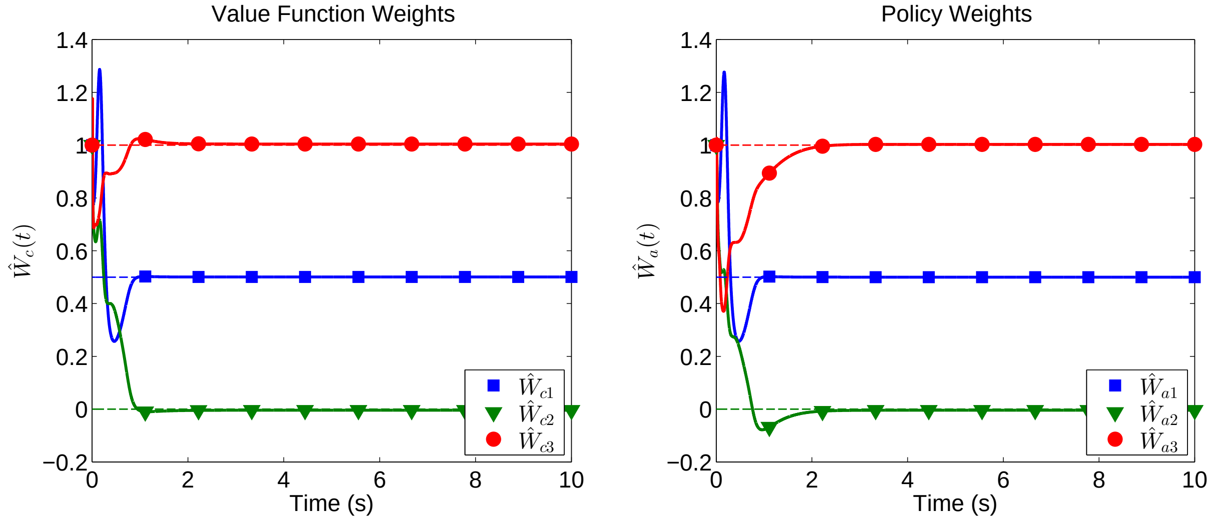

Figure 3-3. Actor and critic weight trajectories generated using the developed method for the system in Section 3.5.1 compared with their true values. The true values computed based on the analytical solution are represented by dotted lines.

dynamics are described by (2–1), where

$$f=\begin{bmatrix} x_3\\ x_4\\ -M^{-1}V_m\begin{bmatrix} x_3\\ x_4\end{bmatrix}\end{bmatrix}+\begin{bmatrix} 0, & 0, & 0, & 0\\ 0, & 0, & 0, & 0\\ \multicolumn{4}{c}{\begin{bmatrix} M^{-1}, & M^{-1}\end{bmatrix}D}\end{bmatrix}\begin{bmatrix} f_{d1}\\ f_{d2}\\ f_{s1}\\ f_{s2}\end{bmatrix},$$

$$g=\left[\begin{bmatrix}0, & 0\end{bmatrix}^T,\ \begin{bmatrix}0, & 0\end{bmatrix}^T,\ \left(M^{-1}\right)^T\right]^T. \tag{3–30}$$

In (3–30), $D\triangleq diag\left[x_3,\ x_4,\ tanh\left(x_3\right),\ tanh\left(x_4\right)\right]$ and the matrices $M,\ V_m,\ F_d,\ F_s\in\mathbb{R}^{2\times 2}$ are defined as $M\triangleq\begin{bmatrix} p_1+2p_3c_2, & p_2+p_3c_2\\ p_2+p_3c_2, & p_2\end{bmatrix}$, $F_d\triangleq\begin{bmatrix} f_{d1}, & 0\\ 0, & f_{d2}\end{bmatrix}$, $V_m\triangleq\begin{bmatrix} -p_3s_2x_4, & -p_3s_2\left(x_3+x_4\right)\\ p_3s_2x_3, & 0\end{bmatrix}$, and $F_s\triangleq\begin{bmatrix} f_{s1}tanh\left(x_3\right), & 0\\ 0, & f_{s2}tanh\left(x_3\right)\end{bmatrix}$, where $c_2=cos\left(x_2\right),\ s_2=sin\left(x_2\right)$, $p_1=3.473$, $p_2=0.196$, and $p_3=0.242$, and $f_{d1},\ f_{d2},\ f_{s1},\ f_{s2}\in\mathbb{R}$ are positive unknown parameters. The parameters are selected as $f_{d1}=5.3$,

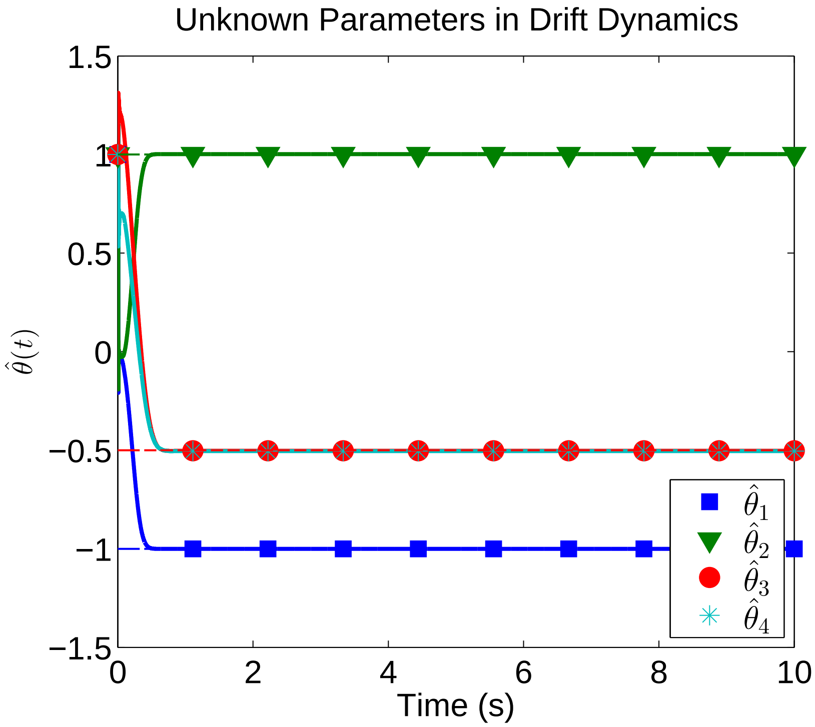

Figure 3-4. Drift parameter estimate trajectories generated using the developed method for the system in Section 3.5.1 compared to the actual drift parameters. The dotted lines represent true values of the drift parameters.

$f_{d2} = 1.1$, $f_{s1} = 8.45$, and $f_{s2} = 2.35$. The control objective is to minimize the cost in (2–4), where $Q = diag\left([10,\ 10,\ 1,\ 1]\right)$ and $R = diag\left([1,\ 1]\right)$.

To facilitate the ADP-based controller, the basis function $\sigma : \mathbb{R}^4 \to \mathbb{R}^{10}$ for value function approximation is selected as

$$\sigma(x) = \left[\begin{array}{cccccccccc} x_1x_3, & x_2x_4, & x_3x_2, & x_4x_1, & x_1x_2, & x_4x_3, & x_1^2, & x_2^2, & x_3^2, & x_4^2 \end{array}\right].$$

The data points for the CL-based update law in (3–14) are selected to be on a $3\times3\times3\times3$ grid around the origin, and the policy weights are updated using a projection-based update law. The learning gains are selected as $\eta_{c1} = 1$, $\eta_{c2} = 30$, $\eta_{a1} = 0.1$, $\nu = 0.0005$, $k_x = 10I_4$, $\Gamma_\theta = diag([90,\ 50,\ 160,\ 50])$, and $k_\theta = 1.1$. The least squares gain is initialized as $\Gamma(0) = 1000I_{10}$ and the policy and the value function weight estimates are initialized as $\hat{W}_c(0) = \hat{W}_a(0) = [5,\ 5,\ 0,\ 0,\ 0,\ 0,\ 25,\ 0,\ 2\ ,2]^T$. The initial condition for the system state is selected as $x(0) = [1,\ 1,\ 0,\ 0]^T$, the state estimates $\hat{x}$ are initialized to be zero,

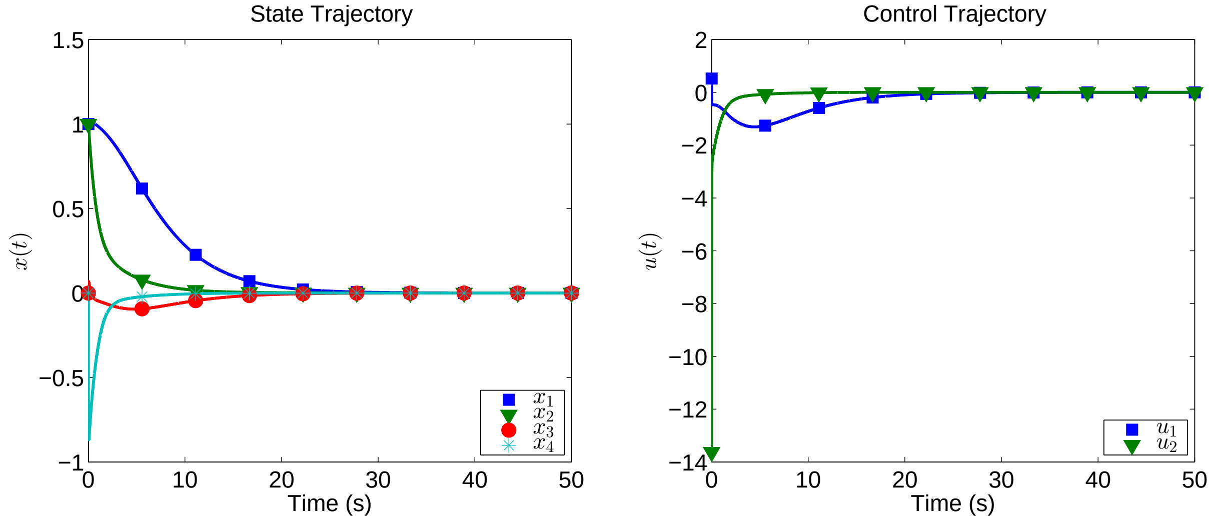

Figure 3-5. System state and control trajectories generated using the developed method for the system in Section 3.5.2.

the parameter estimates $\hat{\theta}$ are initialized to be one, and a history stack containing thirty data points is recorded online using a singular value maximizing algorithm (cf. [93]) for the CL-based adaptive update law in (3–5). The state derivative at the recorded data points is computed using a fifth order Savitzky-Golay smoothing filter (cf. [150]).

Figures 3-5 - 3-7 demonstrates that the system state is regulated to the origin, the unknown parameters in the drift dynamics are identified, and the value function and the policy weights converge. The value function and the policy weights converge to the following values.

$$\hat{W}_c^* = \hat{W}_a^* = [24.7,\ 1.19,\ 2.25,\ 2.67,\ 1.18,\ 0.93,\ 44.34,\ 11.31,\ 3.81\ ,0.10]^T. \tag{3–31}$$

Since the true values of the value function weights are unknown, the weights in (3–31) cannot be compared to their true values. However, a measure of proximity of the weights in (3–31) to the ideal weights W can be obtained by comparing the system trajectories resulting from applying the feedback control policy $\hat{u}^*(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla\sigma^T(x)\hat{W}_a^*$ to the system, against numerically computed optimal system trajectories. In Figure 3-8, the numerical optimal solution is obtained using

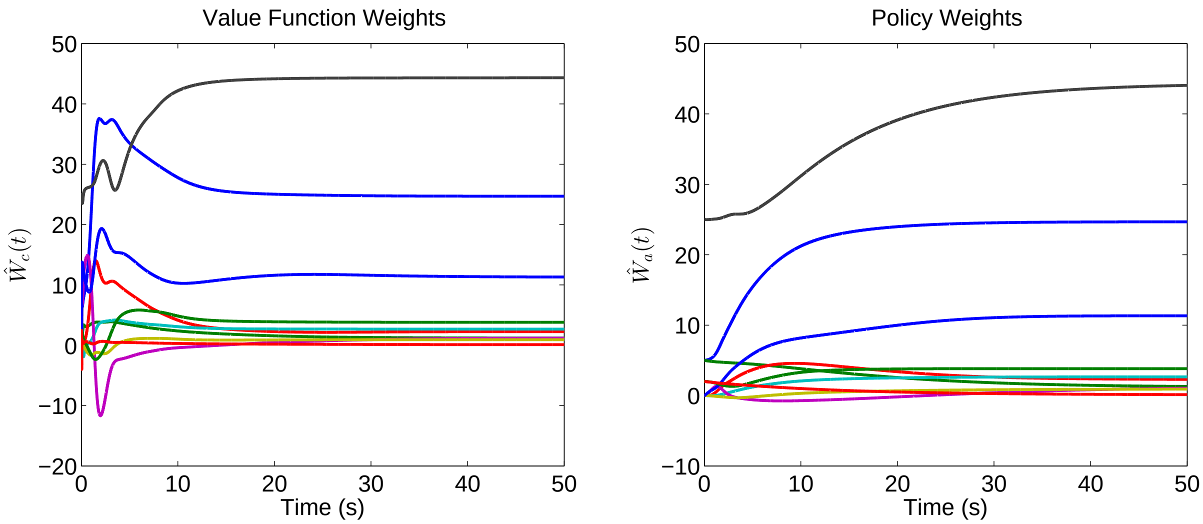

Figure 3-6. Actor and critic weight trajectories generated using the developed method for the system in Section 3.5.2. Since an analytical optimal solution is not available, the weight estimates cannot be compared with their true values.

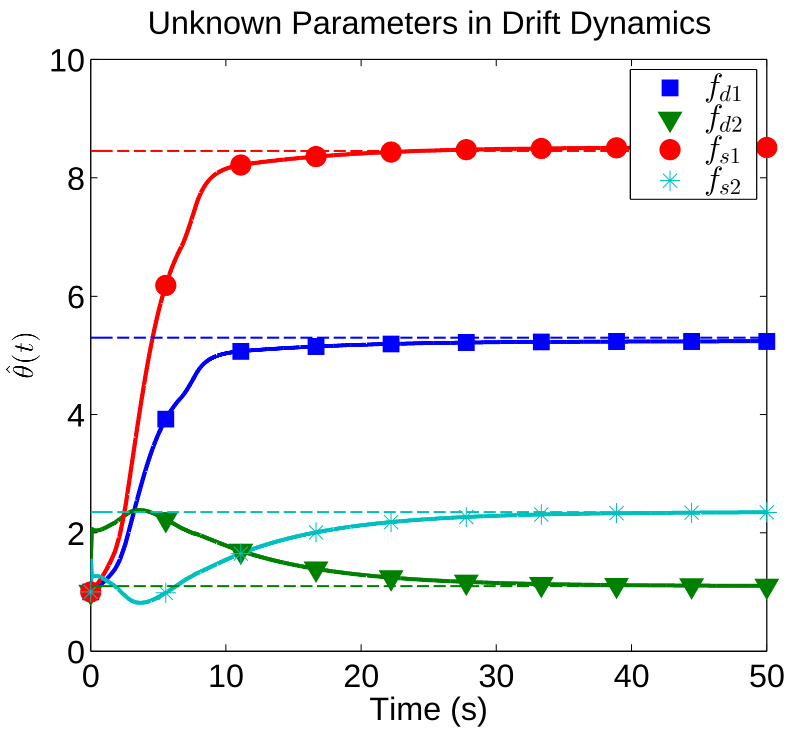

Figure 3-7. Drift parameter estimate trajectories generated using the developed method for the system in Section 3.5.2 compared to the actual drift parameters. The dotted lines represent true values of the drift parameters.

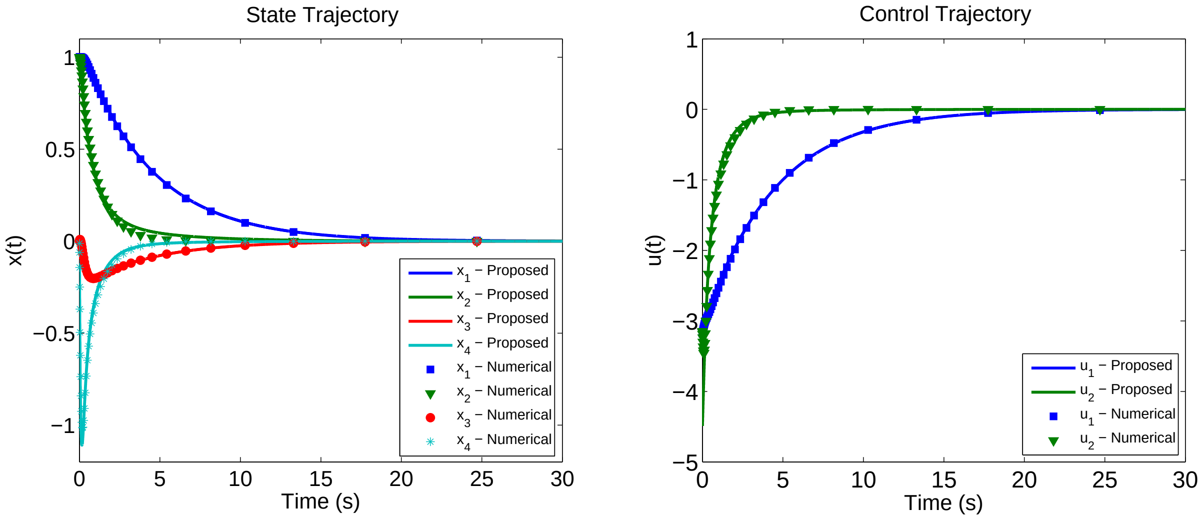

Figure 3-8. State and control trajectories generated using feedback policy $\hat{u}^*(x)$ compared to a numerical optimal solution for the system in Section 3.5.2.

an infinite-horizon Gauss pseudospectral method (cf. [9]) using 45 collocation points. Figure 3-8 indicates that the weights in (3–31) generate state and control trajectories that closely match the numerically computed optimal trajectories.

3.6 Concluding Remarks

An online approximate optimal controller is developed, where the value function is approximated without PE via novel use of a CL-based system identifier to implement simulation of experience in model-based RL. The PE condition is replaced by a weaker rank condition that can be verified online from recorded data. UB regulation of the system states to a neighborhood of the origin, and convergence of the policy to a neighborhood of the optimal policy are established using a Lyapunov-based analysis. Simulations demonstrate that the developed technique generates an approximation to the optimal controller online, while maintaining system stability, without the use of an ad-hoc probing signal. The Lyapunov analysis suggests that the convergence critically depends on the amount of collective information available in the set of BEs evaluated at the predefined points. This relationship is similar to the conditions on the strength and the interval of PE that are required for parameter convergence in adaptive systems in the presence of bounded or Lipschitz additive disturbances.

The control technique developed in this chapter does not account for additive external disturbances. Traditionally, optimal disturbance rejection is achieved via feedback-Nash equilibrium solution of an H_∞ control problem. The H_∞ control problem is a two-player zero-sum differential game problem. Motivated by the need to accomplish disturbance rejection, the following chapter extends the results of this chapter to obtain feedback-Nash equilibrium solutions to a more general $N-$player nonzero-sum differential game.

CHAPTER 4
MODEL-BASED REINFORCEMENT LEARNING FOR ONLINE APPROXIMATE FEEDBACK-NASH EQUILIBRIUM SOLUTION OF N-PLAYER NONZERO-SUM DIFFERENTIAL GAMES

In this chapter, a CL-based ACI architecture (cf. [59]) is used to obtain an approximate feedback-Nash equilibrium solution to an infinite-horizon N-player nonzero-sum differential game online, without requiring PE, for a nonlinear control-affine system with uncertain LP drift dynamics.

A system identifier is used to estimate the unknown parameters in the drift dynamics. The solutions to the coupled HJ equations and the corresponding feedback-Nash equilibrium policies are approximated using parametric universal function approximators. Based on estimates of the unknown drift parameters, estimates for the Bellman errors are evaluated at a set of pre-selected points in the state-space. The value function and the policy weights are updated using a concurrent learning-based least-squares approach to minimize the instantaneous BEs and the BEs evaluated at pre-selected points. Simultaneously, the unknown parameters in the drift dynamics are updated using a history stack of recorded data via a concurrent learning-based gradient descent approach. It is shown that under a condition milder than PE, UB convergence of the unknown drift parameters, the value function weights and the policy weights to their true values can be established. Simulation results are presented to demonstrate the performance of the developed technique without an added excitation signal.

4.1 Problem Formulation and Exact Solution

Consider a class of control-affine multi-input systems

$$\dot{x} = f\left(x\right) + \sum_{i=1}^{N} g_i\left(x\right) u_i, \tag{4–1}$$

where $x \in \mathbb{R}^n$ is the state and $u_i \in \mathbb{R}^{m_i}$ are the control inputs (i.e. the players). In (4–1), the unknown function $f : \mathbb{R}^n \to \mathbb{R}^n$ is LP[1] , the functions $g_i : \mathbb{R}^n \to \mathbb{R}^{n \times m_i}$ are known, locally Lipschitz continuous and uniformly bounded, the function f is locally Lipschitz, and $f(0) = 0$. Define a cost functional

$$J_i\left(x_i, u_i, .., u_N\right) = \int\limits_0^\infty r_i\left(x_i\left(\sigma\right), u_i\left(\sigma\right)\right) d\sigma \tag{4–2}$$

where $r_i : \mathbb{R}^n \times \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_N} \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as $r_i\left(x, u_i, .., u_N\right) \triangleq x^T Q_i x + \sum_{j=1}^N u_j^T R_{ij} u_j$, where $Q_i \in \mathbb{R}^{n \times n}$ and $R_{ij} \in \mathbb{R}^{m_j \times m_j}$ are constant positive definite matrices. The objective of each agent is to minimize the cost functional in (4–2). To facilitate the definition of a feedback-Nash equilibrium solution, let

$$U \triangleq \left\{\left\{\overline{u}_i : \mathbb{R}^n \to \mathbb{R}^{m_i}, i = 1, .., N\right\} \mid \left\{\overline{u}_1, .., \overline{u}_N\right\} \text{ is admissible with respect to (4–1)}\right\}$$

be the set of all admissible tuples of feedback policies. A tuple $\left\{\overline{u}_1, .., \overline{u}_N\right\}$ is called admissible if the functions $\overline{u}_i$ are continuous for all $i = 1, .., N$, and result in finite costs J_i for all $i = 1, .., N$. Let $V_i^{\left\{\overline{u}_1, .., \overline{u}_N\right\}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ denote the value function of the i^{th} player with respect to the tuple of feedback policies $\left\{\overline{u}_1, .., \overline{u}_N\right\} \in U$, defined as

$$V_i^{\left\{\overline{u}_1, .., \overline{u}_N\right\}}\left(x\right) \triangleq \int\limits_t^\infty r_i\left(\phi\left(\tau, x\right), \overline{u}_1\left(\phi\left(\tau, x\right)\right), .., \overline{u}_N\left(\phi\left(\tau, x\right)\right)\right) d\tau, \tag{4–3}$$

where $\phi\left(\tau, x\right)$ for $\tau \in [t, \infty)$ denotes the trajectory of (4–1) obtained using the feedback controller $u_i\left(\tau\right) = \overline{u}_i\left(\phi\left(\tau, x\right)\right)$ and the initial condition $\phi\left(t, x\right) = x$. In (4–3), $r_i : \mathbb{R}^n \times \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_N} \to \mathbb{R}_{\geq 0}$ denotes the instantaneous cost defined as $r_i\left(x, u_i, .., u_N\right) \triangleq x^T Q_i x + \sum_{j=1}^N u_j^T R_{ij} u_j$, where $Q_i \in \mathbb{R}^{n \times n}$ is a positive definite matrix. The control

[1] The function f is assumed to be LP for ease of exposition. The system identifier can also be developed using multi-layer NNs. For example, a system identifier developed using single-layer NNs is presented in Chapter 6.

objective is to find an approximate feedback-Nash equilibrium solution to the infinite-horizon regulation differential game online, i.e., to find a tuple $\{u_1^*, .., u_N^*\} \in U$ such that for all $i \in \{1, .., N\}$, for all $x \in \mathbb{R}^n$, the corresponding value functions satisfy

$$V_i^*(x) \triangleq V_i^{\left\{u_1^*, u_2^*, .., u_i^*, .., u_N^*\right\}}(x) \leq V_i^{\left\{u_1^*, u_2^*, .., \overline{u}_i, .., u_N^*\right\}}(x)$$

for all $\overline{u}_i$ such that $\{u_1^*, u_2^*, .., \overline{u}_i, .., u_N^*\} \in U$.

Provided a feedback-Nash equilibrium solution exists and provided the value functions are continuously differentiable, an exact closed-loop feedback-Nash equilibrium solution $\{u_i^*, .., u_N^*\}$ can be expressed in terms of the value functions as [100, 103, 104, 107, 112]

$$u_i^*(x^o) = -\frac{1}{2} R_{ii}^{-1} g_i^T(x^o) \left(\nabla V_i^*(x^o)\right)^T, \; \forall x^o \in \mathbb{R}^n, \tag{4–4}$$

and the value functions $\{V_1^*, .., V_N^*\}$ are the solutions to the coupled HJ equations

$$x^{oT} Q_i x^o + \sum_{j=1}^{N} \frac{1}{4} \nabla V_j^*(x^o) G_{ij}(x^o) \left(\nabla V_j^*(x^o)\right)^T - \frac{1}{2} \nabla V_i^*(x^o) \sum_{j=1}^{N} G_j(x^o) \left(\nabla V_j^*(x^o)\right)^T + \nabla V_i^*(x^o) f(x^o) = 0, \tag{4–5}$$

for all $x^o \in \mathbb{R}^n$. In (4–5), $G_j(x^o) \triangleq g_j(x^o) R_{jj}^{-1} g_j^T(x^o)$ and $G_{ij}(x^o) \triangleq g_j(x^o) R_{jj}^{-1} R_{ij} R_{jj}^{-1} g_j^T(x^o)$. The HJ equations in (4–5) are in the so-called closed-loop form; they can be expressed in an open-loop form as

$$x^{oT} Q_i x^o + \sum_{j=1}^{N} u_j^{*T}(x^o) R_{ij} u_j^*(x^o) + \nabla V_i^*(x^o) f(x^o) + \nabla V_i^*(x^o) \sum_{j=1}^{N} g_j(x^o) u_j^*(x^o) = 0,$$

for all $x^o \in \mathbb{R}^n$.

4.2 Approximate Solution

Computation of an analytical solution to the coupled nonlinear HJ equations in (4–5) is, in general, infeasible. Hence, similar to Chapter 3, a parametric approximate solution $\left\{\hat{V}_1\left(x, \hat{W}_{c1}\right), .., \hat{V}_N\left(x, \hat{W}_{cN}\right)\right\}$ is sought. Based on $\left\{\hat{V}_1\left(x, \hat{W}_{c1}\right), .., \hat{V}_N\left(x, \hat{W}_{cN}\right)\right\}$, an approximation $\left\{\hat{u}_1\left(x,\hat{W}_{a1}\right),..,\hat{u}_N\left(x,\hat{W}_{aN}\right)\right\}$ to the closed-loop feedback-Nash equilibrium solution is computed, where $\hat{W}_{ci}\in\mathbb{R}^{pW_i}$, i.e., the value function weights, and $\hat{W}_{ai}\in\mathbb{R}^{pW_i}$, i.e., the policy weights, denote the parameter estimates. Since the approximate solution, in general, does not satisfy the HJ equations, a set of residual errors $\delta_i:\mathbb{R}^n\times\mathbb{R}^{pW_i}\times\mathbb{R}^{pW_1}\times,\cdots\times\mathbb{R}^{pW_N}\to\mathbb{R}$, called BEs, is defined as

$$\delta_i\left(x,\hat{W}_{ci},\hat{W}_{a1},\cdots,\hat{W}_{aN}\right)\triangleq x^TQ_ix+\sum_{j=1}^{N}\hat{u}_j^T\left(x,\hat{W}_{aj}\right)R_{ij}\hat{u}_j\left(x,\hat{W}_{aj}\right)$$
$$+\nabla\hat{V}_i\left(x,\hat{W}_{ci}\right)f\left(x\right)+\nabla\hat{V}_i\left(x,\hat{W}_{ci}\right)\sum_{j=1}^{N}g_j\left(x\right)\hat{u}_j\left(x,\hat{W}_{aj}\right),\quad\text{(4–6)}$$

and the approximate solution is recursively improved to drive the BEs to zero. The computation of the BEs in (4–6) requires knowledge of the drift dynamics f. To eliminate this requirement, and to enable simulation of experience via BE extrapolation, a concurrent learning-based system identifier is developed in the following section.

4.2.1 System Identification

Let $f\left(x^o\right)=Y\left(x^o\right)\theta$, for all $x^o\in\mathbb{R}^n$, be the linear parameterization of the drift dynamics, where $Y:\mathbb{R}^n\to\mathbb{R}^{n\times p_\theta}$ denotes the locally Lipschitz regression matrix, and $\theta\in\mathbb{R}^{p_\theta}$ denotes the vector of constant, unknown drift parameters. The system identifier is designed as

$$\dot{\hat{x}}=Y\left(x\right)\hat{\theta}+\sum_{i=1}^{N}g_i\left(x\right)u_i+k_x\tilde{x},\quad\text{(4–7)}$$

where the measurable state estimation error $\tilde{x}$ is defined as $\tilde{x}\triangleq x-\hat{x}$, $k_x\in\mathbb{R}^{n\times n}$ is a positive definite, constant diagonal observer gain matrix, and $\hat{\theta}\in\mathbb{R}^{p_\theta}$ denotes the vector of estimates of the unknown drift parameters. In traditional adaptive systems, the estimates are updated to minimize the instantaneous state estimation error, and convergence of parameter estimates to their true values can be established under a restrictive PE condition. In this result, a concurrent learning-based data-driven approach is developed to relax the PE condition to a weaker, verifiable rank condition as follows.

Assumption 4.1. [92, 93] A history stack $\mathcal{H}_{id}$ containing state-action tuples $\left\{\left(x_j, u_{i_j}\right) \mid i = 1, \cdots, N,\ j = 1, \cdots, M_\theta\right\}$ recorded along the trajectories of (4–1) that satisfies

$$\text{rank}\left(\sum_{j=1}^{M_\theta} Y_j^T Y_j\right) = p_\theta,$$

is available a priori, where $Y_j \triangleq Y\left(x_j\right)$, and p_θ denotes the number of unknown parameters in the drift dynamics.

To facilitate the concurrent learning-based parameter update, numerical methods are used to compute the state derivative $\dot{x}_j$ corresponding to $\left(x_j, \hat{u}_{i_j}\right)$. The update law for the drift parameter estimates is designed as

$$\dot{\hat{\theta}} = \Gamma_\theta Y^T \tilde{x} + \Gamma_\theta k_\theta \sum_{j=1}^{M_\theta} Y_j^T \left(\dot{x}_j - \sum_{i=1}^{N} g_{i_j} u_{i_j} - Y_j \hat{\theta}\right), \tag{4–8}$$

where $g_{i_j} \triangleq g_i\left(x_j\right)$, $\Gamma_\theta \in \mathbb{R}^{p_\theta \times p_\theta}$ is a constant positive definite adaptation gain matrix, and $k_\theta \in \mathbb{R}$ is a constant positive concurrent learning gain. The update law in (4–8) requires the unmeasurable state derivative $\dot{x}_j$. Since the state derivative at a past recorded point on the state trajectory is required, past and future recorded values of the state can be used along with accurate noncausal smoothing techniques to obtain good estimates of $\dot{x}_j$. In the presence of derivative estimation errors, the parameter estimation errors can be shown to be UUB, where the size of the ultimate bound depends on the error in the derivative estimate [93].

To incorporate new information, the history stack is updated with new data. Thus, the resulting closed-loop system is a switched system. To ensure the stability of the switched system, the history stack is updated using a singular value maximizing algorithm (cf. [93]). Using (4–1), the state derivative can be expressed as

$$\dot{x}_j - \sum_{i=1}^{N} g_{i_j} u_{i_j} = Y_j \theta,$$

and hence, the update law in (4–8) can be expressed in the advantageous form

$$\dot{\hat{\theta}} = -\Gamma_\theta Y^T \tilde{x} - \Gamma_\theta k_\theta \left(\sum_{j=1}^{M_\theta} Y_j^T Y_j \right) \tilde{\theta}, \tag{4–9}$$

where $\tilde{\theta} \triangleq \theta - \hat{\theta}$ denotes the drift parameter estimation error. The closed-loop dynamics of the state estimation error are given by

$$\dot{\tilde{x}} = Y\tilde{\theta} - k_x \tilde{x}. \tag{4–10}$$

4.2.2 Value Function Approximation

The value functions, i.e., the solutions to the HJ equations in (4–5), are continuously differentiable functions of the state. Using the universal approximation property of NNs, the value functions can be represented as

$$V_i^* \left(x^o \right) = W_i^T \sigma_i \left(x^o \right) + \epsilon_i \left(x^o \right), \tag{4–11}$$

for all $x^o \in \mathbb{R}^n$, where $W_i \in \mathbb{R}^{p_{W_i}}$ denotes the constant vector of unknown NN weights, $\sigma_i : \mathbb{R}^n \to \mathbb{R}^{p_{W_i}}$ denotes the known NN activation function, $p_{W_i} \in \mathbb{N}$ denotes the number of hidden layer neurons, and $\epsilon_i : \mathbb{R}^n \to \mathbb{R}$ denotes the unknown function reconstruction error. The universal function approximation property guarantees that over any compact domain $\mathcal{C} \subset \mathbb{R}^n$, for all constant $\overline{\epsilon}_i, \overline{\nabla \epsilon_i} > 0$, there exists a set of weights and basis functions such that $\|W_i\| \leq \overline{W}$, $\sup_{x \in \mathcal{C}} \|\sigma_i(x)\| \leq \overline{\sigma}_i$, $\sup_{x \in \mathcal{C}} \|\nabla \sigma_i(x)\| \leq \overline{\nabla \sigma_i}$, $\sup_{x \in \mathcal{C}} \|\epsilon_i(x)\| \leq \overline{\epsilon}_i$ and $\sup_{x \in \mathcal{C}} \|\nabla \epsilon_i(x)\| \leq \overline{\nabla \epsilon_i}$, where $\overline{W}_i, \overline{\sigma}_i, \overline{\nabla \sigma_i}, \overline{\epsilon}_i, \overline{\nabla \epsilon_i} \in \mathbb{R}$ are positive constants. Based on (4–4) and (4–11), the feedback-Nash equilibrium solutions are given by

$$u_i^* \left(x^o \right) = -\frac{1}{2} R_{ii}^{-1} g_i^T \left(x^o \right) \left(\nabla \sigma_i^T \left(x^o \right) W_i + \nabla \epsilon_i^T \left(x^o \right) \right), \; \forall x^o \in \mathbb{R}^n. \tag{4–12}$$

The NN-based approximations to the value functions and the controllers are defined as

$$\hat{V}_i \left(x, \hat{W}_{ci} \right) \triangleq \hat{W}_{ci}^T \sigma_i \left(x \right), \quad \hat{u}_i \left(x, \hat{W}_{ai} \right) \triangleq -\frac{1}{2} R_{ii}^{-1} g_i^T \left(x \right) \nabla \sigma_i^T \left(x \right) \hat{W}_{ai}, \tag{4–13}$$

The use of two different sets $\left\{\hat{W}_{ci}\right\}$ and $\left\{\hat{W}_{ai}\right\}$ of estimates to approximate the same set of ideal weights $\{W_i\}$ is motivated by the subsequent stability analysis and the fact that it facilitates an approximate formulation of the BEs that is affine in the value function weights, enabling least squares-based adaptation. Based on (4–13), measurable approximations $\hat{\delta}_i : \mathbb{R}^n \times \mathbb{R}^{p_{W_i}} \times \mathbb{R}^{p_{W_1}} \times, \cdots \times \mathbb{R}^{p_{W_N}} \times \mathbb{R}^{p_\theta} \to \mathbb{R}$ to the BEs in (4–6) are defined as

$$\hat{\delta}_i\left(x, \hat{W}_{ci}, \hat{W}_{a1}, \cdots, \hat{W}_{aN}, \hat{\theta}\right) \triangleq \hat{W}_{ci}^T \left(\nabla\sigma_i(x) Y(x) \hat{\theta} - \frac{1}{2}\sum_{j=1}^{N} \nabla\sigma_i(x) G_j(x) \nabla\sigma_j^T(x) \hat{W}_{aj}\right)$$
$$+ x^T Q_i x + \sum_{j=1}^{N} \frac{1}{4} \hat{W}_{aj}^T \nabla\sigma_j(x) G_{ij}(x) \nabla\sigma_j^T(x) \hat{W}_{aj}, \quad \text{(4–14)}$$

The following assumption, which in general is weaker than the PE assumption, is required for convergence of the concurrent learning-based value function weight estimates.

Assumption 4.2. For each $i \in \{1, .., N\}$, there exists a finite set of M_{xi} points $\{x_{ij} \in \mathbb{R}^n \mid j = 1, .., M_{xi}\}$ such that

$$\underline{c}_{xi} \triangleq \frac{\left(\inf_{t\in\mathbb{R}_{\geq 0}}\left(\lambda_{\min}\left\{\sum_{k=1}^{M_{xi}} \frac{\omega_i^k(t)\left(\omega_i^k\right)^T(t)}{\rho_i^k(t)}\right\}\right)\right)}{M_{xi}} > 0, \quad \text{(4–15)}$$

where $\lambda_{\min}$ denotes the minimum eigenvalue, and $\underline{c}_{xi} \in \mathbb{R}$ is a positive constant. In (4–15),

$$\omega_i^k = \nabla\sigma_i^{ik} Y^{ik} \hat{\theta} - \frac{1}{2}\sum_{j=1}^{N} \nabla\sigma_i^{ik} G_j^{ik} \left(\nabla\sigma_j^{ik}\right)^T \hat{W}_{aj},$$

where the superscript ik indicates that the function is evaluated at $x = x_{ik}$, and $\rho_i^k \triangleq 1 + \nu_i \left(\omega_i^k\right)^T \Gamma_i \omega_i^k$, where $\nu_i \in \mathbb{R}_{>0}$ is the normalization gain and $\Gamma_i \in \mathbb{R}^{P_{W_i} \times P_{W_i}}$ is the adaptation gain matrix.

The concurrent learning-based least-squares update law for the value function weights is designed as

$$\dot{\hat{W}}_{ci} = -\eta_{c1i}\Gamma_i \frac{\omega_i}{\rho_i}\hat{\delta}_{ti} - \frac{\eta_{c2i}\Gamma_i}{M_{xi}}\sum_{k=1}^{M_{xi}}\frac{\omega_i^k}{\rho_i^k}\hat{\delta}_{ti}^k,$$

$$\dot{\Gamma}_i = \left(\beta_i\Gamma_i - \eta_{c1i}\Gamma_i\frac{\omega_i\omega_i^T}{\rho_i^2}\Gamma_i\right)\mathbf{1}_{\left\{\|\Gamma_i\|\le\overline{\Gamma}_i\right\}},\ \|\Gamma_i\left(t_0\right)\|\le\overline{\Gamma}_i, \tag{4–16}$$

where $\omega_i = \nabla\sigma_i\left(x\right)Y\left(x\right)\hat{\theta} - \frac{1}{2}\sum_{j=1}^{N}\nabla\sigma_i\left(x\right)G_j\left(x\right)\nabla\sigma_j^T\left(x\right)\hat{W}_{aj}\left(t\right),\ \rho_i \triangleq 1+\nu_i\omega_i^T\Gamma_i\omega_i$, $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function, $\overline{\Gamma}_i > 0 \in \mathbb{R}$ is the saturation constant, $\beta_i \in \mathbb{R}$ is the constant positive forgetting factor, $\eta_{c1i}, \eta_{c2i} \in \mathbb{R}$ are constant positive adaptation gains, and the instantaneous BEs $\hat{\delta}_{ti}$ and $\hat{\delta}_{ti}^k$ are defined as

$$\hat{\delta}_{ti}\left(t\right) \triangleq \hat{\delta}_i\left(x\left(t\right), \hat{W}_{ci}\left(t\right), \hat{W}_{a1}\left(t\right), \cdots, \hat{W}_{aN}\left(t\right), \hat{\theta}\left(t\right)\right),$$

$$\hat{\delta}_{ti}^k\left(t\right) \triangleq \hat{\delta}_i\left(x_{ik}, \hat{W}_{ci}\left(t\right), \hat{W}_{a1}\left(t\right), \cdots, \hat{W}_{aN}\left(t\right), \hat{\theta}\left(t\right)\right).$$

The policy weight update laws are designed based on the subsequent stability analysis as

$$\dot{\hat{W}}_{ai} = -\eta_{a1i}\left(\hat{W}_{ai} - \hat{W}_{ci}\right) - \eta_{a2i}\hat{W}_{ai} + \frac{1}{4}\sum_{k=1}^{M_{xi}}\sum_{j=1}^{N}\frac{\eta_{c2i}}{M_{xi}}\nabla\sigma_j^{ik}G_{ij}^{ik}\left(\nabla\sigma_j^{ik}\right)^T\hat{W}_{aj}^T\frac{\left(\omega_i^k\right)^T}{\rho_i^k}\hat{W}_{ci}^T$$

$$+\frac{1}{4}\sum_{j=1}^{N}\eta_{c1i}\nabla\sigma_j\left(x\right)G_{ij}\left(x\right)\nabla\sigma_j^T\left(x\right)\hat{W}_{aj}^T\frac{\omega_i^T}{\rho_i}\hat{W}_{ci}^T, \tag{4–17}$$

where $\eta_{a1i}, \eta_{a2i} \in \mathbb{R}$ are positive constant adaptation gains. The forgetting factor β_i along with the saturation in the update law for the least-squares gain matrix in (4–16) ensure (cf. [91]) that the least-squares gain matrix Γ_i and its inverse are positive definite and bounded for all $i \in \{1, .., N\}$ as

$$\underline{\Gamma}_i \le \|\Gamma_i\left(t\right)\| \le \overline{\Gamma}_i, \forall t \in \mathbb{R}_{\ge 0}, \tag{4–18}$$

where $\underline{\Gamma}_i \in \mathbb{R}$ is a positive constant, and the normalized regressor is bounded as

$$\left\| \frac{\omega_i}{\rho_i} \right\| \leq \frac{1}{2\sqrt{\nu_i \underline{\Gamma}_i}}.$$

For notational brevity, state-dependence of the functions f, g_i, u_i^*, G_i, G_{ij}, σ_i, Y, ϵ and V_i^* and is suppressed hereafter.

4.3 Stability Analysis

Subtracting (4–5) from (4–14), the approximate BE can be expressed in an unmeasurable form as

$$\hat{\delta}_{ti} = \omega_i^T \hat{W}_{ci} + \sum_{j=1}^{N} \frac{1}{4} \hat{W}_{aj}^T \nabla\sigma_j G_{ij} \nabla\sigma_j^T \hat{W}_{aj} - \sum_{j=1}^{N} u_j^{*T} R_{ij} u_j^* - \nabla V_i^* f - \nabla V_i^* \sum_{j=1}^{N} g_j u_j^*$$

Substituting for V^* and u^* from (4–11) and (4–12) and using $f = Y\theta$, the approximate BE can be expressed as

$$\begin{aligned}\hat{\delta}_{ti} = {} & \omega_i^T \hat{W}_{ci} + \sum_{j=1}^{N} \frac{1}{4} \hat{W}_{aj}^T \nabla\sigma_j G_{ij} \nabla\sigma_j^T \hat{W}_{aj} - W_i^T \nabla\sigma_i Y\theta - \nabla\epsilon_i Y\theta - \sum_{j=1}^{N} \frac{1}{4} W_j^T \nabla\sigma_j G_{ij} \nabla\sigma_j^T W_j \\ & - \sum_{j=1}^{N} \frac{1}{2} \epsilon_j' G_{ij} \nabla\sigma_j^T W_j - \sum_{j=1}^{N} \frac{1}{4} \epsilon_j' G_{ij} \nabla\epsilon_j^T + \frac{1}{2} \sum_{j=1}^{N} \nabla\epsilon_i G_j \nabla\epsilon_j^T + \frac{1}{2} \sum_{j=1}^{N} W_i^T \nabla\sigma_i G_j \nabla\sigma_j^T W_j \\ & + \frac{1}{2} \sum_{j=1}^{N} \nabla\epsilon_i G_j \nabla\sigma_j^T W_j + \frac{1}{2} \sum_{j=1}^{N} W_i^T \nabla\sigma_i G_j \nabla\epsilon_j^T,\end{aligned}$$

Adding and subtracting $\frac{1}{4} \hat{W}_{aj}^T \nabla\sigma_j G_{ij} \nabla\sigma_j^T W_j + \omega_i^T W_i$ yields

$$\begin{aligned}\hat{\delta}_{ti} = {} & -\omega_i^T \tilde{W}_{ci} + \frac{1}{4} \sum_{j=1}^{N} \tilde{W}_{aj}^T \nabla\sigma_j G_{ij} \nabla\sigma_j^T \tilde{W}_{aj} - \frac{1}{2} \sum_{j=1}^{N} \left(W_i^T \nabla\sigma_i G_j - W_j^T \nabla\sigma_j G_{ij} \right) \nabla\sigma_j^T \tilde{W}_{aj} \\ & - W_i^T \nabla\sigma_i Y \tilde{\theta} - \nabla\epsilon_i Y\theta + \Delta_i, \quad \text{(4–19)}\end{aligned}$$

where $\Delta_i \triangleq \frac{1}{2} \sum_{j=1}^{N} \left(W_i^T \nabla\sigma_i G_j - W_j^T \nabla\sigma_j G_{ij} \right) \nabla\epsilon_j^T + \frac{1}{2} \sum_{j=1}^{N} W_j^T \nabla\sigma_j G_j \nabla\epsilon_i^T + \frac{1}{2} \sum_{j=1}^{N} \nabla\epsilon_i G_j \nabla\epsilon_j^T - \sum_{j=1}^{N} \frac{1}{4} \epsilon_j' G_{ij} \nabla\epsilon_j^T$. Similarly, the approximate BE evaluated at the selected points can be expressed in an unmeasurable form as

$$\hat{\delta}_{ti}^{k} = -\omega_i^{kT}\tilde{W}_{ci} + \Delta_i^k - W_i^T\nabla\sigma_i^{ik}Y^{ik}\tilde{\theta} - \frac{1}{2}\sum_{j=1}^{N}\left(W_i^T\nabla\sigma_i^{ik}G_j^{ik} - W_j^T\nabla\sigma_j^{ik}G_{ij}^{ik}\right)\left(\nabla\sigma_j^{ik}\right)^T\tilde{W}_{aj}$$

$$+ \frac{1}{4}\sum_{j=1}^{N}\tilde{W}_{aj}^T\nabla\sigma_j^{ik}G_{ij}^{ik}\left(\nabla\sigma_j^{ik}\right)^T\tilde{W}_{aj}, \quad \text{(4–20)}$$

where the constant $\Delta_i^k \in \mathbb{R}$ is defined as $\Delta_i^k \triangleq -\epsilon_i'^{ik}Y^{ik}\theta + \Delta_i^{ik}$. To facilitate the stability analysis, a candidate Lyapunov function is defined as

$$V_L = \sum_{i=1}^{N}V_i^* + \frac{1}{2}\sum_{i=1}^{N}\tilde{W}_{ci}^T\Gamma_i^{-1}\tilde{W}_{ci} + \frac{1}{2}\sum_{i=1}^{N}\tilde{W}_{ai}^T\tilde{W}_{ai} + \frac{1}{2}\tilde{x}^T\tilde{x} + \frac{1}{2}\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}. \quad \text{(4–21)}$$

Since V_i^* are positive definite, the bound in (4–18) and Lemma 4.3 in [149] can be used to bound the candidate Lyapunov function as

$$\underline{v}\left(\|Z^o\|\right) \leq V_L\left(Z^o, t\right) \leq \overline{v}\left(\|Z^o\|\right) \quad \text{(4–22)}$$

for all $Z^o = \left[x^T, \tilde{W}_{c1}^T, .., \tilde{W}_{cN}^T, \tilde{W}_{a1}^T, .., \tilde{W}_{aN}^T, \tilde{x}, \tilde{\theta}\right]^T \in \mathbb{R}^{2n+2N\sum_i p_{W_i}+p_\theta}$ and $\underline{v}, \overline{v} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. For any compact set $\mathcal{Z} \subset \mathbb{R}^{2n+2N\sum_i p_{W_i}+p_\theta}$, define

$$\iota_1 \triangleq \max_{i,j}\left(\sup_{Z\in\mathcal{Z}}\left\|\frac{1}{2}W_i^T\nabla\sigma_iG_j\nabla\sigma_j^T + \frac{1}{2}\nabla\epsilon_iG_j\nabla\sigma_j^T\right\|\right), \ \iota_4 \triangleq \max_{i,j}\left(\sup_{Z\in\mathcal{Z}}\left\|\nabla\sigma_jG_{ij}\nabla\sigma_j^T\right\|\right),$$

$$\iota_{5i} \triangleq \frac{\eta_{c1i}L_Y\overline{\nabla\epsilon_i\theta}}{4\sqrt{\nu_i\underline{\Gamma}_i}}, \ \iota_2 \triangleq \max_{i,j}\Bigg(\sup_{Z\in\mathcal{Z}}\Bigg\|\frac{\eta_{c1i}\omega_i}{4\rho_i}\left(3W_j\nabla\sigma_jG_{ij} - 2W_i^T\nabla\sigma_iG_j\right)\nabla\sigma_j^T$$

$$+\sum_{k=1}^{M_{xi}}\frac{\eta_{c2i}\omega_i^k}{4M_{xi}\rho_i^k}\left(3W_j^T\nabla\sigma_j^{ik}G_{ij}^{ik} - 2W_i^T\nabla\sigma_i^{ik}G_j^{ik}\right)\left(\nabla\sigma_j^{ik}\right)^T\Bigg\|\Bigg)$$

$$\iota_3 \triangleq \max_{i,j}\left(\sup_{Z\in\mathcal{Z}}\left\|\frac{1}{2}\sum_{i,j=1}^{N}\left(W_i^T\nabla\sigma_i + \nabla\epsilon_i\right)G_j\nabla\epsilon_j^T - \frac{1}{4}\sum_{i,j=1}^{N}\left(2W_j^T\nabla\sigma_j + \epsilon_j'\right)G_{ij}\nabla\epsilon_j^T\right\|\right)$$

$$\iota_{6i} \triangleq \frac{\eta_{c1i}L_Y\overline{W}_i\overline{\nabla\sigma_i}}{4\sqrt{\nu_i\underline{\Gamma}_i}}, \ \iota_{7i} \triangleq \frac{\eta_{c2i}\max_k\left\|\nabla\sigma_i^{ik}Y^{ik}\right\|\overline{W}_i}{4\sqrt{\nu_i\underline{\Gamma}_i}}, \ \iota_8 \triangleq \sum_{i=1}^{N}\frac{\left(\eta_{c1i}+\eta_{c2i}\right)\overline{W}_i\iota_4}{8\sqrt{\nu_i\underline{\Gamma}_i}},$$

$$\iota_{9i} \triangleq \left(\iota_1N + \left(\eta_{a2i} + \iota_8\right)\overline{W}_i\right), \ \iota_{10i} \triangleq \frac{\eta_{c1i}\sup_{Z\in\mathcal{Z}}\left\|\Delta_i\right\| + \eta_{c2i}\max_k\left\|\Delta_i^k\right\|}{2\sqrt{\nu_i\underline{\Gamma}_i}}$$

$$v_l \triangleq \min\left(\frac{\underline{q_i}}{2}, \frac{\eta_{c2i}\underline{c}_{xi}}{4}, \underline{k_x}, \frac{2\eta_{a1i}+\eta_{a2i}}{8}, \frac{k_\theta\underline{y}}{2}\right), \ \iota \triangleq \sum_{i=1}^{N}\left(\frac{2\iota_{9i}^2}{2\eta_{a1i}+\eta_{a2i}} + \frac{\iota_{10i}^2}{\eta_{c2i}\underline{c}_{xi}}\right) + \iota_3, \quad \text{(4–23)}$$

where $\underline{q_i}$ denotes the minimum eigenvalue of Q_i, $\underline{y}$ denotes the minimum eigenvalue of $\sum_{j=1}^{M_\theta} Y_j^T Y_j$, $\underline{k_x}$ denotes the minimum eigenvalue of k_x, and the suprema exist since $\frac{\omega_i}{\rho_i}$ is uniformly bounded for all Z, and the functions G_i, G_{ij}, σ_i, and $\nabla\epsilon_i$ are continuous. In (4–23), $L_Y \in \mathbb{R}_{\geq 0}$ denotes the Lipschitz constant such that $\|Y(\varpi)\| \leq L_Y \|\varpi\|$ for all $\varpi \in \mathcal{Z} \cap \mathbb{R}^n$. The sufficient conditions for UB convergence are derived based on the subsequent stability analysis as

$$\begin{aligned}
\underline{q_i} &> 2\iota_{5i}, \\
\eta_{c2i}\underline{c}_{xi} &> 2\iota_{5i} + 2\zeta_1\iota_{7i} + \iota_2\zeta_2 N + \eta_{a1i} + 2\zeta_3\iota_{6i}\overline{Z}, \\
2\eta_{a1i} + \eta_{a2i} &> 4\iota_8 + \frac{2\iota_2 N}{\zeta_2}, \\
k_\theta \underline{y} &> \frac{2\iota_{7i}}{\zeta_1} + 2\frac{\iota_{6i}}{\zeta_3}\overline{Z},
\end{aligned} \tag{4–24}$$

where $\overline{Z} \triangleq \underline{v}^{-1}\left(\overline{v}\left(\max\left(\|Z(t_0)\|, \sqrt{\frac{\iota}{v_l}}\right)\right)\right)$ and $\zeta_1, \zeta_2, \zeta_3 \in \mathbb{R}$ are known positive adjustable constants. Furthermore, the compact set $\mathcal{Z}$ satisfies the sufficient condition

$$\sqrt{\frac{\iota}{v_l}} \leq r, \tag{4–25}$$

where $r \in \mathbb{R}_{\geq 0}$ denotes the radius of the set $\mathcal{Z}$.

Since the NN function approximation error and the Lipschitz constant L_Y depend on the compact set that contains the state trajectories, the compact set needs to be established before the gains can be selected using (4–24). Based on the subsequent stability analysis, an algorithm is developed to compute the required compact set (denoted by $\mathcal{Z}$) based on the initial conditions. In Algorithm 4.1, the notation $\{\varpi\}_i$ for any parameter ϖ denotes the value of ϖ computed in the i^{th} iteration. Since the constants ι and v_l depend on L_Y only through the products $L_Y\overline{\nabla\epsilon_i}$ and $L_Y\zeta_3$, Algorithm 4.1 ensures the satisfaction of the sufficient condition in that

Theorem 4.1. *Provided Assumptions 4.1-4.2 hold and the control gains satisfy the sufficient conditions in* (4–24)*, where the constants in* (4–23) *are computed based on*

Algorithm 4.1 Gain Selection

First iteration:

Given $z \in \mathbb{R}_{\geq 0}$ such that $\left\| Z\left(t_0\right) \right\| < z$, let $\mathcal{Z}_1 \triangleq \left\{ \xi \in \mathbb{R}^{2n+2N\sum_i \left\{ p_{W_i} \right\}_1 + p_\theta} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(z\right)\right) \right\}$. Using $\mathcal{Z}_1$, compute the bounds in (4–23) and select the gains according to (4–24). If $\left\{ \sqrt{\frac{\iota}{v_l}} \right\}_1 \leq z$, set $\mathcal{Z} = \mathcal{Z}_1$ and terminate.

Second iteration:

If $z < \left\{ \sqrt{\frac{\iota}{v_l}} \right\}_1$, let $\mathcal{Z}_2 \triangleq \left\{ \xi \in \mathbb{R}^{2n+2N\sum_i \left\{ p_{W_i} \right\}_1 + p_\theta} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(\left\{ \sqrt{\frac{\iota}{v_l}} \right\}_1 \right) \right) \right\}$. Using $\mathcal{Z}_2$, compute the bounds in (4–23) and select the gains according to (4–24). If $\left\{ \sqrt{\frac{\iota}{v_l}} \right\}_2 \leq \left\{ \sqrt{\frac{\iota}{v_l}} \right\}_1$, set $\mathcal{Z} = \mathcal{Z}_2$ and terminate.

Third iteration:

If $\left\{ \sqrt{\frac{\iota}{v_l}} \right\}_2 > \left\{ \sqrt{\frac{\iota}{v_l}} \right\}_1$, increase the number of NN neurons to $\left\{ p_{Wi} \right\}_3$ to ensure $\left\{ L_Y \right\}_2 \left\{ \overline{\nabla \epsilon_i} \right\}_3 \leq \left\{ L_Y \right\}_2 \left\{ \overline{\nabla \epsilon_i} \right\}_2, \forall i = 1, .., N$, decrease the constant ζ_3 to ensure $\left\{ L_Y \right\}_2 \left\{ \zeta_3 \right\}_3 \leq \left\{ L_Y \right\}_2 \left\{ \zeta_3 \right\}_2$, and increase the gain k_θ to satisfy the gain conditions in (4–24). These adjustments ensure $\left\{ \iota \right\}_3 \leq \left\{ \iota \right\}_2$. Set $\mathcal{Z} = \left\{ \xi \in \mathbb{R}^{2n+2N\sum_i \left\{ p_{W_i} \right\}_3 + p_\theta} \mid \|\xi\| \leq \underline{v}^{-1}\left(\overline{v}\left(\left\{ \sqrt{\frac{\iota}{v_l}} \right\}_2 \right) \right) \right\}$ and terminate.

the compact set $\mathcal{Z}$ selected using Algorithm 4.1, the system identifier in (4–7) along with the adaptive update law in (4–8) and the controllers $u_i\left(t\right) = \hat{u}_i\left(x\left(t\right), \hat{W}_{ai}\left(t\right)\right)$ along with the adaptive update laws in (4–16) and (4–17) ensure that the state x, the state estimation error $\tilde{x}$, the value function weight estimation errors $\tilde{W}_{ci}$ and the policy weight estimation errors $\tilde{W}_{ai}$ are UB, resulting in UB convergence of the controllers u_i to the feedback-Nash equilibrium controllers $u_i^*\left(x\right)$.

Proof. The derivative of the candidate Lyapunov function in (4–21) along the trajectories of (4–1), (4–9), (4–10), (4–16), and (4–17) is given by

$$\begin{aligned}
\dot{V}_L = & \sum_{i=1}^{N} \left(\nabla V_i^* \left(f + \sum_{j=1}^{N} g_j u_j \right) \right) + \tilde{x}^T \left(Y \tilde{\theta} - k_x \tilde{x} \right) + \tilde{\theta}^T \left(-Y^T \tilde{x} - k_\theta \left(\sum_{j=1}^{M_\theta} Y_j^T Y_j \right) \tilde{\theta} \right) \\
& - \frac{1}{2} \sum_{i=1}^{N} \tilde{W}_{ci}^T \left(\beta_i \Gamma_i^{-1} - \eta_{c1i} \frac{\omega_i \omega_i^T}{\rho_i^2} \right) \tilde{W}_{ci} + \sum_{i=1}^{N} \tilde{W}_{ci}^T \left(\frac{\eta_{c1i} \omega_i}{\rho_i} \hat{\delta}_{ti} + \frac{\eta_{c2i}}{M_{xi}} \sum_{i=1}^{M_{xi}} \frac{\omega_i^k}{\rho_i^k} \hat{\delta}_{ti}^k \right) \\
& - \sum_{i=1}^{N} \tilde{W}_{ai}^T \Bigg(- \eta_{a1i} \left(\hat{W}_{ai}^T - \hat{W}_{ci}^T \right) - \eta_{a2i} \hat{W}_{ai}^T + \frac{1}{4} \sum_{j=1}^{N} \eta_{c1i} \hat{W}_{ci}^T \frac{\omega_i}{\rho_i} \hat{W}_{aj}^T \nabla \sigma_j G_{ij} \nabla \sigma_j^T
\end{aligned}$$

$$+\frac{1}{4}\sum_{k=1}^{M_{xi}}\sum_{j=1}^{N}\frac{\eta_{c2i}}{M_{xi}}\hat{W}_{ci}^{T}\frac{\omega_{i}^{k}}{\rho_{i}^{k}}\hat{W}_{aj}^{T}\nabla\sigma_{j}^{ik}G_{ij}^{ik}\left(\nabla\sigma_{j}^{ik}\right)^{T}\Bigg). \quad (4–26)$$

Substituting the unmeasurable forms of the BEs from (4–19) and (4–20) into (4–26) and using the triangle inequality, the Cauchy-Schwarz inequality and Young's inequality, the Lyapunov derivative in (4–26) can be bounded as

$$\begin{aligned}\dot{V} \leq & -\sum_{i=1}^{N}\frac{\underline{q}_i}{2}\left\Vert x\right\Vert^{2}-\sum_{i=1}^{N}\frac{\eta_{c2i}\underline{c}_{xi}}{2}\left\Vert \tilde{W}_{ci}\right\Vert^{2}-\underline{k_x}\left\Vert \tilde{x}\right\Vert^{2}-\frac{k_{\theta}\underline{y}}{2}\left\Vert \tilde{\theta}\right\Vert^{2}-\sum_{i=1}^{N}\left(\frac{2\eta_{a1i}+\eta_{a2i}}{4}\right)\left\Vert \tilde{W}_{ai}\right\Vert^{2}\\ & +\sum_{i=1}^{N}\iota_{9i}\left\Vert \tilde{W}_{ai}\right\Vert+\sum_{i=1}^{N}\iota_{10i}\left\Vert \tilde{W}_{ci}\right\Vert-\sum_{i=1}^{N}\left(\frac{\underline{q}_i}{2}-\iota_{5i}\right)\left\Vert x\right\Vert^{2}+\sum_{i=1}^{N}\left(\frac{k_{\theta}\underline{y}}{2}-\frac{\iota_{7i}}{\zeta_{1}}-\frac{\iota_{6i}}{\zeta_{3}}\left\Vert x\right\Vert\right)\left\Vert \tilde{\theta}_{i}\right\Vert^{2}\\ & -\sum_{i=1}^{N}\left(\frac{\eta_{c2i}\underline{c}_{xi}}{2}-\iota_{5i}-\zeta_{1}\iota_{7i}-\frac{1}{2}\iota_{2}\zeta_{2}N-\frac{1}{2}\eta_{a1i}-\zeta_{3}\iota_{6i}\left\Vert x\right\Vert\right)\left\Vert \tilde{W}_{ci}\right\Vert^{2}\\ & +\sum_{i=1}^{N}\left(\frac{2\eta_{a1i}+\eta_{a2i}}{4}-\iota_{8}-\frac{\iota_{2}N}{2\zeta_{2}}\right)\left\Vert \tilde{W}_{ai}\right\Vert^{2}+\iota_{3}. \quad (4–27)\end{aligned}$$

Provided the sufficient conditions in (4–24) hold and the conditions

$$\begin{aligned}\frac{\eta_{c2i}\underline{c}_{xi}}{2} & >\iota_{5i}+\zeta_{1}\iota_{7i}+\frac{1}{2}\iota_{2}\zeta_{2}N+\frac{1}{2}\eta_{a1i}+\zeta_{3}\iota_{6i}\left\Vert x\right\Vert,\\ \frac{k_{\theta}\underline{y}}{2} & >\frac{\iota_{7i}}{\zeta_{1}}+\frac{\iota_{6i}}{\zeta_{3}}\left\Vert x\right\Vert \quad (4–28)\end{aligned}$$

hold for all $Z\in\mathcal{Z}$. Completing the squares in (4–27), the bound on the Lyapunov derivative can be expressed as

$$\begin{aligned}\dot{V} & \leq-\sum_{i=1}^{N}\frac{\underline{q}_i}{2}\left\Vert x\right\Vert^{2}-\sum_{i=1}^{N}\frac{\eta_{c2i}\underline{c}_{xi}}{4}\left\Vert \tilde{W}_{ci}\right\Vert^{2}-\underline{k_x}\left\Vert \tilde{x}\right\Vert^{2}-\sum_{i=1}^{N}\left(\frac{2\eta_{a1i}+\eta_{a2i}}{8}\right)\left\Vert \tilde{W}_{ai}\right\Vert^{2}-\frac{k_{\theta}\underline{y}}{2}\left\Vert \tilde{\theta}\right\Vert^{2}+\iota,\\ & \leq-v_{l}\left\Vert Z\right\Vert^{2},\quad\forall\left\Vert Z\right\Vert>\sqrt{\frac{\iota}{v_{l}}},\ Z\in\mathcal{Z}. \quad (4–29)\end{aligned}$$

Using (4–22), (4–25), and (4–29), Theorem 4.18 in [149] can be invoked to conclude that $\limsup_{t\to\infty}\left\Vert Z\left(t\right)\right\Vert \leq \underline{v}^{-1}\left(\overline{v}\left(\sqrt{\frac{\iota}{v_{l}}}\right)\right)$. Furthermore, the system trajectories are bounded as $\left\Vert Z\left(t\right)\right\Vert\leq\overline{Z}$ for all $t\in\mathbb{R}_{\geq0}$. Hence, the conditions in (4–24) are sufficient for the conditions in (4–28) to hold for all $t\in\mathbb{R}_{\geq0}$.

The error between the feedback-Nash equilibrium controller and the approximate controller can be expressed as

$$\left\| u_i^* \left(x \left(t \right) \right) - u_i \left(t \right) \right\| \leq \frac{1}{2} \left\| R_{ii} \right\| \overline{g_i} \overline{\nabla \sigma_i} \left(\left\| \tilde{W}_{ai} \left(t \right) \right\| + \overline{\nabla \epsilon_i} \right),$$

for all $i = 1, .., N$, where $\overline{g_i} \triangleq \sup_{x^o} \left\| g_i \left(x^o \right) \right\|$. Since the weights $\tilde{W}_{ai}$ are UB, UB convergence of the approximate controllers to the feedback-Nash equilibrium controller is obtained. □

Remark 4.1. The closed-loop system analyzed using the candidate Lyapunov function in (4–21) is a switched system. The switching happens when the history stack is updated and when the least-squares regression matrices Γ_i reach their saturation bound. Similar to least squares-based adaptive control (cf. [91]), (4–21) can be shown to be a common Lyapunov function for the regression matrix saturation, and the use of a singular value maximizing algorithm to update the history stack ensures that (4–21) is a common Lyapunov function for the history stack updates (cf. [93]). Since (4–21) is a common Lyapunov function, (4–22), (4–25), and (4–29) establish UB convergence of the switched system.

4.4 Simulation

4.4.1 Problem Setup

To portray the performance of the developed approach, the concurrent learning-based adaptive technique is applied to the nonlinear control-affine system [112]

$$\dot{x} = f\left(x\right) + g_1\left(x\right) u_1 + g_2\left(x\right) u_2, \tag{4–30}$$

where $x \in \mathbb{R}^2$, $u_1, u_2 \in \mathbb{R}$, and

$$f = \left[\begin{array}{c} x_2 - 2x_1 \\ \left(\begin{array}{c} -\frac{1}{2}x_1 - x_2 + \frac{1}{4}x_2 \left(\cos\left(2x_1\right) + 2\right)^2 \\ +\frac{1}{4}x_2 \left(\sin\left(4x_1^2\right) + 2\right)^2 \end{array} \right) \end{array} \right],$$

$$g_1 = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}, \; g_2 = \begin{bmatrix} 0 \\ \sin(4x_1^2) + 2 \end{bmatrix}.$$

The value function has the structure shown in (4–3) with the weights $Q_1 = 2Q_2 = 2I_2$ and $R_{11} = R_{12} = 2R_{21} = 2R_{22} = 2$. The system identification protocol given in Section 4.2.1 and the concurrent learning-based scheme given in Section 4.2.2 are implemented simultaneously to provide an approximate online feedback-Nash equilibrium solution to the given nonzero-sum two-player game.

4.4.2 Analytical Solution

The control-affine system in (4–30) is selected for this simulation because it is constructed using the converse HJ approach [12] such that the analytical feedback-Nash equilibrium solution of the nonzero-sum game is

$$V_1^* = \begin{bmatrix} 0.5 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}, \quad V_2^* = \begin{bmatrix} 0.25 \\ 0 \\ 0.5 \end{bmatrix}^T \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix},$$

and the feedback-Nash equilibrium control policies for player 1 and player 2 are

$$u_1^* = -\frac{1}{2} R_{11}^{-1} g_1^T \begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0 & 2x_2 \end{bmatrix}^T \begin{bmatrix} 0.5 \\ 0 \\ 1 \end{bmatrix}, \; u_2^* = -\frac{1}{2} R_{22}^{-1} g_2^T \begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0 & 2x_2 \end{bmatrix}^T \begin{bmatrix} 0.25 \\ 0 \\ 0.5 \end{bmatrix}.$$

Since the analytical solution is available, the performance of the developed method can be evaluated by comparing the obtained approximate solution against the analytical solution.

4.4.3 Simulation Parameters

The dynamics are linearly parameterized as $f(x) = Y(x)\theta$, where

$$Y(x) = \begin{bmatrix} x_2 & x_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_1 & x_2 & x_2 (\cos(2x_1) + 2)^2 & x_2 (\cos(2x_1) + 2)^2 \end{bmatrix}$$

Table 4-1. Learning gains for for value function approximation

	Player 1	Player 2
ν	0.005	0.005
η_{c1}	1.0	1.0
η_{c2}	1.5	1.0
η_{a1}	10.0	10.0
η_{a2}	0.1	0.1
β	3.0	3.0
$\bar{\Gamma}$	10,000.0	10,000.0

is known and the constant vector of parameters $\theta = \left[1, -2, -\frac{1}{2}, -1, \frac{1}{4}, -\frac{1}{4}\right]^T$ is assumed to be unknown. The initial guess for θ is selected as $\hat{\theta}\left(t_0\right) = 0.5 \times \mathbf{1}_{6\times1}$. The system identification gains are selected as $k_x = 5$, $\Gamma_\theta = \text{diag}\left(20, 20, 100, 100, 60, 60\right)$, $k_\theta = 1.5$. A history stack of 30 points is selected using a singular value maximizing algorithm (cf. [93]) for the concurrent learning-based update law in (4–8), and the state derivatives are estimated using a fifth order Savitzky-Golay filter (cf. [150]). Based on the structure of the feedback-Nash equilibrium value functions, the basis function for value function approximation is selected as $\sigma = [x_1^2, x_1x_2, x_2^2]^T$, and the adaptive learning parameters and initial conditions are shown for both players in Tables 4-1 and 4-2. Twenty-five points lying on a 5×5 grid on a 2×2 square around the origin are selected for the concurrent learning-based update laws in (4–16) and (4–17).

Table 4-2. Initial conditions for the system and the two players

	Player 1	Player 2
$\hat{W}_c\left(t_0\right)$	$[3,3,3]^T$	$[3,3,3]^T$
$\hat{W}_a\left(t_0\right)$	$[3,3,3]^T$	$[3,3,3]^T$
$\Gamma\left(t_0\right)$	$100I_3$	$100I_3$
$x\left(t_0\right)$	$[1,1]^T$	$[1,1]^T$
$\hat{x}\left(t_0\right)$	$[0,0]^T$	$[0,0]^T$

4.4.4 Simulation Results

Figures 4-1 and 4-2 show the rapid convergence of the actor and critic weights to the approximate feedback-Nash equilibrium values for both players, resulting in the value functions and control policies

$$V_1(x) = \begin{bmatrix} 0.5021 \\ -0.0159 \\ 0.9942 \end{bmatrix}^T \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}, \; \overline{u}_1(x) = -\frac{1}{2} R_{11}^{-1} g_1^T \begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0 & 2x_2 \end{bmatrix}^T \begin{bmatrix} 0.4970 \\ -0.0137 \\ 0.9810 \end{bmatrix},$$

$$V_2(x) = \begin{bmatrix} 0.2510 \\ -0.0074 \\ 0.4968 \end{bmatrix}^T \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}, \; \overline{u}_2(x) = -\frac{1}{2} R_{22}^{-1} g_2^T \begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0 & 2x_2 \end{bmatrix}^T \begin{bmatrix} 0.2485 \\ -0.0055 \\ 0.4872 \end{bmatrix}.$$

Figure 4-3 demonstrates that (without the injection of a PE signal) the system identification parameters also approximately converged to the correct values. The state and control signal trajectories are displayed in Figure 4-4.

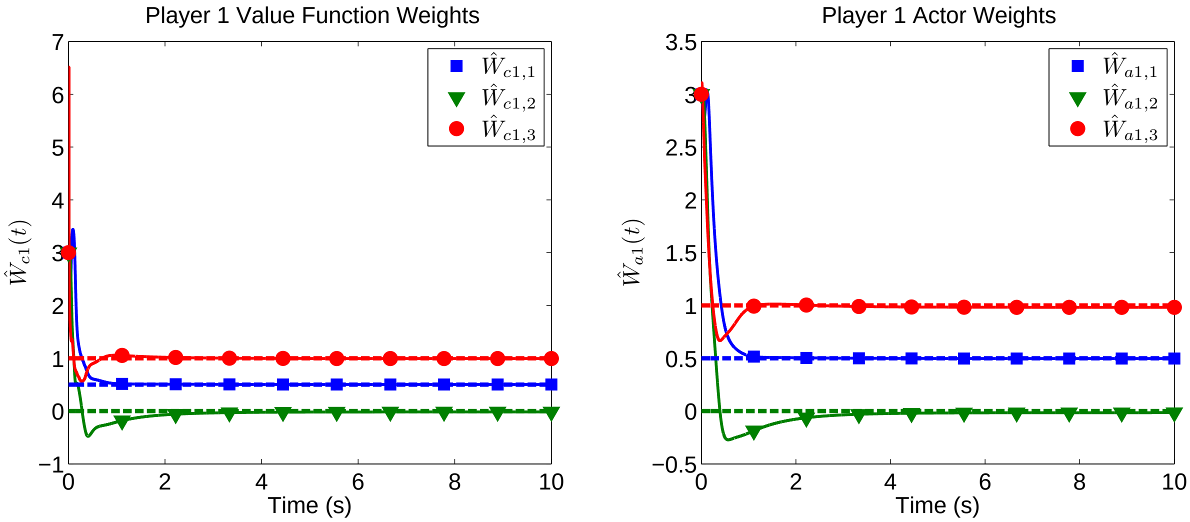

Figure 4-1. Trajectories of actor and critic weights for player 1 compared against their true values. The true values computed based on the analytical solution are represented by dotted lines.

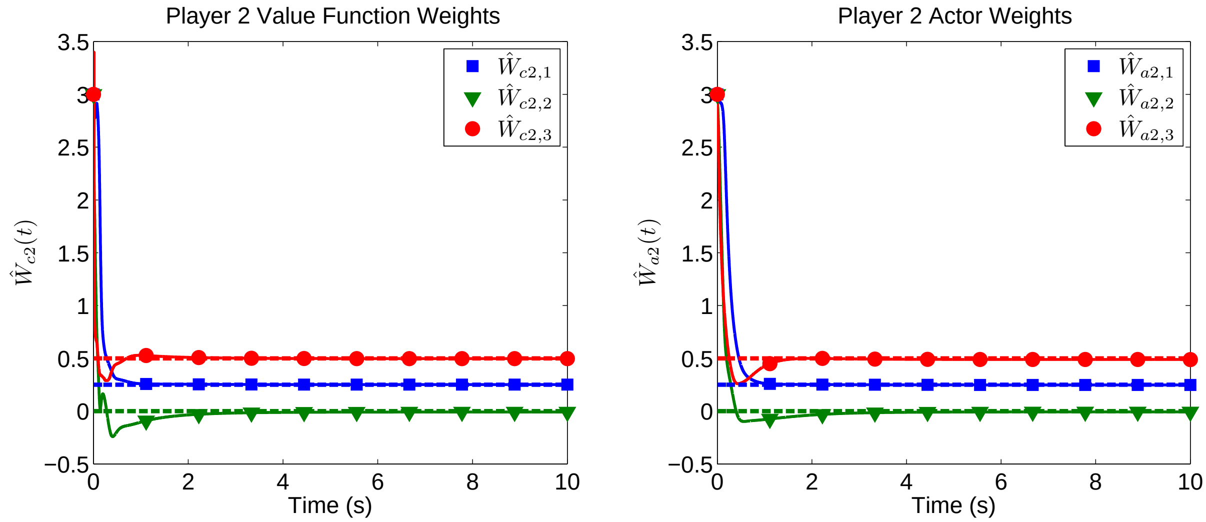

Figure 4-2. Trajectories of actor and critic weights for player 2 compared against their true values. The true values computed based on the analytical solution are represented by dotted lines.

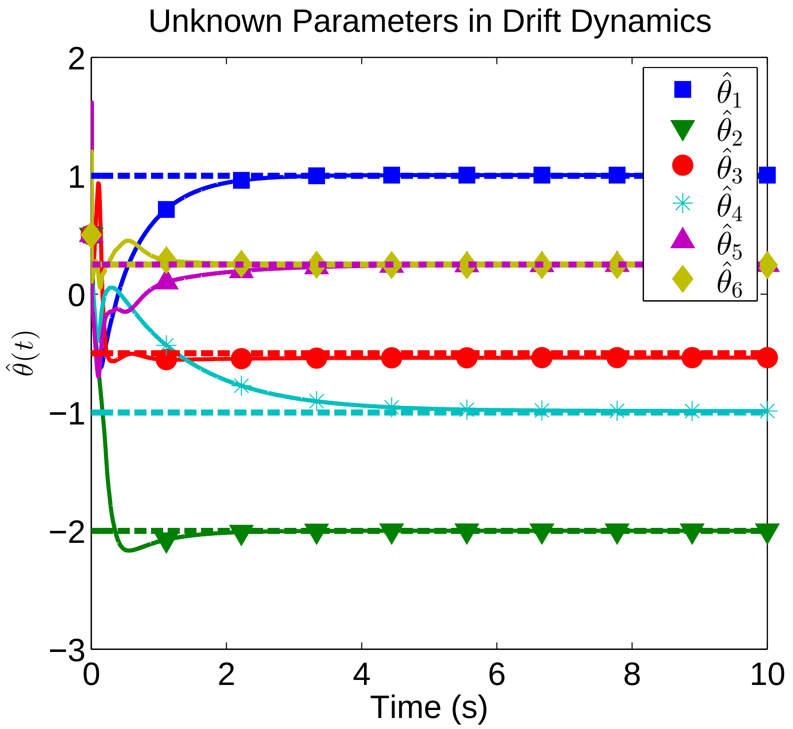

Figure 4-3. Trajectories of the estimated parameters in the drift dynamics compared against their true values. The true values are represented by dotted lines.

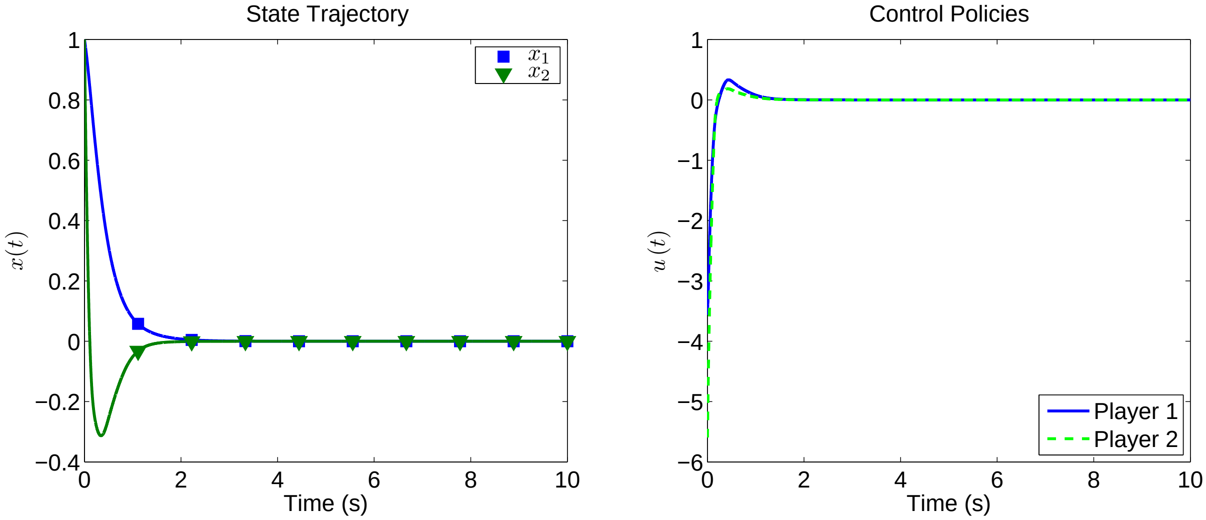

Figure 4-4. System state trajectory and the control trajectories for players 1 and 2 generated using the developed technique

4.5 Concluding Remarks

A concurrent learning-based adaptive approach is developed to determine the feedback-Nash equilibrium solution to an N-player nonzero-sum game online. The solutions to the associated coupled HJ equations and the corresponding feedback-Nash equilibrium policies are approximated using parametric universal function approximators. Based on estimates of the unknown drift parameters, estimates for the Bellman errors are evaluated at a set of preselected points in the state-space. The value function and the policy weights are updated using a concurrent learning-based least-squares approach to minimize the instantaneous BEs and the BEs evaluated at the preselected points. Simultaneously, the unknown parameters in the drift dynamics are updated using a history stack of recorded data via a concurrent learning-based gradient descent approach.

The simulation-based ACI technique developed in this chapter and Chapter 3 achieves approximate optimal control for autonomous system and stationary cost functions. Extension of the ACI techniques to optimal trajectory tracking problems presents unique challenges for value function approximation due to the time-varying nature of the problem. The following chapter describes the challenges and presents a solution to extend the ACI architecture to solve infinite-horizon trajectory tracking problems.

CHAPTER 5
EXTENSION TO APPROXIMATE OPTIMAL TRACKING

ADP has been investigated and used as a tool to approximately solve optimal regulation problems. For these problems, function approximation techniques can be used to approximate the value function because it is a time invariant function. In tracking problems, the tracking error, and hence the value function, is a function of the state and an explicit function of time. Approximation techniques like NNs are commonly used in ADP literature for value function approximation. However, NNs can only approximate functions on compact domains, thus leading to a technical challenge to approximate the value function for a tracking problem because the infinite-horizon nature of the problem implies that time does not lie on a compact set. Hence, the extension of this technique to optimal tracking problems for continuous-time nonlinear systems has remained a non-trivial open problem.

In this result, the tracking error and the desired trajectory both serve as inputs to the NN. This makes the developed controller fundamentally different from previous results, in the sense that a different HJB equation must be solved and its solution, i.e. the feedback component of the controller, is a time-varying function of the tracking error. In particular, this chapter addresses the technical obstacles that result from the time-varying nature of the optimal control problem by including the partial derivative of the value function with respect to the desired trajectory in the HJB equation, and by using a system transformation to convert the problem into a time-invariant optimal control problem in such a way that the resulting value function is a time-invariant function of the transformed states, and hence, lends itself to approximation using a NN. A Lyapunov-based analysis is used to prove ultimately bounded tracking and that the controller converges to the approximate optimal policy. Simulation results are presented to demonstrate the applicability of the presented technique. To gauge the performance of the proposed method, a comparison with a numerical optimal solution is presented.

5.1 Formulation of Time-invariant Optimal Control Problem

Consider the class of nonlinear control-affine systems described in (2–1). The control objective is to track a bounded continuously differentiable signal $x_d \in \mathbb{R}^n$. To quantify this objective, a tracking error is defined as $e \triangleq x - x_d$. The open-loop tracking error dynamics can then be written as

$$\dot{e} = f\left(x\right) + g\left(x\right) u - \dot{x}_d. \tag{5–1}$$

The following assumptions are made to facilitate the formulation of an approximate optimal tracking controller.

Assumption 5.1. The function g is bounded, the matrix $g\left(x^o\right)$ has full column rank for all $x^o \in \mathbb{R}^n$, and the function $g^+ : \mathbb{R}^n \to \mathbb{R}^{m \times n}$ defined as $g^+ \triangleq \left(g^T g\right)^{-1} g^T$ is bounded and locally Lipschitz.

Assumption 5.2. The desired trajectory is bounded such that $\left\|x_d\right\| \leq d \in \mathbb{R}$, and there exists a locally Lipschitz function $h_d : \mathbb{R}^n \to \mathbb{R}^n$ such that $\dot{x}_d = h_d\left(x_d\right)$ and $g\left(x_d\right) g^+\left(x_d\right)\left(h_d\left(x_d\right) - f\left(x_d\right)\right) = h_d\left(x_d\right) - f\left(x_d\right)$, $\forall t \in \mathbb{R}_{\geq t_0}$.

The steady-state control policy $u_d : \mathbb{R}^n \to \mathbb{R}^m$ corresponding to the desired trajectory x_d is

$$u_d\left(x_d\right) = g_d^+\left(h_d\left(x_d\right) - f_d\right), \tag{5–2}$$

where $g_d^+ \triangleq g^+\left(x_d\right)$ and $f_d \triangleq f\left(x_d\right)$. To transform the time-varying optimal control problem into a time-invariant optimal control problem, a new concatenated state $\zeta \in \mathbb{R}^{2n}$ is defined as [86]

$$\zeta \triangleq \left[e^T, x_d^T\right]^T. \tag{5–3}$$

Based on (5–1) and Assumption 5.2, the time derivative of (5–3) can be expressed as

$$\dot{\zeta} = F\left(\zeta\right) + G\left(\zeta\right) \mu, \tag{5–4}$$

where the functions $F:\mathbb{R}^{2n}\to\mathbb{R}^{2n}$, $G:\mathbb{R}^{2n}\to\mathbb{R}^{2n\times m}$, and the control $\mu\in\mathbb{R}^{m}$ are defined as

$$F\left(\zeta\right)\triangleq\begin{bmatrix}f\left(e+x_{d}\right)-h_{d}\left(x_{d}\right)+g\left(e+x_{d}\right)u_{d}\left(x_{d}\right)\\ h_{d}\left(x_{d}\right)\end{bmatrix},\quad G\left(\zeta\right)\triangleq\begin{bmatrix}g\left(e+x_{d}\right)\\ \mathbf{0}_{n\times m}\end{bmatrix},\quad \mu\triangleq u-u_{d}\left(x_{d}\right). \tag{5–5}$$

Local Lipschitz continuity of f and g, the fact that $f\left(0\right)=0$, and Assumption 5.2 imply that $F\left(0\right)=0$ and F is locally Lipschitz. The objective of the optimal control problem is to minimize the cost functional $J\left(\zeta,\mu\right)$, introduced in (2–2), subject to the dynamic constraints in (5–4) while tracking the desired trajectory. For ease of exposition, let the function $Q:\mathbb{R}^{2n}\to\mathbb{R}_{\geq 0}$ in (2–3) be defined as $Q\left(\zeta\right)\triangleq\zeta^{T}\overline{Q}\zeta$, where $\overline{Q}\in\mathbb{R}^{2n\times 2n}$ is a constant matrix defined as

$$\overline{Q}\triangleq\begin{bmatrix}Q & \mathbf{0}_{n\times n}\\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n}\end{bmatrix}, \tag{5–6}$$

where $Q\in\mathbb{R}^{n\times n}$ is a positive definite symmetric matrix of constants with the minimum eigenvalue $\underline{q}\in\mathbb{R}_{>0}$. Thus, the reward $r:\mathbb{R}^{2n}\times\mathbb{R}^{m}\to\mathbb{R}$ is given by

$$r\left(\zeta,\mu\right)\triangleq\zeta^{T}\overline{Q}\zeta+\mu^{T}R\mu. \tag{5–7}$$

5.2 Approximate Optimal Solution

Similar to the development in Chapter 2, assuming that a minimizing policy exists and assuming that the optimal value function satisfies $V^{*}\in C^{1}$ and $V^{*}\left(0\right)=0$, the local cost in (5–7) and the dynamics in (5–4), yield the optimal policy $\mu^{*}:\mathbb{R}^{2n}\to\mathbb{R}^{m}$ as

$$\mu^{*}\left(\zeta^{o}\right)=-\frac{1}{2}R^{-1}G^{T}\left(\zeta^{o}\right)\left(\nabla V^{*}\left(\zeta^{o}\right)\right)^{T},\ \forall\zeta^{o}\in\mathbb{R}^{2n} \tag{5–8}$$

where $V^* : \mathbb{R}^{2n} \to \mathbb{R}_{\geq 0}$ denotes the optimal value function defined as in (2–4) with the local cost defined in (5–7).[1] The policy in (5–8) and the value function V^* satisfy the HJB equation [1]

$$\nabla V^* \left(\zeta^o\right) \left(F\left(\zeta^o\right) + G\left(\zeta^o\right) \mu^* \left(\zeta^o\right)\right) + r\left(\zeta^o, \mu^* \left(\zeta^o\right)\right) = 0, \tag{5–9}$$

$\forall \zeta^o \in \mathbb{R}^{2n}$, with the initial condition $V^*(0) = 0$.

The value function V^* can be represented using a NN with L neurons as

$$V^* \left(\zeta^o\right) = W^T \sigma\left(\zeta^o\right) + \epsilon\left(\zeta^o\right), \; \forall \zeta^o \in \mathbb{R}^{2n} \tag{5–10}$$

where $W \in \mathbb{R}^L$ is the ideal weight matrix bounded above by a known positive constant $\overline{W} \in \mathbb{R}$ in the sense that $\|W\| \leq \overline{W}$, $\sigma : \mathbb{R}^{2n} \to \mathbb{R}^L$ is a bounded continuously differentiable nonlinear activation function, and $\epsilon : \mathbb{R}^{2n} \to \mathbb{R}$ is the function reconstruction error [151, 152].

Using (5–8) and (5–10) the optimal policy can be represented as

$$\mu^* \left(\zeta^o\right) = -\frac{1}{2} R^{-1} G^T \left(\zeta^o\right) \left(\nabla \sigma^T \left(\zeta^o\right) W + \nabla \epsilon^T \left(\zeta^o\right)\right), \; \forall \zeta^o \in \mathbb{R}^{2n}. \tag{5–11}$$

Based on (5–10) and (5–11), the NN approximations to the optimal value function and the optimal policy are defined as

$$\hat{V}\left(\zeta, \hat{W}_c\right) \triangleq \hat{W}_c^T \sigma\left(\zeta\right), \qquad \hat{\mu}\left(\zeta, \hat{W}_a\right) \triangleq -\frac{1}{2} R^{-1} G^T \left(\zeta\right) \nabla \sigma^T \left(\zeta\right) \hat{W}_a, \tag{5–12}$$

where $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ are estimates of the ideal neural network weights W. The use of two separate sets of weight estimates $\hat{W}_a$ and $\hat{W}_c$ for W is motivated by the fact that the BE is linear with respect to the value function weight estimates and nonlinear

[1] Since the closed-loop system corresponding to (5–4) under a feedback policy is autonomous, the cost-to-go, i.e., the integral in (2–5) is independent of initial time. Hence, the value function is only a function of ζ.

with respect to the policy weight estimates. Use of a separate set of weight estimates for the value function facilitates least squares-based adaptive updates.

The controller for the dynamics in (5–4) is $\mu(t) = \hat{\mu}\left(\zeta(t), \hat{W}_a(t)\right)$, and the controller implemented on the actual system is obtained from (5–2), (5–5), and (5–12) as

$$u = -\frac{1}{2} R^{-1} G^T(\zeta) \nabla\sigma^T(\zeta) \hat{W}_a + g_d^+ \left(h_d(x_d) - f_d\right). \tag{5–13}$$

Using the approximations $\hat{\mu}$ and $\hat{V}$ in (5–9) for μ^* and V^*, respectively, the BE in (2–12), is given in a measurable form by

$$\delta_t = \nabla_\zeta \hat{V}\left(\zeta, \hat{W}_c\right)\dot{\zeta} + r(\zeta, \mu), \tag{5–14}$$

where the derivative $\dot{\zeta}$ is measurable because the system model is known. For notational brevity, state-dependence of the functions h_d, F, G, V^*, μ^*, σ, and ϵ and the arguments to the functions $\hat{\mu}$, and $\hat{V}$ are suppressed hereafter. The value function weights are updated to minimize $\int_0^t \delta_t^2(\rho)\, d\rho$ using a normalized least squares update law[2] with an exponential forgetting factor as [91]

$$\begin{aligned} \dot{\hat{W}}_c &= -\eta_c \Gamma \frac{\omega}{1 + \nu\omega^T\Gamma\omega}\delta_t, \\ \dot{\Gamma} &= -\eta_c\left(-\lambda\Gamma + \Gamma\frac{\omega\omega^T}{1 + \nu\omega^T\Gamma\omega}\Gamma\right), \end{aligned} \tag{5–15}$$

where $\nu, \eta_c \in \mathbb{R}$ are positive adaptation gains, $\omega \in \mathbb{R}^L$ is defined as $\omega \triangleq \nabla\sigma\dot{\zeta}$, and $\lambda \in (0, 1)$ is the forgetting factor for the estimation gain matrix $\Gamma \in \mathbb{R}^{L\times L}$. The policy

[2] The least-squares approach is motivated by faster convergence. With minor modifications to the stability analysis, the result can also be established for a gradient descent update law.

weights are updated to follow the critic weights[3] as

$$\dot{\hat{W}}_a = -\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a, \tag{5–16}$$

where $\eta_{a1}, \eta_{a2} \in \mathbb{R}$ are positive adaptation gains. The following assumption facilitates the stability analysis using PE.

Assumption 5.3. The regressor $\psi : \mathbb{R}_{\geq 0} \to \mathbb{R}^L$ defined as $\psi \triangleq \frac{\omega}{\sqrt{1+\nu\omega^T\Gamma\omega}}$ satisfies the PE condition, i.e., there exist constants $T, \underline{\psi} \in \mathbb{R}_{>0}$ such that $\underline{\psi}I \leq \int_t^{t+T} \psi\left(\tau\right)\psi\left(\tau\right)^T d\tau.$[4]

Using Assumption 5.3 and [91, Corollary 4.3.2] it can be concluded that

$$\underline{\varphi}I_{L\times L} \leq \Gamma\left(t\right) \leq \overline{\varphi}I_{L\times L},\ \forall t \in \mathbb{R}_{\geq 0} \tag{5–17}$$

where $\overline{\varphi}, \underline{\varphi} \in \mathbb{R}$ are constants such that $0 < \underline{\varphi} < \overline{\varphi}$.[5] Based on (5–17), the regressor vector can be bounded as

$$\left\|\psi\left(t\right)\right\| \leq \frac{1}{\sqrt{\nu\underline{\varphi}}},\ \forall t \in \mathbb{R}_{\geq 0}. \tag{5–18}$$

Using (5–10), (5–11), and (5–14), an unmeasurable form of the BE can be written as

$$\delta_t = -\tilde{W}_c^T\omega + \frac{1}{4}\tilde{W}_a^T\mathcal{G}_\sigma\tilde{W}_a + \frac{1}{4}\nabla\epsilon\mathcal{G}\nabla\epsilon^T + \frac{1}{2}W^T\nabla\sigma\mathcal{G}\nabla\epsilon^T - \nabla\epsilon F, \tag{5–19}$$

[3] The least-squares approach cannot be used to update the policy weights because the BE is a nonlinear function of the policy weights.

[4] The regressor is defined here as a trajectory indexed by time. This definition suppresses the fact that different initial conditions result in different regressor trajectories. Assumption 5.3 describes the properties of one specific trajectory starting from one specific initial condition. Naturally, the final result of the chapter also describes limiting properties of one specific state trajectory. That is, the final result is not uniform in the initial conditions.

[5] Since the evolution of ψ is dependent on the initial condition, the constants $\overline{\varphi}$ and $\underline{\varphi}$ depend on the initial condition.

where $\mathcal{G} \triangleq GR^{-1}G^T$ and $\mathcal{G}_\sigma \triangleq \nabla\sigma GR^{-1}G^T\nabla\sigma^T$. The weight estimation errors for the value function and the policy are defined as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$, respectively.

5.3 Stability Analysis

Before stating the main result of the chapter, three supplementary technical lemmas are stated. To facilitate the discussion, let $\mathcal{Y} \in \mathbb{R}^{2n+2L}$ be a compact set, and let $\mathcal{Z} \triangleq \mathcal{Y} \cap \mathbb{R}^{n+2L}$. Using the universal approximation property of NNs, on the compact set $\mathcal{Y} \cap \mathbb{R}^{2n}$, the NN approximation errors can be bounded such that $\sup_{\zeta^o \in \mathcal{Y}\cap\mathbb{R}^{2n}} |\epsilon(\zeta^o)| \leq \bar{\epsilon}$ and $\sup_{\zeta^o \in \mathcal{Y}\cap\mathbb{R}^{2n}} |\nabla\epsilon(\zeta^o)| \leq \overline{\nabla\epsilon}$, where $\bar{\epsilon} \in \mathbb{R}$ and $\overline{\nabla\epsilon} \in \mathbb{R}$ are positive constants. Using Assumptions 5.1 and 5.2 and the fact that on the compact set $\mathcal{Y} \cap \mathbb{R}^{2n}$, there exists a positive constant $L_F \in \mathbb{R}$ such that[6] $\sup_{\zeta^o \in \mathcal{Y}\cap\mathbb{R}^{2n}} \|F(\zeta^o)\| \leq L_F \|\zeta^o\|$, the following bounds are developed to aid the subsequent stability analysis:

$$\left\| \left(\frac{\nabla\epsilon}{4} + \frac{W^T\nabla\sigma}{2} \right) \mathcal{G}\nabla\epsilon^T \right\| + \overline{\nabla\epsilon} L_F \|x_d\| \leq \iota_1, \quad \|\mathcal{G}_\sigma\| \leq \iota_2, \quad \left\|\nabla\epsilon\mathcal{G}\nabla\epsilon^T\right\| \leq \iota_3,$$
$$\left\| \frac{1}{2}W^T\mathcal{G}_\sigma + \frac{1}{2}\nabla\epsilon\mathcal{G}\nabla\sigma^T \right\| \leq \iota_4, \quad \left\| \frac{1}{4}\nabla\epsilon\mathcal{G}\nabla\epsilon^T + \frac{1}{2}W^T\nabla\sigma\mathcal{G}\nabla\epsilon^T \right\| \leq \iota_5, \tag{5–20}$$

where $\iota_1, \iota_2, \iota_3, \iota_4, \iota_5 \in \mathbb{R}$ are positive bounds that are constant for a fixed initial condition.

5.3.1 Supporting Lemmas

The contribution in the previous section was the development of a transformation that enables the optimal policy and the optimal value function to be expressed as a time-invariant function of ζ. The use of this transformation presents a challenge in the sense that the optimal value function, which is used as the Lyapunov function for the stability analysis, is not a positive definite function of ζ, because the matrix $\overline{Q}$ is positive

[6] Instead of using the fact that locally Lipschitz functions on compact sets are Lipschitz, it is possible to bound the function F as $\|F(\zeta)\| \leq \rho(\|\zeta\|)\|\zeta\|$, where $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is non-decreasing. This approach is feasible and results in additional gain conditions.

semi-definite. In this section, this technical obstacle is addressed by exploiting the fact that the time-invariant optimal value function $V^* : \mathbb{R}^{2n} \to \mathbb{R}$ can be interpreted as a time-varying map $V_t^* : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$, such that

$$V_t^*\left(e,t\right) \triangleq V^*\left(\left[\begin{array}{c} e \\ x_d\left(t\right) \end{array}\right]\right) \tag{5–21}$$

for all $e \in \mathbb{R}^n$ and for all $t \in \mathbb{R}_{\geq 0}$. Specifically, the time-invariant form facilitates the development of the approximate optimal policy, whereas the equivalent time-varying form can be shown to be a positive definite and decrescent function of the tracking error. In the following, Lemma 5.1 is used to prove that $V_t^* : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ is positive definite and decrescent, and hence, a candidate Lyapunov function.

Lemma 5.1. *Let B_a denote a closed ball around the origin with the radius $a \in \mathbb{R}_{>0}$. The optimal value function $V_t^* : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ satisfies the following properties*

$$V_t^*\left(e,t\right) \geq \underline{v}\left(\|e\|\right), \tag{5–22a}$$

$$V_t^*\left(0,t\right) = 0, \tag{5–22b}$$

$$V_t^*\left(e,t\right) \leq \overline{v}\left(\|e\|\right), \tag{5–22c}$$

$\forall t \in \mathbb{R}_{\geq 0}$ and $\forall e \in B_a$ where $\underline{v} : [0,a] \to \mathbb{R}_{\geq 0}$ and $\overline{v} : [0,a] \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions.

Proof. See Appendix B.1. □

Since the stability analysis is subject to the PE condition in Assumption 5.3, the behavior of the system states is examined over the time interval $[t, t+T]$. The following two lemmas establish growth bounds on the tracking error and the actor and the critic weights.

Lemma 5.2. *Let $Z \triangleq \begin{bmatrix} e^T & \tilde{W}_c^T & \tilde{W}_a^T \end{bmatrix}^T$, and suppose that $Z\left(\tau\right) \in \mathcal{Z}$, for all $\tau \in [t, t+T]$. Then, the NN weights and the tracking errors satisfy*

$$-\inf_{\tau\in[t,t+T]} \|e\left(\tau\right)\|^2 \leq -\varpi_0 \sup_{\tau\in[t,t+T]} \|e\left(\tau\right)\|^2 + \varpi_1 T^2 \sup_{\tau\in[t,t+T]} \left\|\tilde{W}_a\left(\tau\right)\right\|^2 + \varpi_2 \tag{5–23}$$

$$-\inf_{\tau\in[t,t+T]}\left\Vert \tilde{W}_a(\tau)\right\Vert^2 \leq -\varpi_3 \sup_{\tau\in[t,t+T]}\left\Vert \tilde{W}_a(\tau)\right\Vert^2 + \varpi_4 \inf_{\tau\in[t,t+T]}\left\Vert \tilde{W}_c(\tau)\right\Vert^2 + \varpi_5 \sup_{\tau\in[t,t+T]}\left\Vert e(\tau)\right\Vert^2 + \varpi_6, \quad \text{(5–24)}$$

where

$$\varpi_1 = \tfrac{3n}{4}\sup_{t\in\mathbb{R}_{\geq t_o}}\left\Vert gR^{-1}G^T\nabla\sigma^T\right\Vert^2,\ \varpi_6 = \frac{18\left(L\eta_{a1}\eta_c\overline{\varphi}\left(\overline{\nabla\epsilon}L_F d+\iota_5\right)T^2\right)^2}{\nu\underline{\varphi}\left(1-6L(\eta_c\overline{\varphi}T)^2/\left(\nu\underline{\varphi}\right)^2\right)} + 3L\left(\eta_{a2}\overline{W}T\right)^2,$$

$$\varpi_3 = \frac{\left(1-6L(\eta_{a1}+\eta_{a2})^2T^2\right)}{2},\ \varpi_4 = \frac{6L\eta_{a1}^2T^2}{\left(1-6L(\eta_c\overline{\varphi}T)^2/\left(\nu\underline{\varphi}\right)^2\right)},\ \varpi_5 = \frac{18\left(\eta_{a1}L\eta_c\overline{\varphi}\overline{\nabla\epsilon}L_FT^2\right)^2}{\nu\underline{\varphi}\left(1-6L(\eta_c\overline{\varphi}T)^2/\left(\nu\underline{\varphi}\right)^2\right)},$$

$$\varpi_0 = \frac{\left(1-6nT^2L_F^2\right)}{2},\ \varpi_2 = \frac{3n^2T^2\left(dL_F+\sup_t\left\Vert gg_d^+(h_d-f_d)-\frac{1}{2}gR^{-1}G^T\nabla\sigma^TW-h_d\right\Vert\right)^2}{n}.$$

Proof. See Appendix B.2. □

Lemma 5.3. *Let* $Z \triangleq \begin{bmatrix} e^T & \tilde{W}_c^T & \tilde{W}_a^T\end{bmatrix}^T$*, and suppose that* $Z(\tau)\in\mathcal{Z}$*, for all* $\tau\in[t,t+T]$*. Then, the critic weights satisfy*

$$-\int_t^{t+T}\left\Vert \tilde{W}_c^T\psi\right\Vert^2 d\tau \leq -\underline{\psi}\varpi_7\left\Vert \tilde{W}_c\right\Vert^2 + \varpi_8\int_t^{t+T}\Vert e\Vert^2 d\tau + 3\iota_2^2\int_t^{t+T}\left\Vert \tilde{W}_a(\sigma)\right\Vert^4 d\sigma + \varpi_9 T, \quad \text{(5–25)}$$

where $\varpi_7 = \frac{\nu^2\underline{\varphi}^2}{2\left(\nu^2\underline{\varphi}^2+\eta_c^2\overline{\varphi}^2T^2\right)}$, $\varpi_8 = 3\overline{\epsilon}'^2L_F^2$, and $\varpi_9 = 2\left(\iota_5^2+\overline{\epsilon}'^2L_F^2d^2\right)$.

Proof. See Appendix B.3.

□

5.3.2 Gain Conditions and Gain Selection

This section details sufficient gain conditions derived based on a stability analysis performed using the candidate Lyapunov function $V_L : \mathbb{R}^{n+2L}\times\mathbb{R}_{\geq 0}\to\mathbb{R}$ defined as $V_L(Z,t)\triangleq V_t^*(e,t)+\frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c+\frac{1}{2}\tilde{W}_a^T\tilde{W}_a$. Using (5–17) and Lemma 5.1,

$$\underline{v_l}\left(\Vert Z^o\Vert\right)\leq V_L\left(Z^o,t\right)\leq\overline{v_l}\left(\Vert Z^o\Vert\right), \quad \text{(5–26)}$$

$\forall Z^o\in B_b,\ \forall t\in\mathbb{R}_{\geq 0}$, where $\underline{v_l}:[0,b]\to\mathbb{R}_{\geq 0}$ and $\overline{v_l}:[0,b]\to\mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions, and $B_b\subset\mathbb{R}^{n+2L}$ denotes a ball of radius $b\in\mathbb{R}_{>0}$ around the origin, containing $\mathcal{Z}$.

To facilitate the discussion, define $\eta_{a12} \triangleq \eta_{a1} + \eta_{a2}$, $\iota \triangleq \frac{\left(\eta_{a2}\overline{W}+\iota_4\right)^2}{\eta_{a12}} + 2\eta_c\left(\iota_1\right)^2 + \frac{1}{4}\iota_3$, $\varpi_{10} \triangleq \frac{\varpi_6\eta_{a12}+2\varpi_2\underline{q}+\eta_c\varpi_9}{8} + \iota$, $Z \triangleq \begin{bmatrix} e^T & \tilde{W}_c^T & \tilde{W}_a^T \end{bmatrix}^T$, and $\varpi_{11} \triangleq \frac{1}{16}\min(\eta_c\underline{\psi}\varpi_7,\ 2\varpi_0\underline{q}T,\ \varpi_3\eta_{a12}T)$. Let $Z_0 \in \mathbb{R}_{\geq 0}$ denote a known constant bound on the initial condition such that $\left\Vert Z\left(t_0\right)\right\Vert \leq Z_0$, and let

$$\overline{Z} \triangleq \underline{v_l}^{-1}\left(\overline{v_l}\left(\max\left(\left\Vert Z_0\right\Vert, \sqrt{\frac{\varpi_{10}T}{\varpi_{11}}}\right)\right) + \iota T\right). \tag{5–27}$$

The sufficient gain conditions for the subsequent Theorem 5.1 are given by[7]

$$\eta_{a12} > \max\left(\eta_{a1}\xi_2 + \frac{\eta_c\iota_2}{4}\sqrt{\frac{\overline{Z}}{\nu\underline{\varphi}}}, 3\eta_c\iota_2^2\overline{Z}\right), \quad \xi_1 > 2\overline{\nabla\epsilon}L_F, \quad \eta_c > \frac{\eta_{a1}}{\lambda\underline{\gamma}\xi_2}, \quad \underline{\psi} > \frac{2\varpi_4\eta_{a12}}{\eta_c\varpi_7}T,$$
$$\underline{q} > \max\left(\frac{\varpi_5\eta_{a12}}{\varpi_0}, \frac{1}{2}\eta_c\varpi_8, \eta_c L_F\overline{\nabla\epsilon}\xi_1\right),$$
$$T < \min\left(\frac{1}{\sqrt{6L}\eta_{a12}}, \frac{\nu\underline{\varphi}}{\sqrt{6L}\eta_c\overline{\varphi}}, \frac{1}{2\sqrt{n}L_F}, \sqrt{\frac{\eta_{a12}}{6L\eta_{a12}^3 + 8\underline{q}\varpi_1}}\right). \tag{5–28}$$

Furthermore, the compact set $\mathcal{Z}$ satisfies the sufficient condition

$$\overline{Z} \leq r, \tag{5–29}$$

where $r \triangleq \frac{1}{2}\sup_{z,y\in\mathcal{Z}}\left\Vert z-y\right\Vert$ denotes the radius of $\mathcal{Z}$. Since the Lipschitz constant and the bounds on NN approximation error depend on the size of the compact set $\mathcal{Z}$, the constant $\overline{Z}$ depends on r; hence, feasibility of the sufficient condition in (5–29) is not apparent. Algorithm 5.1 details an iterative gain selection process in order to ensure satisfaction of the sufficient condition in (5–29). In Algorithm 5.1, the notation $\left\{\varpi\right\}_i$ for any parameter ϖ denotes the value of ϖ computed in the i^{th} iteration. Algorithm 5.1 ensures satisfaction of the sufficient condition in (5–29).

[7] Similar conditions on $\underline{\psi}$ and T can be found in PE-based adaptive control in the presence of bounded or Lipschitz uncertainties (cf. [153, 154]).

Algorithm 5.1 Gain Selection

First iteration:

Given $Z_0 \in \mathbb{R}_{\geq 0}$ such that $\|Z(t_0)\| < Z_0$, let $\mathcal{Z}_1 = \left\{ \varrho \in \mathbb{R}^{n+2\{L\}_1} \mid \|\varrho\| \leq \beta_1 \underline{v_l}^{-1}\left(\overline{v_l}\left(Z_0\right)\right) \right\}$ for some $\beta_1 > 1$. Using $\mathcal{Z}_1$, compute the bounds in (5–20) and (5–27), and select the gains according to (5–28). If $\left\{\overline{Z}\right\}_1 \leq \beta_1 \underline{v_l}^{-1}\left(\overline{v_l}\left(\|Z_0\|\right)\right)$, set $\mathcal{Z} = \mathcal{Z}_1$ and terminate.

Second iteration:

If $\left\{\overline{Z}\right\}_1 > \beta_1 \underline{v_l}^{-1}\left(\overline{v_l}\left(\|Z_0\|\right)\right)$, let $\mathcal{Z}_2 \triangleq \left\{ \varrho \in \mathbb{R}^{n+2\{L\}_1} \mid \|\varrho\| \leq \beta_2 \left\{\overline{Z}\right\}_1 \right\}$. Using $\mathcal{Z}_2$, compute the bounds in (5–20) and (5–27) and select the gains according to (5–28).

If $\left\{\overline{Z}\right\}_2 \leq \left\{\overline{Z}\right\}_1$, set $\mathcal{Z} = \mathcal{Z}_2$ and terminate.

Third iteration:

If $\left\{\overline{Z}\right\}_2 > \left\{\overline{Z}\right\}_1$, increase the number of NN neurons to $\{L\}_3$ to yield a lower function approximation error $\left\{\overline{\nabla\epsilon}\right\}_3$ such that $\{L_F\}_2 \left\{\overline{\nabla\epsilon}\right\}_3 \leq \{L_F\}_1 \left\{\overline{\nabla\epsilon}\right\}_1$. The increase in the number of NN neurons ensures that $\{\iota\}_3 \leq \{\iota\}_1$. Furthermore, the assumption that the PE interval $\{T\}_3$ is small enough such that $\{L_F\}_2 \{T\}_3 \leq \{T\}_1 \{L_F\}_1$ and $\{L\}_3 \{T\}_3 \leq \{T\}_1 \{L\}_1$ ensures that $\left\{\frac{\varpi_{10}}{\varpi_{11}}\right\}_3 \leq \left\{\frac{\varpi_{10}}{\varpi_{11}}\right\}_1$, and hence, $\left\{\overline{Z}\right\}_3 \leq \beta_2 \left\{\overline{Z}\right\}_1$. Set $\mathcal{Z} = \left\{ \varrho \in \mathbb{R}^{n+2\{L\}_3} \mid \|\varrho\| \leq \beta_2 \left\{\overline{Z}\right\}_1 \right\}$ and terminate.

5.3.3 Main Result

Theorem 5.1. *Provided that the sufficient conditions in (5–28) and (5–29) are satisfied and Assumptions 5.1 - 5.3 hold, the controller in (5–13) and the update laws in (5–15) - (5–16) guarantee that the tracking error is ultimately bounded, and the error $\|\mu(t) - \mu^*(\zeta(t))\|$ is ultimately bounded as $t \to \infty$.*

Proof. The time derivative of V_L is $\dot{V}_L = \nabla V^* F + \nabla V^* G \hat{\mu} + \tilde{W}_c^T \Gamma^{-1} \dot{\tilde{W}}_c - \frac{1}{2}\tilde{W}_c^T \Gamma^{-1} \dot{\Gamma} \Gamma^{-1} \tilde{W}_c - \tilde{W}_a^T \dot{\hat{W}}_a$. Using (5–19) and the facts that $\nabla V^* F = -\nabla V^* G \mu^* - r(\zeta, \mu^*)$ and $\nabla V^* G = -2\mu^{*T} R$ yields

$$\dot{V}_L = -e^T Q e + \mu^{*T} R \mu^* - 2\mu^{*T} R \hat{\mu} - \eta_c \tilde{W}_c^T \psi \psi^T \tilde{W}_c - \lambda \frac{\eta_C}{2} \tilde{W}_c^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2} \eta_c \tilde{W}_c^T \frac{\omega \omega^T}{\rho} \tilde{W}_c$$
$$- \tilde{W}_a^T \dot{\hat{W}}_a + \frac{\eta_c \tilde{W}_c^T \psi}{\sqrt{1 + \nu \omega^T \Gamma \omega}} \left(\frac{1}{4} \tilde{W}_a^T \mathcal{G}_\sigma \tilde{W}_a - \nabla\epsilon F + \frac{1}{4} \nabla\epsilon \mathcal{G} \nabla\epsilon^T + \frac{1}{2} W^T \nabla\sigma \mathcal{G} \nabla\epsilon^T \right), \quad (5–30)$$

where $\rho \triangleq 1 + \nu \omega^T \Gamma \omega$. Using (5–15), (5–19) and the bounds in (5–18) - (5–20) the Lyapunov derivative in (5–30) can be bounded above on the set $\mathcal{Z}$ as

$$\dot{V}_L \leq -\frac{q}{2} \|e\|^2 - \frac{1}{4} \eta_c \left\| \tilde{W}_c^T \psi \right\|^2 - \frac{\eta_{a12}}{2} \left\| \tilde{W}_a \right\|^2 + \left(2\eta_{a2} \overline{W} + \iota_4 \right) \left\| \tilde{W}_a \right\| - \frac{\eta_c}{2} \left(1 - \frac{\overline{\nabla\epsilon}}{\xi_1} \right) \left\| \tilde{W}_c^T \psi \right\|^2$$

$$-\frac{1}{2}\left(\eta_{a12}-\eta_{a1}\xi_{2}-\frac{\eta_{c}\iota_{2}}{4}\left\Vert \tilde{W}_{c}^{T}\psi\right\Vert \right)\left\Vert \tilde{W}_{a}\right\Vert ^{2}-\frac{\left(\underline{q}-\eta_{c}L_{F}\overline{\nabla\epsilon}\xi_{1}\right)}{2}\left\Vert e\right\Vert ^{2}-\frac{1}{2}\left(\lambda\eta_{c}\underline{\gamma}-\frac{\eta_{a1}}{\xi_{2}}\right)\left\Vert \tilde{W}_{c}\right\Vert ^{2}$$
$$+\eta_{c}\left(\iota_{1}+\iota_{2}\overline{W}^{2}\right)\left\Vert \tilde{W}_{c}^{T}\psi\right\Vert +\frac{1}{4}\iota_{3},$$

where $\xi_1, \xi_2 \in \mathbb{R}$ are known adjustable positive constants. Provided the sufficient conditions in (5–28) are satisfied, completion of squares yields

$$\dot{V}_{L}\leq-\frac{\underline{q}}{2}\left\Vert e\right\Vert ^{2}-\frac{1}{8}\eta_{c}\left\Vert \tilde{W}_{c}^{T}\psi\right\Vert ^{2}-\frac{\eta_{a12}}{4}\left\Vert \tilde{W}_{a}\right\Vert ^{2}+\iota. \tag{5–31}$$

The inequality in (5–31) is valid provided $Z(t) \in \mathcal{Z}$. Integrating (5–31) and using Lemma 5.3 and the gain conditions in (5–28) yields

$$V_{L}\left(Z\left(t+T\right),t+T\right)-V_{L}\left(Z\left(t\right),t\right)\leq-\frac{1}{8}\eta_{c}\underline{\psi}\varpi_{7}\left\Vert \tilde{W}_{c}\left(t\right)\right\Vert ^{2}-\frac{\underline{q}}{4}\int\limits_{t}^{t+T}\left\Vert e\left(\tau\right)\right\Vert ^{2}d\tau+\frac{1}{8}\eta_{c}\varpi_{9}$$
$$-\frac{\eta_{a12}}{8}\int\limits_{t}^{t+T}\left\Vert \tilde{W}_{a}\left(\tau\right)\right\Vert ^{2}d\tau+\iota T,$$

provided $Z(\tau) \in \mathcal{Z}, \ \forall\tau \in [t, t+T]$. Using the facts that $-\int_{t}^{t+T}\left\Vert e\left(\tau\right)\right\Vert ^{2}d\tau \leq -T\inf_{\tau\in[t,t+T]}\left\Vert e\left(\tau\right)\right\Vert ^{2}$ and $-\int_{t}^{t+T}\left\Vert \tilde{W}_{a}\left(\tau\right)\right\Vert ^{2}d\tau \leq -T\inf_{\tau\in[t,t+T]}\left\Vert \tilde{W}_{a}\left(\tau\right)\right\Vert ^{2}$, and Lemma 5.2 yield

$$V_{L}\left(Z\left(t+T\right),t+T\right)-V_{L}\left(Z\left(t\right),t\right)\leq-\frac{\eta_{c}\underline{\psi}\varpi_{7}}{16}\left\Vert \tilde{W}_{c}\left(t\right)\right\Vert ^{2}-\frac{\varpi_{3}\eta_{a12}T}{16}\left\Vert \tilde{W}_{a}\left(t\right)\right\Vert ^{2}+\varpi_{10}T$$
$$-\frac{\varpi_{0}\underline{q}T}{8}\left\Vert e\left(t\right)\right\Vert ^{2},$$

provided $Z(\tau) \in \mathcal{Z}, \forall\tau \in [t, t+T]$. Thus, $V_L(Z(t+T), t+T) - V_L(Z(t), t) < 0$ provided $\left\Vert Z\left(t\right)\right\Vert > \sqrt{\frac{\varpi_{10}T}{\varpi_{11}}}$ and $Z(\tau) \in \mathcal{Z}, \forall\tau \in [t, t+T]$. The bounds on the Lyapunov function in (5–26) yield $V_L(Z(t+T), t+T) - V_L(Z(t), t) < 0$ provided $V_L(Z(t), t) > \overline{v_l}\left(\sqrt{\frac{\varpi_{10}T}{\varpi_{11}}}\right)$ and $Z(\tau) \in \mathcal{Z}, \ \forall\tau \in [t, t+T]$.

Since $Z(t_0) \in \mathcal{Z}$, (5–31) can be used to conclude that $\dot{V}_L(Z(t_0), t_0) \leq \iota$. The sufficient condition in (5–29) ensures that $\underline{v_l}^{-1}\left(V_L(Z(t_0), t_0) + \iota T\right) \leq r$; hence, $Z(t) \in \mathcal{Z}$ for all $t \in [t_0, t_0 + T]$. If $V_L(Z(t_0), t_0) > \overline{v_l}\left(\sqrt{\frac{\varpi_{10}T}{\varpi_{11}}}\right)$, then $Z(t) \in \mathcal{Z}$

for all $t \in [t_0, t_0 + T]$ implies $V_L(Z(t_0+T), t_0+T) - V_L(Z(t_0), t_0) < 0$; hence, $\underline{v_l}^{-1}(V_L(Z(t_0+T), t_0+T) + \iota T) \leq r$. Thus, $Z(t) \in \mathcal{Z}$ for all $t \in [t_0 + T, t_0 + 2T]$. Inductively, the system state is bounded such that $\sup_{t\in[0,\infty)} \|Z(t)\| \leq r$ and ultimately bounded[8] such that

$$\limsup_{t\to\infty} \|Z(t)\| \leq \underline{v_l}^{-1}\left(\overline{v_l}\left(\sqrt{\frac{\varpi_{10}T}{\varpi_{11}}}\right) + \iota T\right).$$

□

5.4 Simulation

Simulations are performed on a two-link manipulator to demonstrate the ability of the presented technique to approximately optimally track a desired trajectory. The two link robot manipulator is modeled using Euler-Lagrange dynamics as

$$M\ddot{q} + V_m\dot{q} + F_d\dot{q} + F_s = u, \tag{5–32}$$

where $q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$ and $\dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 \end{bmatrix}^T$ are the angular positions in radians and the angular velocities in radian/s respectively. In (5–32), $M \in \mathbb{R}^{2\times 2}$ denotes the inertia matrix, and $V_m \in \mathbb{R}^{2\times 2}$ denotes the centripetal-Coriolis matrix given by $M \triangleq \begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix}$, $V_m \triangleq \begin{bmatrix} -p_3s_2\dot{q}_2 & -p_3s_2(\dot{q}_1 + \dot{q}_2) \\ p_3s_2\dot{q}_1 & 0 \end{bmatrix}$, where $c_2 = cos(q_2)$, $s_2 = sin(q_2)$, $p_1 = 3.473\ kg.m^2$, $p_2 = 0.196\ kg.m^2$, and $p_3 = 0.242\ kg.m^2$, and $F_d =$ diag$\begin{bmatrix} 5.3, & 1.1 \end{bmatrix}$ $Nm.s$ and $F_s(\dot{q}) = \begin{bmatrix} 8.45tanh(\dot{q}_1), & 2.35tanh(\dot{q}_2) \end{bmatrix}^T$ Nm are the models for the static and the dynamic friction, respectively.

The objective is to find a policy $\hat{\mu}$ that ensures that the state $x \triangleq \begin{bmatrix} q_1, & q_2, & \dot{q}_1, & \dot{q}_2 \end{bmatrix}^T$ tracks the desired trajectory $x_d(t) = \begin{bmatrix} 0.5cos(2t), & 0.33cos(3t), & -sin(2t), & -sin(3t) \end{bmatrix}^T$,

[8] If the regressor ψ satisfies a stronger u-PE assumption (cf. [155, 156]), the tracking error and the weight estimation errors can be shown to be uniformly ultimately bounded.

while minimizing the cost $\int_0^\infty \left(e^T Q e + \hat{\mu}^T \hat{\mu}\right) dt$, where $Q = \text{diag}\begin{bmatrix} 10, & 10, & 2, & 2 \end{bmatrix}$. Using (5–2) - (5–5) and the definitions

$$f \triangleq \begin{bmatrix} x_3, & x_4, & \left(M^{-1}\left(-V_m - F_d\right)\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} - F_s\right)^T \end{bmatrix}^T, \; h_d \triangleq \begin{bmatrix} x_{d3}, & x_{d4}, & -4x_{d1}, & -9x_{d2} \end{bmatrix}^T,$$

$$g_d^+ \triangleq \begin{bmatrix} \begin{bmatrix} 0, & 0 \end{bmatrix}^T, & \begin{bmatrix} 0, & 0 \end{bmatrix}^T, & M\left(x_d\right) \end{bmatrix}, \quad g \triangleq \begin{bmatrix} \begin{bmatrix} 0, & 0 \end{bmatrix}^T, & \begin{bmatrix} 0, & 0 \end{bmatrix}^T, & \left(M^{-1}\right)^T \end{bmatrix}^T, \tag{5–33}$$

the optimal tracking problem can be transformed into the time-invariant form in (5–5).

The two major challenges in the application of ADP to systems such as (5–33) include selecting an appropriate basis for the value function approximation and ensuring that the regressor ψ introduced in Assumption 5.3 is PE. Due to the size of the state space and the complexity of the dynamics, obtaining an analytical solution to the HJB equation for this problem is prohibitively difficult. Furthermore, since the regressor is a complex nonlinear function of the states, it is difficult to ensure that it remains PE. As a result, this serves as a model problem to demonstrate the applicability of ADP-based approximate online optimal control.

In this effort, the basis selected for the value function approximation is a polynomial basis with 23 elements given by

$$\sigma(\zeta) = \frac{1}{2}\begin{bmatrix} \zeta_1^2 & \zeta_2^2 & \zeta_1\zeta_3 & \zeta_1\zeta_4 & \zeta_2\zeta_3 & \zeta_2\zeta_4 & \zeta_1^2\zeta_2^2 & \zeta_1^2\zeta_5^2 & \zeta_1^2\zeta_6^2 & \zeta_1^2\zeta_7^2 & \zeta_1^2\zeta_8^2 & \zeta_2^2\zeta_5^2 \\ \zeta_2^2\zeta_6^2 & \zeta_2^2\zeta_7^2 & \zeta_2^2\zeta_8^2 & \zeta_3^2\zeta_5^2 & \zeta_3^2\zeta_6^2 & \zeta_3^2\zeta_7^2 & \zeta_3^2\zeta_8^2 & \zeta_4^2\zeta_5^2 & \zeta_4^2\zeta_6^2 & \zeta_4^2\zeta_7^2 & \zeta_4^2\zeta_8^2 \end{bmatrix}^T. \tag{5–34}$$

The control gains are selected as $\eta_{a1} = 5$, $\eta_{a2} = 0.001$, $\eta_c = 1.25$, $\lambda = 0.001$, and $\nu = 0.005$, and the initial conditions are $x\left(0\right) = \begin{bmatrix} 1.8 & 1.6 & 0 & 0 \end{bmatrix}^T$, $\hat{W}_c\left(0\right) = 10 \times \mathbf{1}_{23\times1}$,

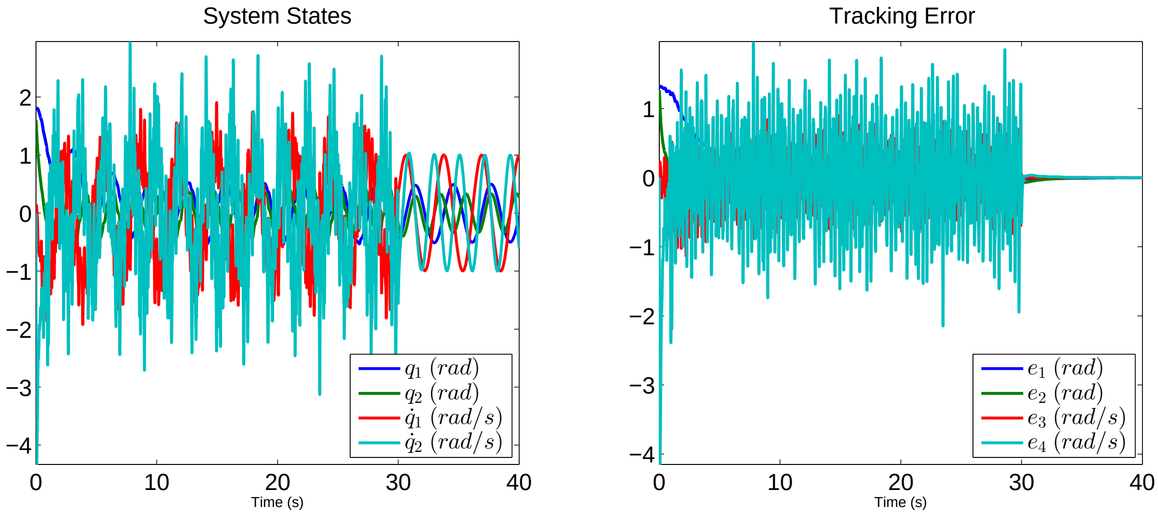

Figure 5-1. State and error trajectories with probing signal.

$\hat{W}_a(0) = 6 \times \mathbf{1}_{23\times 1}$, and $\Gamma(0) = 2000 I_{23}$. To ensure PE, a probing signal

$$p(t) = \begin{bmatrix} 2.55tanh(2t)\left(20sin\left(\sqrt{232}\pi t\right)cos\left(\sqrt{20}\pi t\right)\right. \\ \left.+6sin\left(18e^2 t\right) + 20cos\left(40t\right)cos\left(21t\right)\right) \\ 0.01tanh(2t)\left(20sin\left(\sqrt{132}\pi t\right)cos\left(\sqrt{10}\pi t\right)\right. \\ \left.+6sin\left(8et\right) + 20cos\left(10t\right)cos\left(11t\right)\right) \end{bmatrix} \tag{5–35}$$

is added to the control signal for the first 30 seconds of the simulation [57].

It is clear from Figure 5-1 that the system states are bounded during the learning phase and the algorithm converges to a stabilizing controller in the sense that the tracking errors go to zero when the probing signal is eliminated. Furthermore, Figure 5-2 shows that the weight estimates for the value function and the policy are bounded and they converge. Thus, Figures 5-1 and 5-2 demonstrate that an approximate optimal policy can be generated online to solve an optimal tracking problem using a simple polynomial basis such as (5–34), and a probing signal that consists of a combination of sinusoidal signals such as (5–35).

The NN weights converge to the following values

$$\hat{W}_c = \hat{W}_a = \left[\begin{array}{ccccccccccc} 83.36 & 2.37 & 27.0 & 2.78 & -2.83 & 0.20 & 14.13 & 29.81 & 18.87 & 4.11 & 3.47 \end{array}\right.$$

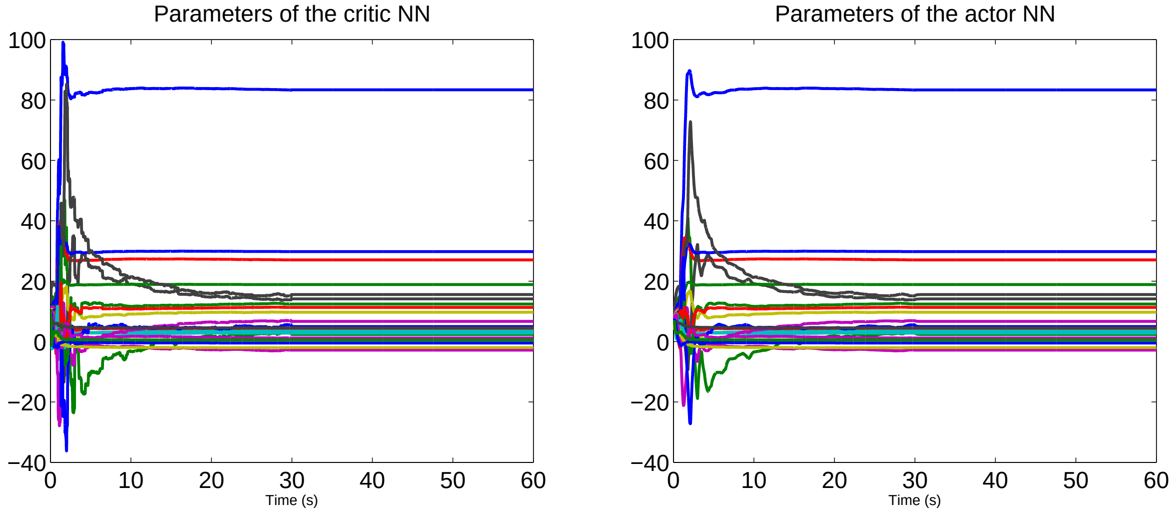

Figure 5-2. Evolution of value function and policy weights.

$$6.69 \quad 9.71 \quad 15.58 \quad 4.97 \quad 12.42 \quad 11.31 \quad 3.29 \quad 1.19 \quad -1.99 \quad 4.55 \quad -0.47 \quad 0.56 \Bigg]^T.$$

Note that the last sixteen weights that correspond to the terms containing the desired trajectories $\zeta_5, \cdots, \zeta_8$ are non-zero. Thus, the resulting value function V and the resulting policy $\hat{\mu}$ depend on the desired trajectory, and hence, are time-varying functions of the tracking error. Since the true weights are unknown, a direct comparison of the weights in (5.4) with the true weights is not possible. Instead, to gauge the performance of the presented technique, the state and the control trajectories obtained using the estimated policy are compared with those obtained using Radau-pseudospectral numerical optimal control computed using the GPOPS software [7]. Since an accurate numerical solution is difficult to obtain for an infinite-horizon optimal control problem, the numerical optimal control problem is solved over a finite horizon ranging over approximately 5 times the settling time associated with the slowest state variable. Based on the solution obtained using the proposed technique, the slowest settling time is estimated to be approximately 20 seconds. Thus, to approximate the infinite-horizon solution, the numerical solution is computed over a 100 second time horizon using 300 collocation points.

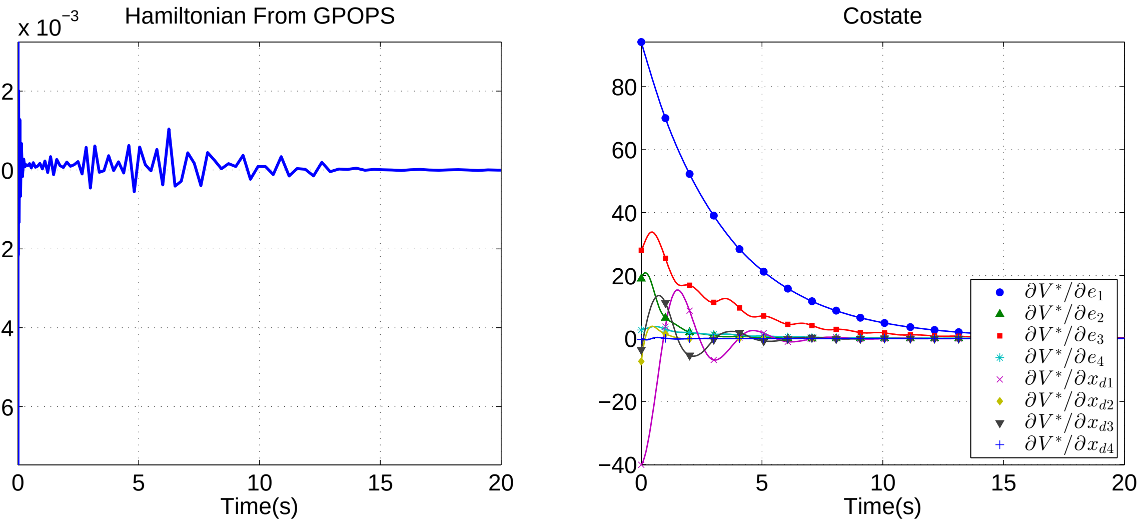

Figure 5-3. Hamiltonian and costate of the numerical solution computed using GPOPS.

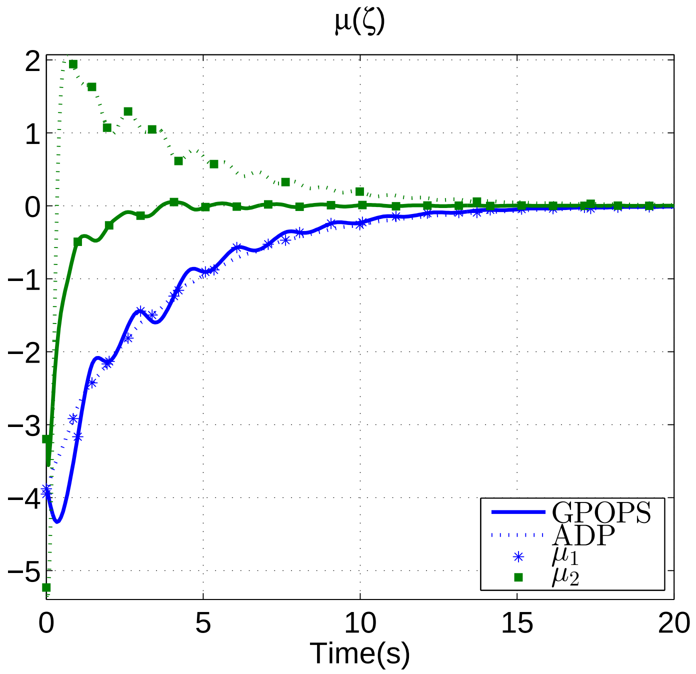

Figure 5-4. Control trajectories $\hat{\mu}(t)$ obtained from GPOPS and the developed technique.

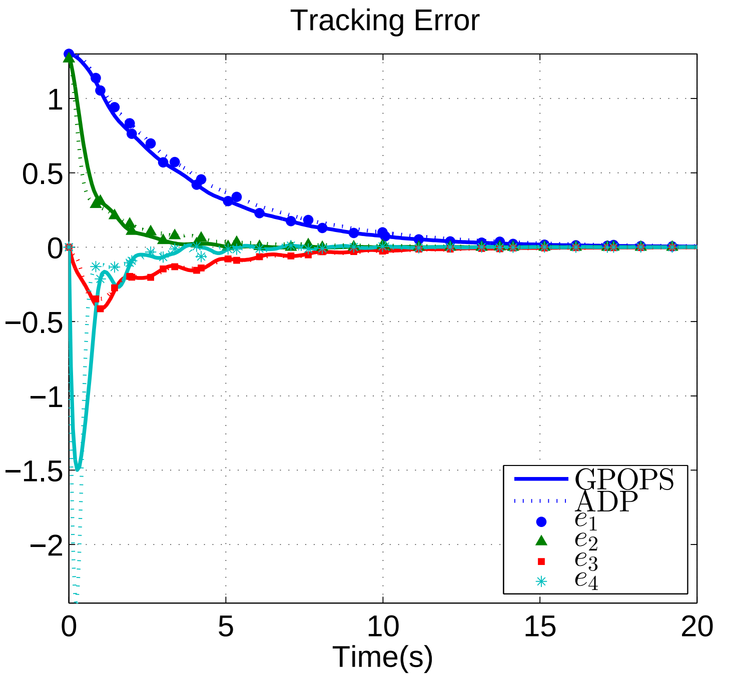

Figure 5-5. Tracking error trajectories $e(t)$ obtained from GPOPS and the developed technique.

As seen in Figure 5-3, the Hamiltonian of the numerical solution is approximately zero. This supports the assertion that the optimal control problem is time-invariant. Furthermore, since the Hamiltonian is close to zero, the numerical solution obtained using GPOPS is sufficiently accurate as a benchmark to compare against the ADP-based solution obtained using the proposed technique. Note that in Figure 5-3, the costate variables corresponding to the desired trajectories are nonzero. Since these costate variables represent the sensitivity of the cost with respect to the desired trajectories, this further supports the assertion that the optimal value function depends on the desired trajectory, and hence, is a time-varying function of the tracking error.

Figures 5-4 and 5-5 show the control and the tracking error trajectories obtained from the developed technique (dashed lines) plotted alongside the numerical solution obtained using GPOPS (solid lines). The trajectories obtained using the developed technique are close to the numerical solution. The inaccuracies are a result of the facts that the set of basis functions in (5–34) is not exact, and the proposed method attempts to find the weights that generate the least total cost for the given set of basis functions. The accuracy of the approximation can be improved by choosing a more appropriate set of basis functions, or at an increased computational cost, by adding more basis functions to the existing set in (5–34). The total cost $\int_0^{100}\left(e\left(t\right)^T Qe\left(t\right)+\mu\left(t\right)^T R\mu\left(t\right)\right)dt$ obtained using the numerical solution is found to be 75.42 and the total cost $\int_0^{\infty}\left(e\left(t\right)^T Qe\left(t\right)+\mu\left(t\right)^T R\mu\left(t\right)\right)dt$ obtained using the developed method is found to be 84.31. Note that from Figures 5-4 and 5-5, it is clear that both the tracking error and the control converge to zero after approximately 20 seconds, and hence, the total cost obtained from the numerical solution is a good approximation of the infinite-horizon cost.

5.5 Concluding Remarks

An ADP-based approach using the policy evaluation and policy improvement architecture is presented to approximately solve the infinite-horizon optimal tracking problem for control-affine nonlinear systems with quadratic cost. The problem is solved by transforming the system to convert the tracking problem that has a time-varying value function, into a time-invariant optimal control problem. The ultimately bounded tracking and estimation result was established using Lyapunov analysis for nonautonomous systems. Simulations are performed to demonstrate the applicability and the effectiveness of the developed method. The developed method can be applied to high-dimensional nonlinear dynamical systems using simple polynomial basis functions and sinusoidal probing signals. However, the accuracy of the approximation depends on the choice of basis functions and the result hinges on the system states being PE. Furthermore, computation of the desired control in (5–2) requires exact model knowledge. The following chapter uses model-based RL ideas from Chapter 3 to relax the PE requirement and to allow for uncertainties in the system dynamics.

CHAPTER 6
MODEL-BASED REINFORCEMENT LEARNING FOR APPROXIMATE OPTIMAL TRACKING

In this chapter, the tracking controller developed in Chapter 5 is extended to solve infinite-horizon optimal tracking problems control-affine continuous-time nonlinear systems with uncertain drift dynamics using model-based RL. In Chapter 5, model knowledge is used in the computation of the BE and in the computation of the steady-state control signal. In this chapter, a CL-based system identifier is used to simulate experience by evaluating the BE over unexplored areas of the state space. The system identifier is also utilized to approximate the steady-state control signal. A Lyapunov-based stability analysis is presented to establish simultaneous identification and trajectory tracking. Effectiveness of the developed technique is demonstrated via numerical simulations.

6.1 Problem Formulation and Exact Solution

Consider the concatenated nonlinear control-affine system described by the differential equation (5–4). Similar to Chapter 5, the objective of the optimal control problem is to minimize the cost functional $J(\zeta, \mu)$, introduced in (2–2), subject to the dynamic constraints in (5–4) while tracking the desired trajectory. In this chapter, a more general form of the reward signal is considered. The reward signal $r : \mathbb{R}^{2n} \times \mathbb{R}^{m} \to \mathbb{R}$ is given by

$$r(\zeta, \mu) \triangleq \overline{Q}(\zeta) + \mu^T R \mu,$$

where the function $\overline{Q} : \mathbb{R}^{2n} \to \mathbb{R}$ is defined as

$$\overline{Q}\left(\begin{bmatrix} e \\ x_d \end{bmatrix}\right) \triangleq Q(e), \ \forall x_d \in \mathbb{R}^n, \tag{6–1}$$

where $Q : \mathbb{R}^n \to \mathbb{R}$ is a continuous positive definite function that satisfies

$$\underline{q}(\|e^o\|) \leq Q(e^o) \leq \overline{q}(\|e^o\|), \ \forall e^o \in \mathbb{R}^n$$

where $\underline{q} : \mathbb{R} \to \mathbb{R}$ and $\overline{q} : \mathbb{R} \to \mathbb{R}$ are class $\mathcal{K}$ functions.

Using the estimates $\hat{V}\left(\zeta, \hat{W}_c\right)$ and $\hat{\mu}\left(\zeta, \hat{W}_a\right)$ in (5–9) the BE can be obtained as

$$\delta\left(\zeta, \hat{W}_c, \hat{W}_a\right) \triangleq \nabla_\zeta \hat{V}\left(\zeta, \hat{W}_c\right)\left(F\left(\zeta\right) + G\left(\zeta\right)\hat{\mu}\left(\zeta, \hat{W}_a\right)\right) + r\left(\zeta, \hat{\mu}\left(\zeta, \hat{W}_a\right)\right). \tag{6–2}$$

In this chapter, simulation of experience via BE extrapolation is used to improve data efficiency, based on the observation that if a dynamic system identifier is developed to generate an estimate $F_\theta\left(\zeta, \hat{\theta}\right)$ of the drift dynamics F, an estimate of the BE in (6–2) can be evaluated at any $\zeta \in \mathbb{R}^{2n}$. That is, using $\hat{F}$, experience can be simulated by extrapolating the BE over unexplored off-trajectory points in the operating domain. Hence, if an identifier can be developed such that $\hat{F}$ approaches F exponentially fast, learning laws for the optimal policy can utilize simulated experience along with experience gained and stored along the state trajectory.

If parametric approximators are used to approximate F, convergence of $\hat{F}$ to F is implied by convergence of the parameters to their unknown ideal values. It is well known that adaptive system identifiers require PE to achieve parameter convergence. To relax the PE condition, a CL-based (cf. [92, 93, 97, 147]) system identifier that uses recorded data for learning is developed in the following section.

6.2 System Identification

On any compact set $\mathcal{C} \subset \mathbb{R}^n$ the function f can be represented using a NN as

$$f\left(x^o\right) = \theta^T \sigma_f\left(Y^T x_1\right) + \epsilon_\theta\left(x^o\right), \; \forall x^o \in \mathbb{R}^n \tag{6–3}$$

where $x_1 \triangleq \begin{bmatrix} 1 & \left(x^o\right)^T \end{bmatrix}^T \in \mathbb{R}^{n+1}$, $\theta \in \mathbb{R}^{p+1\times n}$ and $Y \in \mathbb{R}^{n+1\times p}$ denote the unknown output-layer and hidden-layer NN weights, $\sigma_f : \mathbb{R}^p \to \mathbb{R}^{p+1}$ denotes a bounded NN basis function, $\epsilon_\theta : \mathbb{R}^n \to \mathbb{R}^n$ denotes the function reconstruction error, and $p \in \mathbb{N}$ denotes the number of NN neurons. Using the universal function approximation property of single layer NNs, given a constant matrix Y such that the rows of $\sigma_f\left(Y^T x_1\right)$ form a proper basis, there exist constant ideal weights θ and known constants $\overline{\theta}$, $\overline{\epsilon_\theta}$, and $\overline{\epsilon_\theta'} \in \mathbb{R}$ such that $\|\theta\|_F \leq \overline{\theta} < \infty$, $\sup_{x^o \in \mathcal{C}} \|\epsilon_\theta (x^o)\| \leq \overline{\epsilon_\theta}$, and $\sup_{x^o \in \mathcal{C}} \|\nabla_{x^o} \epsilon_\theta (x^o)\| \leq \overline{\epsilon_\theta'}$, where $\|\cdot\|_F$ denotes the Frobenius norm.

Using an estimate $\hat{\theta} \in \mathbb{R}^{p+1\times n}$ of the weight matrix θ, the function f can be approximated by the function $\hat{f} : \mathbb{R}^{2n} \times \mathbb{R}^{p+1\times n} \to \mathbb{R}^n$ defined as

$$\hat{f}\left(\zeta, \hat{\theta}\right) \triangleq \hat{\theta}^T \sigma_\theta (\zeta), \tag{6–4}$$

where $\sigma_\theta : \mathbb{R}^{2n} \to \mathbb{R}^{p+1}$ is defined as $\sigma_\theta(\zeta) = \sigma_f\left(Y^T \begin{bmatrix} 1 & e^T + x_d^T \end{bmatrix}^T\right)$. Based on (6–3), an estimator for online identification of the drift dynamics is developed as

$$\dot{\hat{x}} = \hat{\theta}^T \sigma_\theta(\zeta) + g(x)u + k\tilde{x}, \tag{6–5}$$

where $\tilde{x} \triangleq x - \hat{x}$, and $k \in \mathbb{R}$ is a positive constant learning gain. The following assumption facilitates CL-based system identification.

Assumption 6.1. [92] A history stack containing recorded state-action pairs $\{x_j, u_j\}_{j=1}^M$ along with numerically computed state derivatives $\{\dot{\bar{x}}_j\}_{j=1}^M$ that satisfies

$$\lambda_{\min}\left(\sum_{j=1}^M \sigma_{fj}\sigma_{fj}^T\right) = \underline{\sigma_\theta} > 0,$$
$$\|\dot{\bar{x}}_j - \dot{x}_j\| < \overline{d},\ \forall j \tag{6–6}$$

is available a priori. In (6–6), $\sigma_{fj} \triangleq \sigma_f\left(Y^T \begin{bmatrix} 1 & x_j^T \end{bmatrix}^T\right)$, $\overline{d} \in \mathbb{R}$ is a known positive constant, and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue.

The weight estimates $\hat{\theta}$ are updated using the following CL-based update law:

$$\dot{\hat{\theta}} = \Gamma_\theta \sigma_f\left(Y^T x_1\right)\tilde{x}^T + k_\theta \Gamma_\theta \sum_{j=1}^M \sigma_{fj}\left(\dot{\bar{x}}_j - g_j u_j - \hat{\theta}^T \sigma_{fj}\right)^T, \tag{6–7}$$

where $k_\theta \in \mathbb{R}$ is a constant positive CL gain, and $\Gamma_\theta \in \mathbb{R}^{p+1\times p+1}$ is a constant, diagonal, and positive definite adaptation gain matrix.

To facilitate the subsequent stability analysis, a candidate Lyapunov function $V_0 : \mathbb{R}^n \times \mathbb{R}^{p+1\times n} \to \mathbb{R}$ is selected as

$$V_0\left(\tilde{x}, \tilde{\theta}\right) \triangleq \frac{1}{2}\tilde{x}^T\tilde{x} + \frac{1}{2}\text{tr}\left(\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}\right), \tag{6–8}$$

where $\tilde{\theta} \triangleq \theta - \hat{\theta}$ and tr $(\cdot)$ denotes the trace of a matrix. Using (6–5)-(6–7), the following bound on the time derivative of V_0 is established:

$$\dot{V}_0 \leq -k\left\|\tilde{x}\right\|^2 - k_\theta\underline{\sigma_\theta}\left\|\tilde{\theta}\right\|_F^2 + \overline{\epsilon_\theta}\left\|\tilde{x}\right\| + k_\theta\overline{d_\theta}\left\|\tilde{\theta}\right\|_F, \tag{6–9}$$

where $\overline{d_\theta} = \overline{d}\sum_{j=1}^M \overline{\|\sigma_{\theta j}\|} + \sum_{j=1}^M \left(\overline{\|\epsilon_{\theta j}\|\|\sigma_{\theta j}\|}\right)$. Using (6–8) and (6–9) a Lyapunov-based stability analysis can be used to show that $\hat{\theta}$ converges exponentially to a neighborhood around θ.

Using (6–4), the BE in (6–2) can be approximated as

$$\hat{\delta}\left(\zeta, \hat{W}_c, \hat{W}_a, \hat{\theta}\right) \triangleq \nabla_\zeta\hat{V}\left(\zeta, \hat{W}_c\right)\left(F_\theta\left(\zeta, \hat{\theta}\right) + F_1(\zeta) + G(\zeta)\hat{\mu}\left(\zeta, \hat{W}_a\right)\right) + \overline{Q}(\zeta) + \hat{\mu}^T\left(\zeta, \hat{W}_a\right)R\hat{\mu}\left(\zeta, \hat{W}_a\right),$$

where

$$F_\theta\left(\zeta, \hat{\theta}\right) \triangleq \begin{bmatrix} \hat{\theta}^T\sigma_\theta(\zeta) - g(x)\,g^+(x_d)\,\hat{\theta}^T\sigma_\theta\left(\begin{bmatrix}\mathbf{0}_{n\times 1} \\ x_d\end{bmatrix}\right) \\ \mathbf{0}_{n\times 1}\end{bmatrix},$$

and $F_1(\zeta) \triangleq \begin{bmatrix} \left(-h_d + g(e + x_d)\,g^+(x_d)\,h_d\right)^T & h_d^T \end{bmatrix}^T$. The optimal tracking problem is thus reformulated as the need to find estimates $\hat{\mu}$ and $\hat{V}$ online, to minimize the total integral error

$$\hat{E}_{\hat{\theta}}\left(\hat{W}_c, \hat{W}_a\right) \triangleq \int\limits_{\zeta\in\mathbb{R}^{2n}} \left(\delta\left(\zeta, \hat{W}_c, \hat{W}_a, \hat{\theta}\right)\right)^2 d\zeta,$$

for a given $\hat{\theta}$, while simultaneously improving $\hat{\theta}$ using (6–7), and ensuring stability of the system in (2–1) using the control law $u = \hat{\mu} + \hat{u}_d$, where

$$\hat{u}_d\left(\zeta, \hat{\theta}\right) \triangleq g_d^+\left(h_d - \hat{\theta}^T \sigma_{\theta d}\right), \tag{6–10}$$

and $\sigma_{\theta d} \triangleq \sigma_\theta\left(\begin{bmatrix} \mathbf{0}_{1\times n} & x_d^T \end{bmatrix}^T\right)$.

6.3 Value Function Approximation

Since V^* and μ^* are functions of the state ζ, the minimization problem stated in Section 6.2 is infinite-dimensional, and hence, intractable. To obtain a finite-dimensional minimization problem, the optimal value function is represented over any compact operating domain $\mathcal{C} \subset \mathbb{R}^{2n}$ using a NN as

$$V^*\left(\zeta^o\right) = W^T \sigma\left(\zeta^o\right) + \epsilon\left(\zeta^o\right), \; \forall \zeta^o \in \mathbb{R}^{2n}$$

where $W \in \mathbb{R}^L$ denotes a vector of unknown NN weights, $\sigma : \mathbb{R}^{2n} \to \mathbb{R}^L$ denotes a bounded NN basis function, $\epsilon : \mathbb{R}^{2n} \to \mathbb{R}$ denotes the function reconstruction error, and $L \in \mathbb{N}$ denotes the number of NN neurons. Using the universal function approximation property of single layer NNs, there exist constant ideal weights W and known constants $\overline{W}$, $\overline{\epsilon}$, and $\overline{\nabla \epsilon} \in \mathbb{R}$ such that $\|W\| \leq \overline{W} < \infty$, $\sup_{\zeta^o \in \mathcal{C}} \|\epsilon\left(\zeta^o\right)\| \leq \overline{\epsilon}$, and $\sup_{\zeta^o \in \mathcal{C}} \|\nabla \epsilon\left(\zeta^o\right)\| \leq \overline{\nabla \epsilon}$.

Using (5–8), a NN representation of the optimal policy is obtained as

$$\mu^*\left(\zeta^o\right) = -\frac{1}{2} R^{-1} G^T\left(\zeta^o\right)\left(\nabla \sigma^T\left(\zeta^o\right) W + \nabla \epsilon^T\left(\zeta^o\right)\right), \; \forall \zeta^o \in \mathbb{R}^{2n}. \tag{6–11}$$

Using estimates $\hat{W}_c$ and $\hat{W}_a$ for the ideal weights W, the optimal value function and the optimal policy are approximated as

$$\hat{V}\left(\zeta, \hat{W}_c\right) \triangleq \hat{W}_c^T \sigma\left(\zeta\right), \quad \hat{\mu}\left(\zeta, \hat{W}_a\right) \triangleq -\frac{1}{2} R^{-1} G^T\left(\zeta\right) \nabla \sigma^T\left(\zeta\right) \hat{W}_a. \tag{6–12}$$

The optimal control problem is thus reformulated as the need to find a set of weights $\hat{W}_c$ and $\hat{W}_a$ online, to minimize the instantaneous squared error in 2–11, where the instantaneous BE is given by

$$\hat{E}_{\hat{\theta}}\left(\hat{W}_c, \hat{W}_a\right) \triangleq \int\limits_{\zeta\in C} \left(\hat{\delta}\left(\zeta, \hat{W}_c, \hat{W}_a, \hat{\theta}\right)\right)^2 d\zeta, \tag{6–13}$$

for a given $\hat{\theta}$, while simultaneously improving $\hat{\theta}$ using (6–7), and ensuring stability of the system in (2–1) using the control law

$$u = \hat{\mu}\left(\zeta, \hat{W}_a\right) + \hat{u}_d\left(\zeta, \hat{\theta}\right). \tag{6–14}$$

Using (5–2), (6–10), and (6–14), the virtual controller μ for the concatenated system in (5–4) can be expressed as[1]

$$\mu = \hat{\mu}\left(\zeta, \hat{W}_a\right) + g_d^+ \tilde{\theta}^T \sigma_{\theta d} + g_d^+ \epsilon_{\theta d}, \tag{6–15}$$

where $\epsilon_{\theta d} \triangleq \epsilon_\theta\left(x_d\right)$.

6.4 Simulation of Experience

Since computation of the integral in (6–13) is, in general, intractable, simulation of experience is implemented by approximation of the integral with a summation over finitely many points in the state space. The following assumption facilitates the aforementioned approximation.

Assumption 6.2. [97] There exists a finite set of points $\{\zeta_i \in \mathcal{C} \mid i = 1, \cdots, N\}$ such that

$$0 < \underline{c} \triangleq \frac{1}{N}\left(\inf_{t\in\mathbb{R}_{\geq t_0}}\left(\lambda_{\min}\left\{\sum_{i=1}^{N}\frac{\omega_i\omega_i^T}{\rho_i}\right\}\right)\right), \tag{6–16}$$

where $\rho_i \triangleq 1 + \nu\omega_i^T\Gamma\omega_i \in \mathbb{R}$, and $\omega_i \triangleq \nabla\sigma\left(\zeta_i\right)\left(F_\theta\left(\zeta_i, \hat{\theta}\right) + F_1\left(\zeta_i\right) + G\left(\zeta_i\right)\hat{\mu}\left(\zeta_i, \hat{W}_a\right)\right).$

[1] The expression in (6–15) is developed to facilitate the stability analysis, whereas the equivalent expression in (6–14) is implemented in practice.

Using Assumption 6.2, simulation of experience is implemented by the weight update laws

$$\dot{\hat{W}}_c = -\eta_{c1}\Gamma\frac{\omega}{\rho}\hat{\delta}_t - \frac{\eta_{c2}}{N}\Gamma\sum_{i=1}^{N}\frac{\omega_i}{\rho_i}\hat{\delta}_{ti}, \tag{6–17}$$

$$\dot{\Gamma} = \left(\beta\Gamma - \eta_{c1}\Gamma\frac{\omega\omega^T}{\rho^2}\Gamma\right)\mathbf{1}_{\left\{\|\Gamma\|\leq\overline{\Gamma}\right\}}, \ \|\Gamma\left(t_0\right)\| \leq \overline{\Gamma}, \tag{6–18}$$

$$\dot{\hat{W}}_a = -\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a + \left(\frac{\eta_{c1}G_\sigma^T\hat{W}_a\omega^T}{4\rho} + \sum_{i=1}^{N}\frac{\eta_{c2}G_{\sigma i}^T\hat{W}_a\omega_i^T}{4N\rho_i}\right)\hat{W}_c, \tag{6–19}$$

where $\omega \triangleq \nabla\sigma\left(\zeta\right)\left(F_\theta\left(\zeta,\hat{\theta}\right) + F_1\left(\zeta\right) + G\left(\zeta\right)\hat{\mu}\left(\zeta,\hat{W}_a\right)\right)$, $\Gamma \in \mathbb{R}^{L\times L}$ is the least-squares gain matrix, $\overline{\Gamma} \in \mathbb{R}$ denotes a positive saturation constant, $\beta \in \mathbb{R}$ denotes the forgetting factor, $\eta_{c1}, \eta_{c2}, \eta_{a1}, \eta_{a2} \in \mathbb{R}$ denote constant positive adaptation gains, $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function of the set $\{\cdot\}$, $G_\sigma \triangleq \sigma'\left(\zeta\right)G\left(\zeta\right)R^{-1}G^T\left(\zeta\right)\left(\nabla\sigma\left(\zeta\right)\right)^T$, and $\rho \triangleq 1 + \nu\omega^T\Gamma\omega$, where $\nu \in \mathbb{R}$ is a positive normalization constant. In (6–17)-(6–19) and in the subsequent development, for any function $\xi\left(\zeta,\cdot\right)$, the notation ξ_i, is defined as $\xi_i \triangleq \xi\left(\zeta_i,\cdot\right)$, and the instantaneous BEs $\hat{\delta}_t$ and $\hat{\delta}_{ti}$ are defined as

$$\hat{\delta}_t\left(t\right) \triangleq \hat{\delta}\left(\zeta\left(t\right), \hat{W}_c\left(t\right), \hat{W}_a\left(t\right), \hat{\theta}\left(t\right)\right) \tag{6–20}$$

and $\hat{\delta}_{ti}\left(t\right) \triangleq \hat{\delta}\left(\zeta_i, \hat{W}_c\left(t\right), \hat{W}_a\left(t\right), \hat{\theta}\left(t\right)\right)$. The saturated least-squares update law in (6–18) ensures that there exist positive constants $\underline{\gamma}, \overline{\gamma} \in \mathbb{R}$ such that

$$\underline{\gamma} \leq \left\|\left(\Gamma\left(t\right)\right)^{-1}\right\| \leq \overline{\gamma}, \ \forall t \in \mathbb{R}. \tag{6–21}$$

6.5 Stability Analysis

If the state penalty function $\overline{Q}$ is positive definite, then the optimal value function V^* is positive definite, and serves as a Lyapunov function for the system in (5–4) under the optimal control policy μ^*; hence, V^* is used (cf. [57, 59, 145]) as a candidate Lyapunov function for the closed-loop system under the policy $\hat{\mu}$. Based on the definition in (6–1), the function $\overline{Q}$, and hence, the function V^* are positive semidefinite; hence, the function V^* is not a valid candidate Lyapunov function. However, the results in Chapter 5 can be used to show that a nonautonomous form of the optimal value function denoted by $V_t^* : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, defined as

$$V_t^*\left(e,t\right) \triangleq V^*\left(\begin{bmatrix} e \\ x_d\left(t\right) \end{bmatrix}\right), \; \forall e \in \mathbb{R}^n, \; t \in \mathbb{R},$$

is positive definite and decrescent. Hence, $V_t^*\left(0,t\right) = 0, \; \forall t \in \mathbb{R}$ and there exist class $\mathcal{K}$ functions $\underline{v} : \mathbb{R} \to \mathbb{R}$ and $\overline{v} : \mathbb{R} \to \mathbb{R}$ such that

$$\underline{v}\left(\left\|e^o\right\|\right) \leq V_t^*\left(e^o,t\right) \leq \overline{v}\left(\left\|e^o\right\|\right), \tag{6–22}$$

for all $e^o \in \mathbb{R}^n$ and for all $t \in \mathbb{R}$.

To facilitate the stability analysis, a concatenated state $Z \in \mathbb{R}^{2n+2L+n(p+1)}$ is defined as

$$Z \triangleq \begin{bmatrix} e^T & \tilde{W}_c^T & \tilde{W}_a^T & \tilde{x}^T & \left(\text{vec}\left(\tilde{\theta}\right)\right)^T \end{bmatrix}^T,$$

and a candidate Lyapunov function is defined as

$$V_L\left(Z,t\right) \triangleq V_t^*\left(e,t\right) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a + V_0\left(\tilde{\theta},\tilde{x}\right), \tag{6–23}$$

where vec $(\cdot)$ denotes the vectorization operator and V_0 is defined in (6–8). Using (6–8), the bounds in (6–21) and (6–22), and the fact that tr $\left(\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}\right) = \left(\text{vec}\left(\tilde{\theta}\right)\right)^T\left(\Gamma_\theta^{-1} \otimes I_{p+1}\right)\left(\text{vec}\left(\tilde{\theta}\right)\right)$, the candidate Lyapunov function in (6–23) can be bounded as

$$\underline{v_l}\left(\left\|Z^o\right\|\right) \leq V_L\left(Z^o,t\right) \leq \overline{v_l}\left(\left\|Z^o\right\|\right), \tag{6–24}$$

for all $Z^o \in \mathbb{R}^{2n+2L+n(p+1)}$ and for all $t \in \mathbb{R}$, where $\underline{v_l} : \mathbb{R} \to \mathbb{R}$ and $\overline{v_l} : \mathbb{R} \to \mathbb{R}$ are class $\mathcal{K}$ functions.

To facilitate the stability analysis, given any compact set $\chi \subset \mathbb{R}^{2n+2L+n(p+1)}$ containing an open ball of radius $\rho \in \mathbb{R}$ centered at the origin, a positive constant $\iota \in \mathbb{R}$ is defined as

$$\iota \triangleq \frac{3\left(\frac{(\eta_{c1}+\eta_{c2})\overline{W}^{2}\overline{\|G_{\sigma}\|}}{16\sqrt{\nu\underline{\Gamma}}}+\frac{\overline{\|(W^{T}G_{\sigma}+\nabla\epsilon G_{r}\nabla\sigma^{T})\|}}{4}+\frac{\eta_{a2}\overline{W}}{2}\right)^{2}}{(\eta_{a1}+\eta_{a2})}+\overline{\left\|\frac{G_{\epsilon}}{2}\right\|}+\overline{\left\|\frac{W^{T}\nabla\sigma G_{r}\nabla\epsilon^{T}}{2}\right\|}$$
$$+\overline{\left\|W^{T}\nabla\sigma Gg_{d}^{+}\epsilon_{\theta d}\right\|},+\frac{3\left(\left(\overline{\left\|W^{T}\nabla\sigma Gg_{d}^{+}\right\|}+\overline{\left\|\nabla\epsilon Gg_{d}^{+}\right\|}\right)\overline{\sigma_{g}}+k_{\theta}\overline{d_{\theta}}\right)^{2}}{4k_{\theta}\underline{\sigma_{\theta}}}+\frac{(\eta_{c1}+\eta_{c2})^{2}\,\overline{\|\Delta\|}^{2}}{4\nu\underline{\Gamma}\eta_{c2}\underline{c}}+\frac{\overline{\epsilon_{\theta}}^{2}}{2k}$$
$$+\overline{\left\|\nabla\epsilon Gg_{d}^{+}\epsilon_{\theta d}\right\|} \quad \text{(6–25)}$$

where $G_r \triangleq GR^{-1}G^T$, and $G_\epsilon \triangleq \nabla\epsilon G_r\left(\nabla\epsilon\right)^T$. Let $v_l : \mathbb{R} \to \mathbb{R}$ be a class $\mathcal{K}$ function such that

$$v_l\left(\|Z\|\right) \leq \frac{\underline{q}\left(\|e\|\right)}{2}+\frac{\eta_{c2}\underline{c}}{8}\left\|\tilde{W}_c\right\|^2+\frac{(\eta_{a1}+\eta_{a2})}{6}\left\|\tilde{W}_a\right\|^2+\frac{k}{4}\left\|\tilde{x}\right\|^2+\frac{k_\theta\underline{\sigma_\theta}}{6}\left\|\text{vec}\left(\tilde{\theta}\right)\right\|^2.$$

The sufficient gain conditions used in the subsequent Theorem 6.1 are

$$v_l^{-1}\left(\iota\right) < \overline{v_l}^{-1}\left(\underline{v_l}\left(\rho\right)\right) \quad \text{(6–26)}$$

$$\eta_{c2}\underline{c} > \frac{3\left(\eta_{c2}+\eta_{c1}\right)^2\overline{W}^2\overline{\|\nabla\sigma\|}^2\overline{\sigma_g}^2}{4k_\theta\underline{\sigma_\theta}\nu\underline{\Gamma}} \quad \text{(6–27)}$$

$$\left(\eta_{a1}+\eta_{a2}\right) > \frac{3\left(\eta_{c1}+\eta_{c2}\right)\overline{W}\overline{\|G_\sigma\|}}{8\sqrt{\nu\underline{\Gamma}}}+\frac{3}{\underline{c}\eta_{c2}}\left(\frac{\left(\eta_{c1}+\eta_{c2}\right)\overline{W}\overline{\|G_\sigma\|}}{8\sqrt{\nu\underline{\Gamma}}}+\eta_{a1}\right)^2. \quad \text{(6–28)}$$

In (6–25)-(6–28), for any function $\varpi : \mathbb{R}^l \to \mathbb{R}$, $l \in \mathbb{N}$, the notation $\overline{\|\varpi\|}$, denotes $\sup_{y\in\chi\cap\mathbb{R}^l}\|\varpi\left(y\right)\|$ and $\overline{\sigma_g} \triangleq \overline{\|\sigma_\theta\|}+\overline{\left\|gg_d^+\right\|}\overline{\|\sigma_{\theta d}\|}$.

The sufficient condition in (6–26) requires the set χ to be large enough based on the constant ι. Since the NN approximation errors depend on the compact set χ, in general, for a fixed number of NN neurons, the constant ι increases with the size of the set χ. However, for a fixed set χ, the constant ι decreases with decreasing function reconstruction errors, i.e., with increasing number of NN neurons. Hence a sufficiently large number of NN neurons is required to satisfy the condition in (6–26).

Theorem 6.1. *Provided Assumptions 5.2-6.2 hold, and the control gains are selected based on* (6–26)-(6–28)*, the controller in* (6–14)*, along with the weight update laws*

(6–17)-(6–19)*, and the identifier in* (6–5) *along with the weight update law* (6–7) *ensure that the system states remain bounded, the tracking error is uniformly ultimately bounded, and that the control policy* $\hat{\mu}$ *converges to a neighborhood around the optimal control policy* μ^**.*

Proof. Using (5–4) and the fact that $\dot{V}_t^*\left(e\left(t\right),t\right)=\dot{V}^*\left(\zeta\left(t\right)\right),\ \forall t\in\mathbb{R}$, the time-derivative of the candidate Lyapunov function in (6–23) is

$$\dot{V}_L=\nabla_\zeta V^*(F+G\mu^*)-\tilde{W}_c^T\Gamma^{-1}\dot{\hat{W}}_c-\frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c-\tilde{W}_a^T\dot{\hat{W}}_a+\dot{V}_0+\nabla V^*G\mu-\nabla V^*G\mu^*. \tag{6–29}$$

Using (5–9), (6–11), (6–12), and (6–15) the expression in (6–29) is bounded as

$$\begin{aligned}\dot{V}_L\leq&-\overline{Q}\left(\zeta\right)-\tilde{W}_c^T\Gamma^{-1}\dot{\hat{W}}_c-\frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c-\tilde{W}_a^T\dot{\hat{W}}_a+\dot{V}_0+\frac{1}{2}\left(W^TG_\sigma+\nabla\epsilon G_r\nabla\sigma^T\right)\tilde{W}_a\\&+W^T\nabla\sigma Gg_d^+\tilde{\theta}^T\sigma_{\theta d}+\nabla\epsilon Gg_d^+\tilde{\theta}^T\sigma_{\theta d}+\frac{1}{2}G_\epsilon+\frac{1}{2}W^T\nabla\sigma G_r\nabla\epsilon^T+W^T\nabla\sigma Gg_d^+\epsilon_{\theta d}-\left(\mu^*\right)^TR\mu^*\\&+\nabla\epsilon Gg_d^+\epsilon_{\theta d}.\end{aligned} \tag{6–30}$$

The approximate BE in (6–20) is expressed in terms of the weight estimation errors as

$$\hat{\delta}_t=-\omega^T\tilde{W}_c-W^T\nabla\sigma F_{\tilde{\theta}}+\frac{1}{4}\tilde{W}_a^TG_\sigma\tilde{W}_a+\Delta, \tag{6–31}$$

where $F_{\tilde{\theta}}\triangleq F_\theta\left(\zeta,\tilde{\theta}\right)$ and $\Delta=O\left(\overline{\epsilon},\overline{\nabla\epsilon},\overline{\epsilon_\theta}\right)$. Using (6–31), the bound in (6–9) and the update laws in (6–17)-(6–19), the expression in (6–30) is bounded as

$$\begin{aligned}\dot{V}_L\leq&-\overline{Q}\left(\zeta\right)-\sum_{i=1}^N\tilde{W}_c^T\frac{\eta_{c2}}{N}\frac{\omega_i\omega_i^T}{\rho_i}\tilde{W}_c-k_\theta\underline{\sigma_\theta}\left\Vert\tilde{\theta}\right\Vert_F^2-\left(\eta_{a1}+\eta_{a2}\right)\tilde{W}_a^T\tilde{W}_a-k\left\Vert\tilde{x}\right\Vert^2\\&-\eta_{c1}\tilde{W}_c^T\frac{\omega}{\rho}W^T\nabla\sigma F_{\tilde{\theta}}+\eta_{a1}\tilde{W}_a^T\tilde{W}_c+\eta_{a2}\tilde{W}_a^TW+\frac{1}{4}\eta_{c1}\tilde{W}_c^T\frac{\omega}{\rho}\tilde{W}_a^TG_\sigma\tilde{W}_a-\sum_{i=1}^N\tilde{W}_c^T\frac{\eta_{c2}}{N}\frac{\omega_i}{\rho_i}W^T\sigma_i'F_{\tilde{\theta}i}\\&+\sum_{i=1}^N\frac{1}{4}\tilde{W}_c^T\frac{\eta_{c2}}{N}\frac{\omega_i}{\rho_i}\tilde{W}_a^TG_{\sigma i}\tilde{W}_a+\tilde{W}_c^T\frac{\eta_{c2}}{N}\sum_{i=1}^N\frac{\omega_i}{\rho_i}\Delta_i-\tilde{W}_a^T\left(\frac{\eta_{c1}G_\sigma^T\hat{W}_a\omega^T}{4\rho}+\sum_{i=1}^N\frac{\eta_{c2}G_{\sigma i}^T\hat{W}_a\omega_i^T}{4N\rho_i}\right)\hat{W}_c\\&+\overline{\epsilon_\theta}\left\Vert\tilde{x}\right\Vert+k_\theta\overline{d_\theta}\left\Vert\tilde{\theta}\right\Vert_F+\frac{1}{2}\left(W^TG_\sigma+\nabla\epsilon G_r\nabla\sigma^T\right)\tilde{W}_a+W^T\nabla\sigma Gg_d^+\tilde{\theta}^T\sigma_{\theta d}+\nabla\epsilon Gg_d^+\tilde{\theta}^T\sigma_{\theta d}+\frac{1}{2}G_\epsilon\\&+\eta_{c1}\tilde{W}_c^T\frac{\omega}{\rho}\Delta+\frac{1}{2}W^T\nabla\sigma G_r\nabla\epsilon^T+W^T\nabla\sigma Gg_d^+\epsilon_{\theta d}+\nabla\epsilon Gg_d^+\epsilon_{\theta d}.\end{aligned}$$

Segregation of terms, completion of squares, and the use of Young's inequalities yields

$$\begin{aligned}
\dot{V}_L \leq & -\overline{Q}\left(\zeta\right) - \frac{\eta_{c2}\underline{c}}{4}\left\|\tilde{W}_c\right\|^2 - \frac{\left(\eta_{a1}+\eta_{a2}\right)}{3}\left\|\tilde{W}_a\right\|^2 - \frac{k}{2}\left\|\tilde{x}\right\|^2 - \frac{k_\theta\underline{\sigma_\theta}}{3}\left\|\tilde{\theta}\right\|_F^2 \\
& -\left(\frac{\eta_{c2}\underline{c}}{4} - \frac{3\left(\eta_{c2}+\eta_{c1}\right)^2\overline{W}^2\overline{\left\|\nabla\sigma\right\|}^2\overline{\sigma_g}^2}{16k_\theta\underline{\sigma_\theta}\nu\underline{\Gamma}}\right)\left\|\tilde{W}_c\right\|^2 - \left(\frac{\left(\eta_{a1}+\eta_{a2}\right)}{3} - \frac{\left(\eta_{c1}+\eta_{c2}\right)\overline{W}\overline{\left\|G_\sigma\right\|}}{8\sqrt{\nu\underline{\Gamma}}}\right)\left\|\tilde{W}_a\right\|^2 \\
& + \frac{1}{\underline{c}\eta_{c2}}\left(\frac{\left(\eta_{c1}+\eta_{c2}\right)\overline{W}\overline{\left\|G_\sigma\right\|}}{8\sqrt{\nu\underline{\Gamma}}} + \eta_{a1}\right)^2\left\|\tilde{W}_a\right\|^2 + \frac{3\left(\left(\overline{\left\|W^T\nabla\sigma Gg_d^+\right\|} + \overline{\left\|\nabla\epsilon Gg_d^+\right\|}\right)\overline{\sigma_g} + k_\theta\overline{d_\theta}\right)^2}{4k_\theta\underline{\sigma_\theta}} \\
& + \frac{3\left(\frac{\left(\eta_{c1}+\eta_{c2}\right)\overline{W}^2\overline{\left\|G_\sigma\right\|}}{16\sqrt{\nu\underline{\Gamma}}} + \frac{\overline{\left\|\left(W^TG_\sigma+\nabla\epsilon G_r\nabla\sigma^T\right)\right\|}}{4} + \frac{\eta_{a2}\left\|W\right\|}{2}\right)^2}{\left(\eta_{a1}+\eta_{a2}\right)} + \frac{\left(\eta_{c1}+\eta_{c2}\right)^2\overline{\left\|\Delta\right\|}^2}{4\nu\underline{\Gamma}\eta_{c2}\underline{c}} + \frac{\overline{\epsilon_\theta}^2}{2k} \\
& + \overline{\left\|\frac{1}{2}G_\epsilon\right\|} + \overline{\left\|\frac{1}{2}W^T\nabla\sigma G_r\nabla\epsilon^T\right\|} + \overline{\left\|W^T\nabla\sigma Gg_d^+\epsilon_{\theta d}\right\|} + \overline{\left\|\nabla\epsilon Gg_d^+\epsilon_{\theta d}\right\|}, \qquad (6–32)
\end{aligned}$$

for all $Z \in \chi$. Provided the sufficient conditions in (6–27)-(6–28) are satisfied, the expression in (6–32) yields

$$\dot{V}_L \leq -v_l\left(\left\|Z\right\|\right),\ \forall\left\|Z\right\| \geq v_l^{-1}\left(\iota\right), \qquad (6–33)$$

for all $Z \in \chi$. Using (6–24), (6–26), and (6–33) Theorem 4.18 in [149] can be invoked to conclude that every trajectory $Z\left(t\right)$ satisfying $\left\|Z\left(t_0\right)\right\| \leq \overline{v_l}^{-1}\left(\underline{v_l}\left(\rho\right)\right)$, is bounded for all $t \in \mathbb{R}$ and satisfies $\limsup_{t\to\infty}\left\|Z\left(t\right)\right\| \leq \underline{v_l}^{-1}\left(\overline{v_l}\left(v_l^{-1}\left(\iota\right)\right)\right).$

□

6.6 Simulation

6.6.1 Nonlinear System

The effectiveness of the developed technique is demonstrated via numerical simulation on a nonlinear system of the form (5–4), where

$$f = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\\ \theta_4 & \theta_5 & \theta_6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_2\left(cos\left(2x_1\right)+2\right)\end{bmatrix}, \quad g = \begin{bmatrix}0\\ cos\left(2x_1\right)+2\end{bmatrix}. \qquad (6–34)$$

The ideal values of the unknown parameters are $\theta_1 = -1$, $\theta_2 = 1$, $\theta_3 = 0$, $\theta_4 = -0.5$, $\theta_5 = 0$, and $\theta_6 = -0.5$. The control objective is to follow a desired trajectory, which is the solution of the initial value problem

$$\dot{x}_d = \begin{bmatrix} -1 & 1 \\ -2 & 1 \end{bmatrix} x_d, \quad x_d(0) = \begin{bmatrix} 0 \\ 2 \end{bmatrix},$$

while ensuring convergence of the estimated policy $\hat{\mu}$ to a neighborhood of the policy μ^*, such that the control law $\mu(t) = \mu^*(\zeta(t))$ minimizes the cost $\int_0^\infty \left(e^T(t)\,\text{diag}\left([10,\ 10]\right)e(t) + (\mu(t))^2\right)dt$, subject to the dynamic constraint in (5–4).

The value function is approximated using the polynomial basis $\sigma(\zeta) = \left[e_1^2, e_2^2, e_1^2 x_{d1}^2, e_2^2 x_{d2}^2, e_2^2 x_{d1}^2, e_1^2 x_{d2}^2, e_1 e_2\right]^T$, and the unknown drift dynamics are approximated using the basis $\sigma_\theta(x) = [x_1, x_2, x_2(cos(2x_1) + 2)]^T$. Learning gains for system identification and value function approximation are selected as

$$\eta_{c1} = 0.1,\ \eta_{c2} = 2.5,\ \eta_{a1} = 1,\ \eta_{a2} = 0.01,\ \beta = 0.3,\ \nu = 0.005,\ \overline{\Gamma} = 100000,\ k = 500,$$
$$\Gamma_\theta = I_3,\ \Gamma(0) = 5000 I_9,\ k_\theta = 20,$$

To implement BE extrapolation, error values $\{\zeta_i\}_{i=1}^{81}$ are selected to be uniformly spaced over the a $2 \times 2 \times 2 \times 2$ hypercube centered at the origin. The history stack required for CL contains ten points, and is recorded online using a singular value maximizing algorithm (cf. [93]), and the required state derivatives are computed using a fifth order Savitzky-Golay smoothing filter (cf. [150]).

The initial values for the state and the state estimate are selected to be $x(0) = [1, 2]^T$ and $\hat{x}(0) = [0, 0]^T$, respectively. The initial values for the NN weights for the value function, the policy, and the drift dynamics are selected to be $5 \times \mathbf{1}_7$, $3 \times \mathbf{1}_7$, and $\mathbf{0}_6$, respectively. Since the system in (6–34) has no stable equilibria, the initial policy $\hat{\mu}(\zeta, \mathbf{0}_{6\times 1})$ is not stabilizing. The stabilization demonstrated in Figure 6-1 is achieved via fast simultaneous learning of the system dynamics and the value function.

Figure 6-1 and 6-2 demonstrates that the controller remains bounded, the tracking error is regulated to the origin, and the NN weights converge. In Figure 6-3, the dashed lines denote the ideal values of the NN weights for the system drift dynamics.

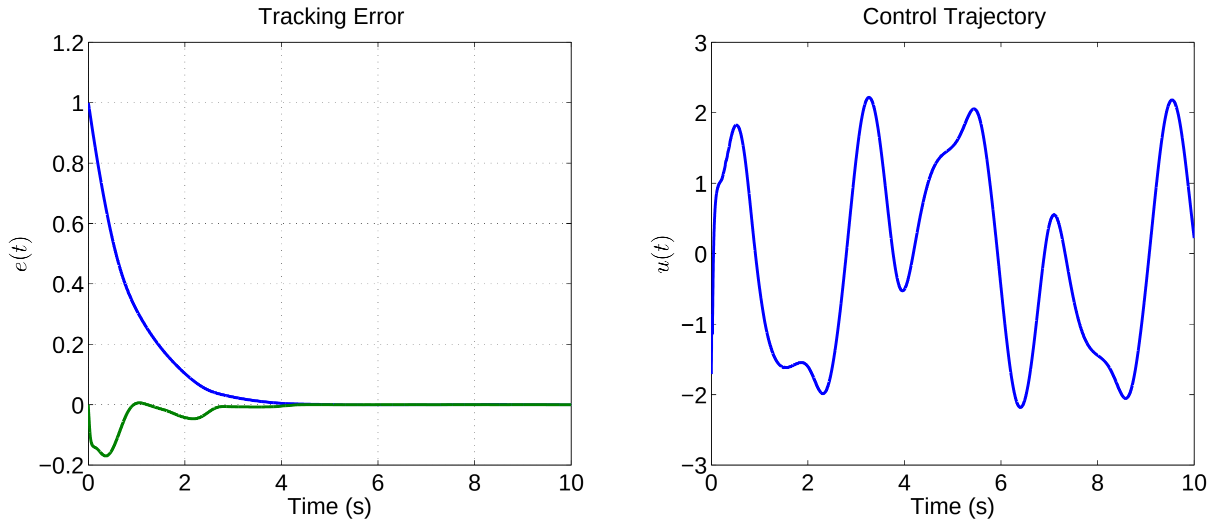

Figure 6-1. System trajectories generated using the proposed method for the nonlinear system.

Figure 6-4 demonstrates satisfaction of the rank conditions in (6–6) and (6–16). The rank condition on the history stack in (6–6) is ensured by selecting points using a singular value maximization algorithm, and the condition in (6–16) is met via oversampling, i.e., by selecting 160 points to identify 9 unknown parameters. Unlike previous results that rely on the addition of an ad-hoc probing signal to satisfy the PE condition, this result ensures sufficient exploration via BE extrapolation.

Since an analytical solution of the optimal tracking problem is not available for the nonlinear system in (6–34), the value function and the policy weights cannot be compared against their ideal values. However, a measure of proximity of the obtained weights $\hat{W}_a^*$ to the ideal weights W can be obtained by comparing the system trajectories resulting from applying the feedback control policy $\hat{\mu}\left(\zeta\right) = -\frac{1}{2}R^{-1}G^T\left(\zeta\right)\nabla\sigma^T\left(\zeta\right)\hat{W}_a^*$ for fixed weights $\hat{W}_a^*$ to the system, against numerically computed optimal system trajectories. Figure 6-5 shows that the control and error trajectories resulting from the

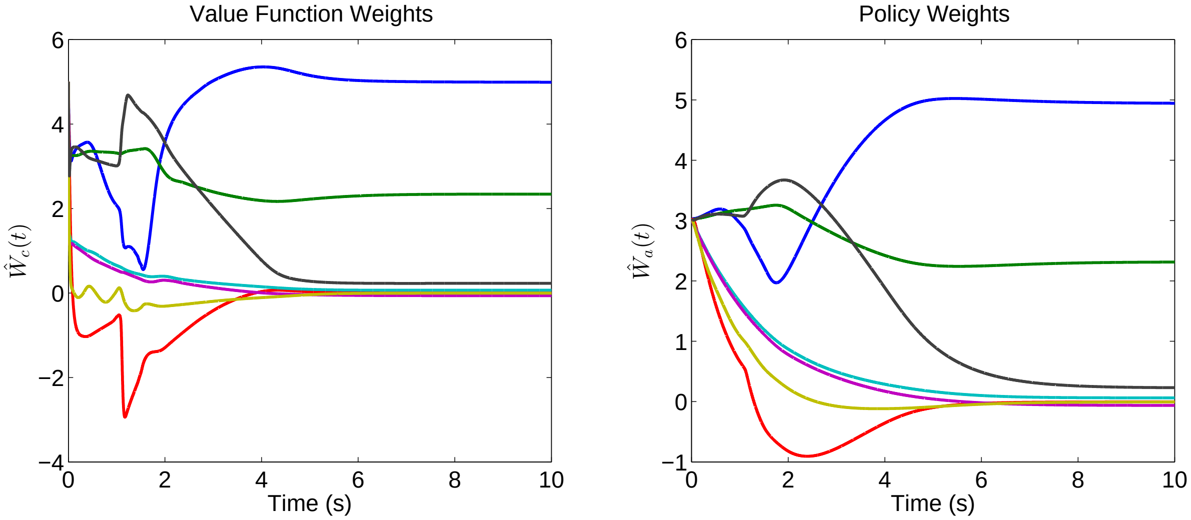

Figure 6-2. Value function and the policy weight trajectories generated using the proposed method for the nonlinear system. Since an analytical solution of the optimal tracking problem is not available, weights cannot be compared against their ideal values

obtained weights are close to the numerical solution. The numerical solution is obtained from GPOPS optimal control software [7] using 300 collocation points.

A comparison between the learned weights and the optimal weights is possible for linear systems provided the dynamics h_d of the desired trajectory are also linear.

6.6.2 Linear System

To demonstrate convergence to the ideal weights, the following linear system is simulated:

$$\dot{x} = \begin{bmatrix} -1 & 1 \\ -0.5 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \tag{6–35}$$

The control objective is to follow a desired trajectory, which is the solution of the initial value problem

$$\dot{x}_d = \begin{bmatrix} -1 & 1 \\ -2 & 1 \end{bmatrix} x_d, \quad x_d(0) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

while ensuring convergence of the estimated policy $\hat{\mu}$ to a neighborhood of the policy μ^*, such that the control law $\mu(t) = \mu^*(\zeta(t))$ minimizes the cost

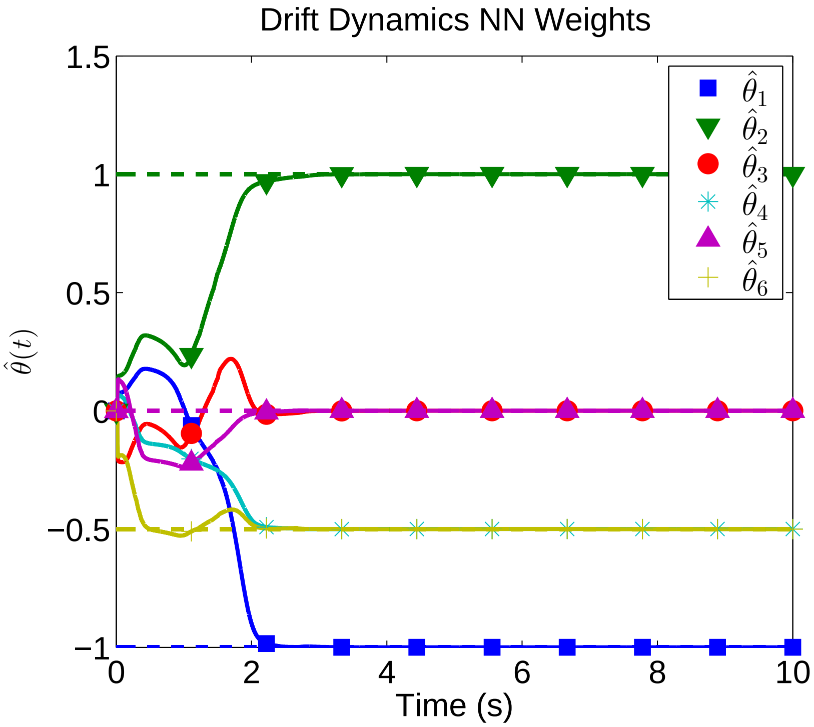

Figure 6-3. Trajectories of the unknown parameters in the system drift dynamics for the nonlinear system. The dotted lines represent the true values of the parameters.

$\int_0^\infty \left(e^T(t)\,\text{diag}\left([10,\ 10]\right)e(t) + (\mu(t))^2\right)dt$, subject to the dynamic constraint in (5–4), over $\mu \in U$.

The value function is approximated using the polynomial basis $\sigma(\zeta) = [e_1^2,\ e_2^2,\ e_1e_2,\ e_1x_{d1},\ e_2x_{d2},\ e_1x_{d2},\ e_2x_{d1}]^T$, and the unknown drift dynamics is approximated using the linear basis $\sigma_\theta(x) = [x_1, x_2]^T$. Learning gains for system identification and value function approximation are selected as

$$\eta_{c1} = 0.5,\ \eta_{c2} = 10,\ \eta_{a1} = 10,\ \eta_{a2} = 0.001,\ \beta = 0.1,\ \nu = 0.005,\ \overline{\Gamma} = 100000,\ k = 500,$$
$$\Gamma_\theta = I_2,\ \Gamma(0) = 1000I_7,\ k_\theta = 10,$$

To implement BE extrapolation, error values $\{e_i\}_{i=1}^{25}$ are selected to be uniformly spaced in a 5×5 grid on a 2×2 square around the origin, and the points $\{x_d(t_j)\}_{j=1}^{11}$ are selected along the desired trajectory such that the time instances t_j are linearly spaced over the

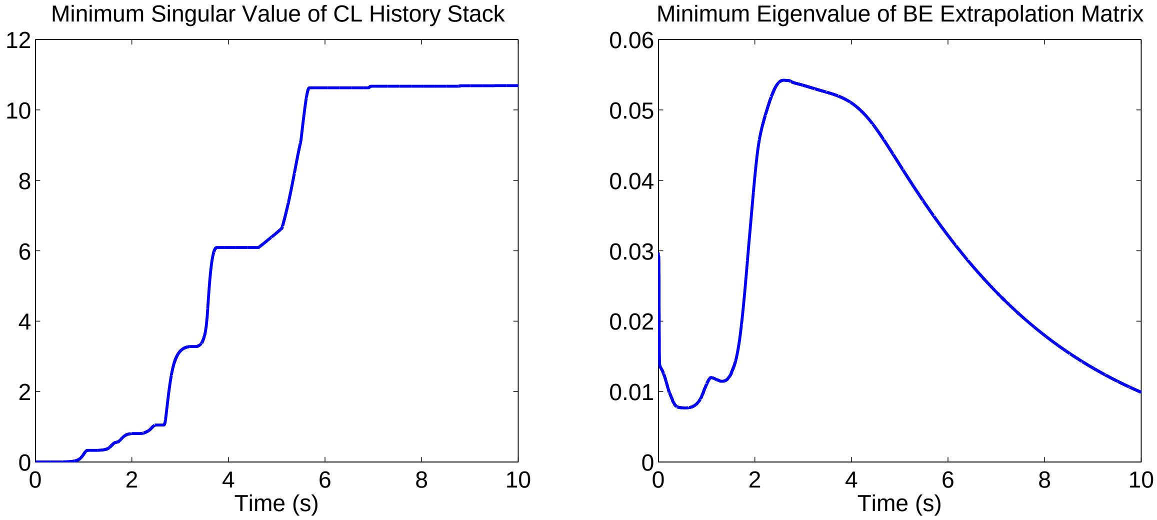

Figure 6-4. Satisfaction of Assumptions 6.1 and 6.2 for the nonlinear system.

interval $[0.1, 2\pi]$. The set of points $\{\zeta_k\}_{k=1}^{275}$ is then computed as $\{\zeta_k\} = \left\{ \begin{bmatrix} e_i^T & x_d^T(t_j) \end{bmatrix}^T \right\}$, $i = 1, \cdots, 25$, $j = 1, \cdots, 11$. The history stack required for CL contains ten points, and is recorded online using a singular value maximizing algorithm (cf. [93]), and the required state derivatives are computed using a fifth order Savitzky-Golay smoothing filter (cf. [150]).

The linear system in (6–35) and the linear desired dynamics result in a linear time-invariant concatenated system. Since the system is linear, the optimal tracking problem reduces to an optimal regulation problem, which can be solved by solving the resulting algebraic Riccati equation. The optimal value function is given by $V(\zeta) = \zeta^T P_\zeta \zeta$, where the matrix P_ζ is given by

$$P_\zeta = \begin{bmatrix} 4.43 & 0.67 & 0 & 0 \\ 0.67 & 2.91 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Using the matrix P_ζ, the ideal weighs corresponding to the selected basis can be computed as $W = [4.43,\ 1.35,\ 0,\ 0,\ 2.91,\ 0,\ 0]$.

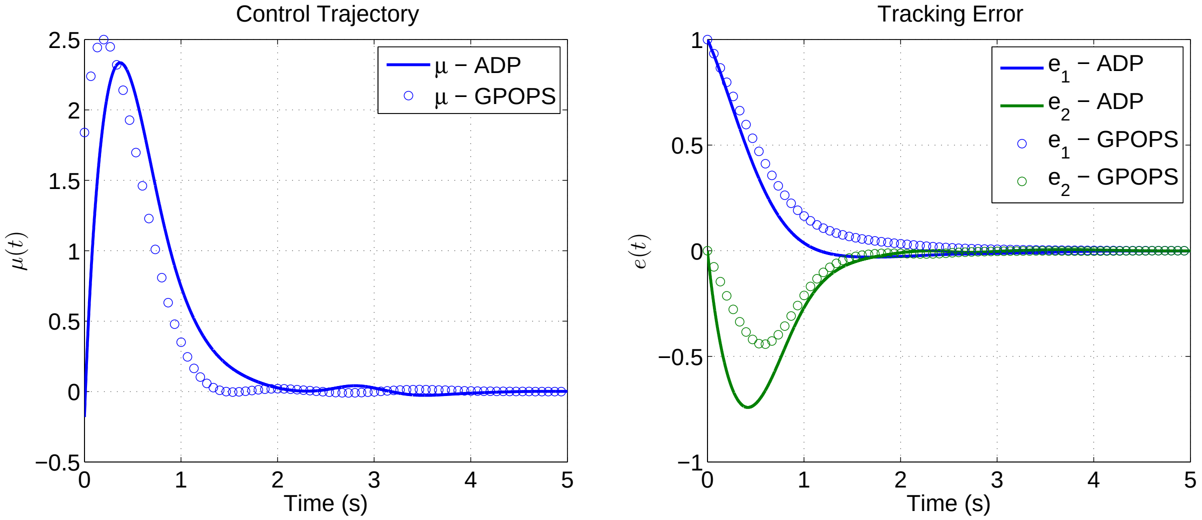

Figure 6-5. Comparison between control and error trajectories resulting from the developed technique and a numerical solution for the nonlinear system.

Figures 6-6 - 6-8 demonstrate that the controller remains bounded, the tracking error goes to zero, and the weight estimates $\hat{W}_c$, $\hat{W}_a$ and $\hat{\theta}$ go to their true values, establishing the convergence of the approximate policy to the optimal policy. Figure 6-9 demonstrates satisfaction of the rank conditions in (6–6) and (6–16).

6.7 Concluding Remarks

A concurrent-learning based implementation of model-based RL is developed to obtain an approximate online solution to infinite-horizon optimal tracking problems for nonlinear continuous-time control-affine systems. The desired steady-state controller is used to facilitate the formulation of a feasible optimal control problem, and the system state is augmented with the desired trajectory to facilitate the formulation of a stationary optimal control problem. A CL-based system identifier is developed to remove the dependence of the desired steady-state controller on the system drift dynamics, and to facilitate simulation of experience via BE extrapolation.

The design variable in (5–5) and inversion of the control effectiveness matrix is necessary because the controller does not asymptotically go to zero, causing the total cost to be infinite for any policy. The definition of μ and the inversion of the control

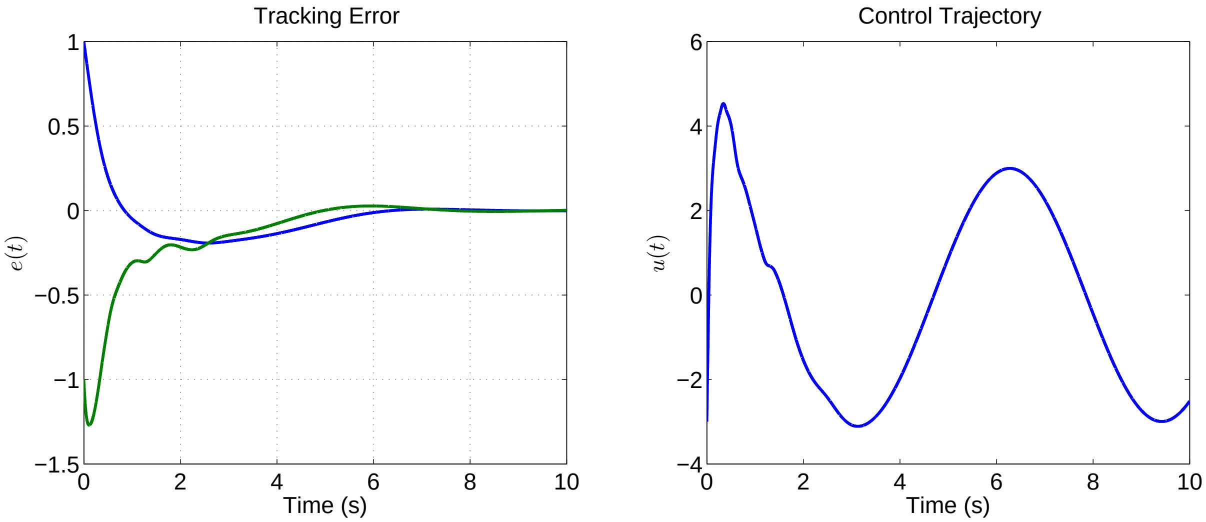

Figure 6-6. System trajectories generated using the proposed method for the linear system.

effectiveness matrix can be avoided if the optimal control problem is formulated in terms of a discounted cost. An online solution of the discounted cost optimal control problem is possible by making minor modifications to the technique developed in this chapter.

The history stack in Assumption 6.1 is assumed to be available a priori for ease of exposition. Provided the system states are exciting over a finite amount of time needed for collection, the history stack can be collected online. For the case when a history stack is not available initially, the developed controller needs to be modified during the data collection phase to ensure stability. The required modifications are similar to those described in Appendix A. Once the condition in Assumption 6.1 is met, the developed controller can be used thereafter.

Technical challenges similar to the optimal tracking problem are encountered while dealing with multiple interacting agents. Since the trajectory of one agent is influenced by other agents, the value function becomes time-varying. The following chapter extends the simulation-based ACI method to obtain an approximate feedback-Nash equilibrium solution to a class of graphical games based on ideas developed in previous chapters.

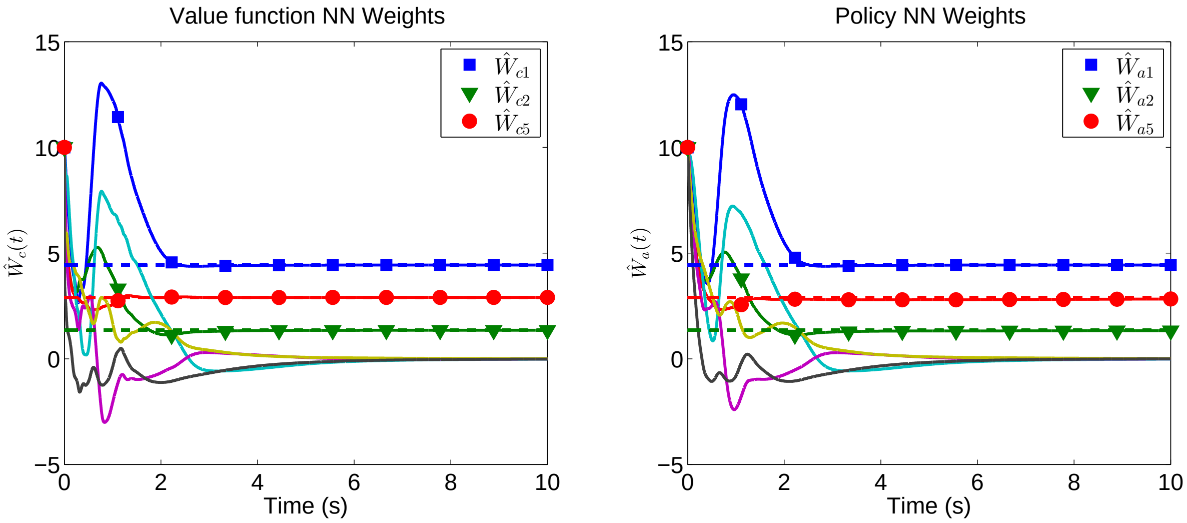

Figure 6-7. Value function and the policy weight trajectories generated using the proposed method for the linear system. Dotted lines denote the ideal values generated by solving the LQR problem.

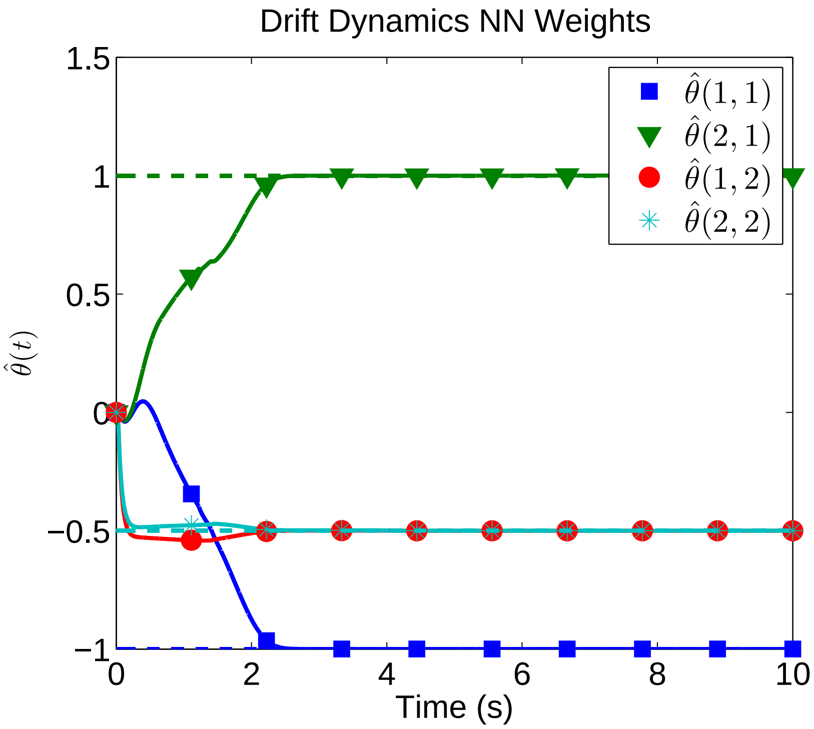

Figure 6-8. Trajectories of the unknown parameters in the system drift dynamics for the linear system. The dotted lines represent the true values of the parameters.

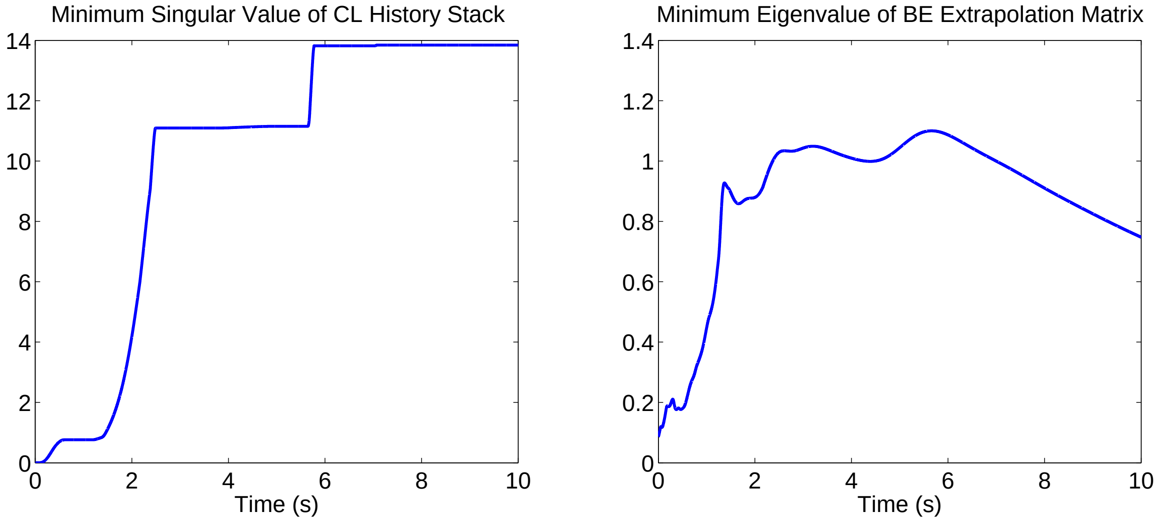

Figure 6-9. Satisfaction of Assumptions 6.1 and 6.2 for the linear system.

CHAPTER 7
MODEL-BASED REINFORCEMENT LEARNING FOR ONLINE APPROXIMATE FEEDBACK-NASH EQUILIBRIUM SOLUTION OF DIFFERENTIAL GRAPHICAL GAMES

Efforts in this chapter seek to combine differential game theory with the ADP framework to determine forward-in-time, approximate optimal controllers for formation tracking in multi-agent systems with uncertain nonlinear dynamics. A continuous control strategy is proposed, using communication feedback from extended neighbors on a communication topology that has a spanning tree. The simulation-based ACI architecture from Chapter 3 is extended to cooperatively control a group of agents to track a trajectory in a desired formation using ideas from Chapter 6.

7.1 Graph Theory Preliminaries

Consider a set of N autonomous agents moving in the state space $\mathbb{R}^n$. The control objective is for the agents to track a desired trajectory while maintaining a desired formation. To aid the subsequent design, another agent (henceforth referred to as the leader) is assumed to be traversing the desired trajectory, denoted by $x_0 \in \mathbb{R}^n$. The agents are assumed to be on a network with a fixed communication topology modeled as a static directed graph (i.e. digraph).

Each agent forms a node in the digraph. The set of all nodes excluding the leader is denoted by $\mathcal{N} = \{1, \cdots N\}$. If node i can receive information from node j then there exists a directed edge from the j^{th} to the i^{th} node of the digraph, denoted by the ordered pair (j,i). Let E denote the set of all edges. Let there be a positive weight $a_{ij} \in \mathbb{R}$ associated with each edge (j,i). Note that $a_{ij} \neq 0$ if and only if $(j,i) \in E$. The digraph is assumed to have no repeated edges i.e. $(i,i) \notin E, \forall i$, which implies $a_{ii} = 0, \forall i$. Note that a_{i0} denotes the edge weight (also referred to as the pinning gain) for the edge between the leader and an node i. Similar to the other edge weights, $a_{i0} \neq 0$ if and only if there exists a directed edge from the leader to the agent i. The neighborhood sets of node i are denoted by $\mathcal{N}_{-i}$ and $\mathcal{N}_i$, defined as $\mathcal{N}_{-i} \triangleq \{j \in \mathcal{N} \mid (j,i) \in E\}$ and

$\mathcal{N}_i \triangleq \mathcal{N}_{-i} \cup \{i\}$. To streamline the analysis, a graph connectivity matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{A} \triangleq [a_{ij} \mid i, j \in \mathcal{N}]$, a diagonal pinning gain matrix $\mathcal{A}_0 \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{A}_0 \triangleq \text{diag}\left(a_{i0}\right) \mid i \in \mathcal{N}$, an in-degree matrix $\mathcal{D} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{D} \triangleq \text{diag}\left(d_i\right)$, where $d_i \triangleq \sum_{j \in \mathcal{N}_i} a_{ij}$, and a graph Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{L} \triangleq \mathcal{D} - \mathcal{A}$. The graph is said to have a spanning tree if given any node β_i, there exists a directed path from the leader β_0 to β_i. A node j is said to be an extended neighbor of node i if there exists a directed path from node j to node i. The extended neighborhood set of node i, denoted by $\mathcal{S}_{-i}$, is defined as the set of all extended neighbors of node i. Formally, $\mathcal{S}_{-i} \triangleq \{j \in \mathcal{N} \mid j \neq i \wedge \exists n \leq N,\ \{j_1, \cdots j_n\} \subset \mathcal{N} \mid \{\left(j, j_1\right), \left(j_1, j_2\right), \cdots, \left(j_n, i\right)\} \subset 2^E\}$. Let $\mathcal{S}_i \triangleq \mathcal{S}_{-i} \cup \{i\}$, and let the edge weights be normalized such that $\sum_j a_{ij} = 1$ for all $i \in \mathcal{N}$. Note that the sub-graphs are nested in the sense that $\mathcal{S}_j \subseteq \mathcal{S}_i$ for all $j \in \mathcal{S}_i$.

7.2 Problem Formulation

The state $x_i \in \mathbb{R}^n$ of each agent evolves according to the control-affine dynamics

$$\dot{x}_i = f_i\left(x_i\right) + g_i\left(x_i\right) u_i, \tag{7–1}$$

where $u_i \in \mathbb{R}^{m_i}$ denotes the control input and $f_i : \mathbb{R}^n \to \mathbb{R}^n$ and $g_i : \mathbb{R}^n \to \mathbb{R}^{n \times m_i}$ are locally Lipschitz continuous functions.

Assumption 7.1. The group of agents follows a virtual leader whose dynamics are described by $\dot{x}_0 = f_0\left(x_0\right)$, where $f_0 : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz continuous function. The function f_0, and the initial condition $x_0\left(t_0\right)$ are selected such that the trajectory $x_0\left(t\right)$ is bounded for all $t \in \mathbb{R}_{\geq t_0}$.

The control objective is for the agents to maintain a predetermined formation around the leader while minimizing a cost function. For all $i \in \mathcal{N}$, the i^{th} agent is aware of its constant desired relative position $x_{dij} \in \mathbb{R}^n$ with respect to all its neighbors $j \in \mathcal{N}_{-i}$, such that the desired formation is realized when $x_i - x_j \to x_{dij}$ for all $i, j \in \mathcal{N}$. To facilitate control design, the formation is expressed in terms of a set of constant vectors $\{x_{di0} \in \mathbb{R}^n\}_{i \in \mathcal{N}}$ where each x_{di0} denotes the constant final desired position of agent i with respect to the leader. The vectors $\{x_{di0}\}_{i\in\mathcal{N}}$ are unknown to the agents not connected to the leader, and the known desired inter agent relative position can be expressed in terms of $\{x_{di0}\}_{i\in\mathcal{N}}$ as $x_{dij} = x_{di0} - x_{dj0}$. The control objective is thus satisfied when $x_i \to x_{di0} + x_0$ for all $i \in \mathcal{N}$. To facilitate control design, define the local neighborhood tracking error signal

$$e_i = \sum_{j\in\{0\}\cup\mathcal{N}_{-i}} a_{ij}\left((x_i - x_j) - x_{dij}\right). \tag{7–2}$$

To facilitate analysis, the error signal in (7–2) is expressed in terms of the unknown leader-relative desired positions as

$$e_i = \sum_{j\in\{0\}\cup\mathcal{N}_{-i}} a_{ij}\left((x_i - x_{di0}) - (x_j - x_{dj0})\right). \tag{7–3}$$

Stacking the error signals in a vector $\mathcal{E} \triangleq \begin{bmatrix} e_1^T, & e_2^T, & \cdots, & e_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$ the equation in (7–3) can be expressed in a matrix form

$$\mathcal{E} = \left((\mathcal{L} + \mathcal{A}_0) \otimes I_n\right)(\mathcal{X} - \mathcal{X}_d - \mathcal{X}_0), \tag{7–4}$$

where $\mathcal{X} = \begin{bmatrix} x_1^T, & x_2^T, & \cdots, & x_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, $\mathcal{X}_d = \begin{bmatrix} x_{d10}^T, & x_{d20}^T, & \cdots, & x_{dN0}^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, $\mathcal{X}_0 = \begin{bmatrix} x_0^T, & x_0^T, & \cdots, & x_0^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, I_n denotes an $n \times n$ identity matrix, and $\otimes$ denotes the Kronecker product. Using (7–4), it can be concluded that provided the matrix $\left((\mathcal{L} + \mathcal{A}_0) \otimes I_n\right) \in \mathbb{R}^{nN\times nN}$ is nonsingular, $\|\mathcal{E}\| \to 0$ implies $x_i \to x_{di0} + x_0$ for all i, and hence, the satisfaction of the control objective. The matrix $\left((\mathcal{L} + \mathcal{A}_0) \otimes I_n\right)$ can be shown to be nonsingular provided the graph has a spanning tree with the leader at the root. To facilitate the formulation of an optimization problem, the following section explores the functional dependence of the state value functions for the network of agents.

7.2.1 Elements of the Value Function

The dynamics for the open-loop neighborhood tracking error are

$$\dot{e}_i = \sum_{j \in \{0\} \cup \mathcal{N}_{-i}} a_{ij} \left(f_i \left(x_i \right) + g_i \left(x_i \right) u_i - f_j \left(x_j \right) - g_j \left(x_j \right) u_j \right).$$

Under the temporary assumption that each controller u_i is an error-feedback controller, i.e. $u_i \left(t \right) = \hat{u}_i \left(e_i \left(t \right), t \right)$, the error dynamics are expressed as

$$\dot{e}_i = \sum_{j \in \{0\} \cup \mathcal{N}_{-i}} a_{ij} \left(f_i \left(x_i \right) + g_i \left(x_i \right) \hat{u}_i \left(e_i, t \right) - f_j \left(x_j \right) - g_j \left(x_j \right) \hat{u}_j \left(e_j, t \right) \right).$$

Thus, the error trajectory $\left\{ e_i \left(t \right) \right\}_{t=t_0}^{\infty}$, where t_0 denotes the initial time, depends on $\hat{u}_j \left(e_j \left(t \right), t \right)$, $\forall j \in \mathcal{N}_i$. Similarly, the error trajectory $\left\{ e_j \left(t \right) \right\}_{t=t_0}^{\infty}$ depends on $\hat{u}_k \left(e_k \left(t \right), t \right), \forall k \in \mathcal{N}_j$. Recursively, the trajectory $\left\{ e_i \left(t \right) \right\}_{t=t_0}^{\infty}$ depends on $\hat{u}_j \left(e_j \left(t \right), t \right)$, and hence, on $e_j \left(t \right), \forall j \in \mathcal{S}_i$. Thus, even if the controller for each agent is restricted to use local error feedback, the resulting error trajectories are interdependent. In particular, a change in the initial condition of one agent in the extended neighborhood causes a change in the error trajectories corresponding to all the extended neighbors. Consequently, the value function corresponding to an infinite-horizon optimal control problem where each agent tries to minimize $\int_{t_0}^{\infty} \left(Q \left(e_i \left(\tau \right) \right) + R \left(u_i \left(\tau \right) \right) \right) d\tau$, where $Q : \mathbb{R}^n \to \mathbb{R}$ and $R : \mathbb{R}^{m_i} \to \mathbb{R}$ are positive definite functions, is dependent on the error states of all the extended neighbors. In other words, the infinite-horizon value of an error state depends on error states of all the extended neighbors; hence, communication with extended neighbors is vital for the solution of an optimal control problem in the presented framework.

7.2.2 Optimal Formation Tracking Problem

When the agents are perfectly tracking the desired trajectory in the desired formation, even though the states of all the agents are different, the time-derivatives of the states of all the agents are identical. Hence, in steady state, the control signal applied by each agent must be such that the time derivatives are all identical. In particular, the relative control signal $u_{ij} \in \mathbb{R}^{m_i}$ that will keep node i in its desired relative position with respect to node j, i.e., $x_i = x_j + x_{dij}$ must be such that the time derivative of x_i is the same as the time derivative of x_j. Using the dynamics of the agent from (7–1), and substituting the desired relative position $x_j + x_{dij}$ for the state x_i, the relative control signal u_{ij} must satisfy

$$f_i\left(x_j + x_{dij}\right) + g_i\left(x_j + x_{dij}\right) u_{ij} = \dot{x}_j. \tag{7–5}$$

The relative steady-state control signal can be expressed in an explicit form provided the following assumption is satisfied.

Assumption 7.2. The matrix $g_i\left(x\right)$ is full rank for all $i \in \mathcal{N}$ and for all $x \in \mathbb{R}^n$, furthermore, the relative steady-state control signal expressed as

$$u_{ij} = f_{ij}\left(x_j\right) + g_{ij}\left(x_j\right) u_j,$$

satisfies (7–5) along the desired trajectory, where $f_{ij}\left(x_j\right) \triangleq g_i^+\left(x_j + x_{dij}\right)\left(f_j\left(x_j\right) - f_i\left(x_j + x_{dij}\right)\right) \in \mathbb{R}^{m_i}$, $g_{ij}\left(x_j\right) \triangleq g_i^+\left(x_j + x_{dij}\right) g_j\left(x_j\right) \in \mathbb{R}^{m_i \times m_j}$, $g_0\left(x\right) \triangleq 0$ for all $x \in \mathbb{R}^n$, $u_{i0} \equiv 0$ for all $i \in \mathcal{N}$, and $g_i^+\left(x\right)$ denotes the Moore-Penrose pseudoinverse of the matrix $g_i\left(x\right)$ for all $x \in \mathbb{R}^n$.

To facilitate the formulation of an optimal formation tracking problem, define the control error $\mu_i \in \mathbb{R}^{m_i}$ as

$$\mu_i \triangleq \sum_{j \in \mathcal{N}_{-i} \cup \{0\}} a_{ij}\left(u_i - u_{ij}\right). \tag{7–6}$$

In the reminder of this chapter, the control errors $\{\mu_i\}$ will be treated as the design variables. In order to implement the controllers $\{u_i\}$ using designed control errors $\{\mu_i\}$, it is essential to invert (7–6). To facilitate the inversion, let $\mathcal{S}_i^o \triangleq \{1, \cdots, s_i\}$, where $s_i \triangleq |\mathcal{S}_i|$. Let $\lambda_i : \mathcal{S}_i^o \to \mathcal{S}_i$ be a bijective map such that $\lambda_i\left(1\right) = i$. For notational brevity, let $\left(\cdot\right)_{\mathcal{S}_i}$ denote the concatenated vector $\left[\left(\cdot\right)_{\lambda_i^1}^T, \left(\cdot\right)_{\lambda_i^2}^T, \cdots, \left(\cdot\right)_{\lambda_i^{s_i}}^T\right]^T$, let $\left(\cdot\right)_{\mathcal{S}_{-i}}$ denote the concatenated vector $\left[\left(\cdot\right)_{\lambda_i^2}^T, \cdots, \left(\cdot\right)_{\lambda_i^{s_i}}^T\right]^T$, let $\sum^i$ denote $\sum_{j \in \mathcal{N}_{-i} \cup \{0\}}$, let λ_i^j denote $\lambda_i\left(j\right)$, let $\mathcal{E}_i \triangleq \left[e_{\mathcal{S}_i}^T, x_{\lambda_i^1}^T\right]^T \in \mathbb{R}^{n(s_i+1)}$, and let $\mathcal{E}_{-i} \triangleq \left[e_{\mathcal{S}_{-i}}^T, x_{\lambda_i^1}^T\right]^T \in \mathbb{R}^{ns_i}$. Then, the control error vector $\mu_{\mathcal{S}_i} \in \mathbb{R}^{\sum_{k \in \mathcal{S}_i^o} m_{\lambda_i^k}}$ can be expressed as

$$\mu_{\mathcal{S}_i} = \mathscr{L}_{gi} \left(\mathcal{E}_i \right) u_{\mathcal{S}_i} - F_i \left(\mathcal{E}_i \right), \tag{7–7}$$

where the matrix $\mathscr{L}_{gi} : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}^{\sum_{k \in \mathcal{S}_i^o} m_{\lambda_i^k} \times \sum_{k \in \mathcal{S}_i^o} m_{\lambda_i^k}}$ is defined as

$$\mathscr{L}_{gi} \left(\mathcal{E}_i \right) \triangleq \begin{bmatrix} \sum^{\lambda_i^1} a_{\lambda_i^1 j} I_{m_{\lambda_i^1}}, \ -a_{\lambda_i^1 \lambda_i^2} g_{\lambda_i^1 \lambda_i^2} \left(x_{\lambda_i^2} \right), \cdots, -a_{\lambda_i^1 \lambda_i^{s_i}} g_{\lambda_i^1 \lambda_i^{s_i}} \left(x_{\lambda_i^{s_i}} \right) \\ -a_{\lambda_i^2 \lambda_i^1} g_{\lambda_i^2 \lambda_i^1} \left(x_{\lambda_i^1} \right), \ \sum^{\lambda_i^2} a_{\lambda_i^2 j} I_{m_{\lambda_i^2}}, \cdots, -a_{\lambda_i^2 \lambda_i^{s_i}} g_{\lambda_i^2 \lambda_i^{s_i}} \left(x_{\lambda_i^{s_i}} \right) \\ \vdots \\ -a_{\lambda_i^{s_i} \lambda_i^1} g_{\lambda_i^{s_i} \lambda_i^1} \left(x_{\lambda_i^1} \right), \ -a_{\lambda_i^{s_i} \lambda_i^2} g_{\lambda_i^{s_i} \lambda_i^2} \left(x_{\lambda_i^2} \right), \cdots, \sum^{\lambda_i^{s_i}} a_{\lambda_i^{s_i} j} I_{m_{\lambda_i^{s_i}}} \end{bmatrix},$$

and $F_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}^{\sum_{k \in \mathcal{S}_i^o} m_{\lambda_i^k}}$ is defined as

$$F_i \left(\mathcal{E}_i \right) \triangleq \begin{bmatrix} \sum^{\lambda_i^1} a_{\lambda_i^1 j} f_{\lambda_i^1 j}^T \left(x_j \right) & \cdots & \sum^{\lambda_i^{s_i}} a_{\lambda_i^{s_i} j} f_{\lambda_i^{s_i} j}^T \left(x_j \right) \end{bmatrix}^T.$$

Assumption 7.3. The matrix $\mathscr{L}_{gi} \left(\mathcal{E}_i \left(t \right) \right)$ is invertible for all $t \in \mathbb{R}$.

Assumption 7.3 is a controllability like condition. Intuitively, Assumption 7.3 requires the control effectiveness matrices to be compatible to ensure the existence of relative control inputs that allow the agents to follow the desired trajectory in the desired formation.

Using Assumption 7.3, the control vector can be expressed as $u_{\mathcal{S}_i} = \mathscr{L}_{gi}^{-1} \left(\mathcal{E}_i \right) \mu_{\mathcal{S}_i} + \mathscr{L}_{gi}^{-1} F_i \left(\mathcal{E}_i \right)$. Let $\mathscr{L}_{gi}^k$ denote the $\left(\lambda_i^{-1} \left(k \right) \right)^{\text{th}}$ block row of $\mathscr{L}_{gi}^{-1}$. Then, the controller u_i can be implemented as

$$u_i = \mathscr{L}_{gi}^i \left(\mathcal{E}_i \right) \mu_{\mathcal{S}_i} + \mathscr{L}_{gi}^i F_i \left(\mathcal{E}_i \right), \tag{7–8}$$

and for any $j \in \mathcal{N}_{-i}$,

$$u_j = \mathscr{L}_{gi}^j \left(\mathcal{E}_i \right) \mu_{\mathcal{S}_i} + \mathscr{L}_{gi}^j F_i \left(\mathcal{E}_i \right). \tag{7–9}$$

Using (7–8) and (7–9), the error and the state dynamics for the agents can be represented as

$$\dot{e}_i = \mathscr{F}_i \left(\mathcal{E}_i \right) + \mathscr{G}_i \left(\mathcal{E}_i \right) \mu_{\mathcal{S}_i}, \tag{7–10}$$

and

$$\dot{x}_i = \mathcal{F}_i\left(\mathcal{E}_i\right) + \mathcal{G}_i\left(\mathcal{E}_i\right)\mu_{\mathcal{S}_i}, \tag{7–11}$$

where $\mathscr{F}_i\left(\mathcal{E}_i\right) \triangleq \sum^i a_{ij}\left(f_i\left(x_i\right) - f_j\left(x_j\right) + \left(g_i\left(x_i\right)\mathscr{L}_{gi}^i\left(\mathcal{E}_i\right) - g_j\left(x_j\right)\mathscr{L}_{gi}^j\left(\mathcal{E}_i\right)\right)F_i\left(\mathcal{E}_i\right)\right)$, $\mathscr{G}_i\left(\mathcal{E}_i\right) \triangleq \sum^i a_{ij}\left(g_i\left(x_i\right)\mathscr{L}_{gi}^i\left(\mathcal{E}_i\right) - g_j\left(x_j\right)\mathscr{L}_{gi}^j\left(\mathcal{E}_i\right)\right)$, $\mathcal{F}_i\left(\mathcal{E}_i\right) \triangleq f_i\left(x_i\right) + g_i\left(x_i\right)\mathscr{L}_{gi}^i\left(\mathcal{E}_i\right)F_i\left(\mathcal{E}_i\right)$, and $\mathcal{G}_i\left(\mathcal{E}_i\right) \triangleq g_i\left(x_i\right)\mathscr{L}_{gi}^i\left(\mathcal{E}_i\right)$.

Let $h_{ei}^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(t, t_0, \mathcal{E}_{i0}\right)$ and $h_{xi}^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(t, t_0, \mathcal{E}_{i0}\right)$ denote the trajectories of (7–10) and (7–11), respectively, with the initial time t_0, initial condition $\mathcal{E}_i\left(t_0\right) = \mathcal{E}_{i0}$, and policies $\overline{\mu}_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}^{m_i}$, and let and $\mathcal{H}_i = \left[\left(h_e\right)_{\mathcal{S}_i}^T, h_{x\lambda_i^1}^T\right]^T$. Define a cost functional

$$J_i\left(e_i, \mu_i\right) \triangleq \int\limits_0^\infty r_i\left(e_i\left(\sigma\right), \mu_i\left(\sigma\right)\right)d\sigma \tag{7–12}$$

where $r_i : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}_{\geq 0}$ denotes the local cost defined as $r_i\left(e_i, \mu_i\right) \triangleq Q_i\left(e_i\right) + \mu_i^T R_i \mu_i$, where $Q_i : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a positive definite function and $R_i \in \mathbb{R}^{m_i \times m_i}$ is a constant positive definite matrix. The objective of each agent is to minimize the cost functional in (7–12). To facilitate the definition of a feedback-Nash equilibrium solution, define value functions $V_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}_{\geq 0}$ as

$$V_i^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(\mathcal{E}_i\right) \triangleq \int\limits_t^\infty r_i\left(h_{ei}^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(\sigma, t, \mathcal{E}_i\right), \overline{\mu}_i\left(\mathcal{H}_i^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(\sigma, t, \mathcal{E}_i\right)\right)\right)d\sigma, \tag{7–13}$$

where the notation $V_i^{\overline{\mu}_i,\overline{\mu}_{\mathcal{S}_{-i}}}\left(\mathcal{E}_i\right)$ denotes the total cost-to-go under the policies $\overline{\mu}_{\mathcal{S}_i}$, starting from the state $\mathcal{E}_i$. Note that the value functions in (7–13) are time-invariant because the dynamical systems $\left\{\dot{e}_j = \mathscr{F}_j\left(\mathcal{E}_i\right) + \mathscr{G}_j\left(\mathcal{E}_i\right)\mu_{\mathcal{S}_j}\right\}_{j \in \mathcal{S}_i}$ and $\dot{x}_i = \mathcal{F}_i\left(\mathcal{E}_i\right) + \mathcal{G}_i\left(\mathcal{E}_i\right)\mu_{\mathcal{S}_i}$ together form an autonomous dynamical system.

A graphical feedback-Nash equilibrium solution within the subgraph $\mathcal{S}_i$ is defined as the tuple of policies $\left\{\mu_j^* : \mathbb{R}^{n(s_j+1)} \to \mathbb{R}^{mj}\right\}_{j \in \mathcal{S}_i}$ such that the value functions in (7–13) satisfy

$$V_j^*\left(\mathcal{E}_j\right) \triangleq V_j^{\mu_j^*,\mu_{\mathcal{S}_{-j}}^*}\left(\mathcal{E}_j\right) \leq V_j^{\overline{\mu}_j,\mu_{\mathcal{S}_{-j}}^*}\left(\mathcal{E}_j\right),$$

for all $j \in \mathcal{S}_i$, for all $\mathcal{E}_i \in \mathbb{R}^{n(s_i+1)}$ and for all admissible policies $\overline{\mu}_j$. Provided a feedback-Nash equilibrium solution exists and the value functions (7–13) are continuously differentiable, the feedback-Nash equilibrium value functions can be characterized in terms of the following system of HJ equations:

$$\sum_{j \in \mathcal{S}_i} \nabla_{e_j} V_i^* \left(\mathcal{E}_i^o\right) \left(\mathscr{F}_j \left(\mathcal{E}_i^o\right) + \mathscr{G}_j \left(\mathcal{E}_i^o\right) \mu_{\mathcal{S}_j}^* \left(\mathcal{E}_i^o\right)\right) + \nabla_{x_i} V_i^* \left(\mathcal{E}_i^o\right) \left(\mathcal{F}_i \left(\mathcal{E}_i^o\right) + \mathcal{G}_i \left(\mathcal{E}_i^o\right) \mu_{\mathcal{S}_i}^* \left(\mathcal{E}_i^o\right)\right)$$
$$+ \overline{Q}_i \left(\mathcal{E}_i^o\right) + \mu_i^{*T} \left(\mathcal{E}_i^o\right) R_i \mu_i^* \left(\mathcal{E}_i^o\right) = 0, \; \forall \mathcal{E}_i^o \in \mathbb{R}^{n(s_i+1)}, \quad \text{(7–14)}$$

where $\overline{Q}_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}$ is defined as $\overline{Q}_i \left(\mathcal{E}_i\right) \triangleq Q_i \left(e_i\right)$.

Theorem 7.1. *Provided a feedback-Nash equilibrium solution exists and that the value functions in* (7–13) *are continuously differentiable, the system of HJ equations in* (7–14) *constitutes a necessary and sufficient condition for feedback-Nash equilibrium.*

Proof. Consider the cost functional in (7–12), and assume that all the extended neighbors of the i^{th} agent follow their feedback-Nash equilibrium policies. The value function corresponding to any admissible policy $\overline{\mu}_i$ can be expressed as

$$V_i^{\overline{\mu}_i, \mu_{\mathcal{S}_{-i}}^*} \left(\left[e_i^T, \, \mathcal{E}_{-i}^T\right]^T\right) = \int_t^\infty r_i \left(h_{ei}^{\overline{\mu}_i, \mu_{\mathcal{S}_{-i}}^*} \left(\sigma, t, \mathcal{E}_i\right), \overline{\mu}_i \left(\mathcal{H}_i^{\overline{\mu}_i, \mu_{\mathcal{S}_{-i}}^*} \left(\sigma, t, \mathcal{E}_i\right)\right)\right) d\sigma.$$

Treating the dependence on $\mathcal{E}_{-i}$ as explicit time dependence define

$$\overline{V}_i^{\overline{\mu}_i, \mu_{\mathcal{S}_{-i}}^*} \left(e_i, t\right) \triangleq V_i^{\overline{\mu}_i, \mu_{\mathcal{S}_{-i}}^*} \left(\left[e_i^T, \, \mathcal{E}_{-i}^T \left(t\right)\right]^T\right), \quad \text{(7–15)}$$

for all $e_i \in \mathbb{R}^n$ and for all $t \in \mathbb{R}_{\geq 0}$. Assuming that the optimal controller that minimizes (7–12) when all the extended neighbors follow their feedback-Nash equilibrium policies exists, and assuming that the optimal value function $\overline{V}_i^* \triangleq \overline{V}_i^{\mu_i^*, \mu_{\mathcal{S}_{-i}}^*}$ exists and is continuously differentiable, optimal control theory for single objective optimization problems (cf. [144]) can be used to derive the following necessary and sufficient condition

$$\frac{\partial \overline{V}_i^*\left(e_i,t\right)}{\partial e_i}\left(\mathscr{F}_i\left(\mathcal{E}_i\right)+\mathscr{G}_i\left(\mathcal{E}_i\right)\mu^*_{\mathcal{S}_i}\left(\mathcal{E}_i\right)\right)+\frac{\partial \overline{V}_i^*\left(e_i,t\right)}{\partial t}=Q_i\left(e_i\right)+\mu_i^{*T}\left(\mathcal{E}_i\right)R_i\mu_i^*\left(\mathcal{E}_i\right). \quad \text{(7–16)}$$

Using (7–15), the partial derivative with respect to the state can be expressed as

$$\frac{\partial \overline{V}_i^*\left(e_i,t\right)}{\partial e_i}=\frac{\partial V_i^*\left(\mathcal{E}_i\right)}{\partial e_i}, \quad \text{(7–17)}$$

for all $e_i \in \mathbb{R}^n$ and for all $t \in \mathbb{R}_{\geq 0}$, and the partial with respect to time can be expressed as

$$\frac{\partial \overline{V}_i^*\left(e_i,t\right)}{\partial t}=\sum_{j\in\mathcal{S}_{-i}}\frac{\partial V_i^*\left(\mathcal{E}_i\right)}{\partial e_j}\left(\mathscr{F}_j\left(\mathcal{E}_i\right)+\mathscr{G}_j\left(\mathcal{E}_i\right)\mu^*_{\mathcal{S}_j}\left(\mathcal{E}_i\right)\right)+\frac{\partial V_i^*\left(\mathcal{E}_i\right)}{\partial x_i}\left(\mathcal{F}_i\left(\mathcal{E}_i\right)+\mathcal{G}_i\left(\mathcal{E}_i\right)\mu^*_{\mathcal{S}_i}\left(\mathcal{E}_i\right)\right), \quad \text{(7–18)}$$

for all $e_i \in \mathbb{R}^n$ and for all $t \in \mathbb{R}_{\geq 0}$. Substituting (7–17) and (7–18) into (7–16) and repeating the process for each i, the system of HJ equations in (7–14) is obtained. □

Minimizing the HJ equations using the stationary condition, the feedback-Nash equilibrium solution is expressed in the explicit form

$$\mu_i^*\left(\mathcal{E}_i^o\right)=-\frac{1}{2}R_i^{-1}\sum_{j\in\mathcal{S}_i}\left(\mathscr{G}_j^i\left(\mathcal{E}_i^o\right)\right)^T\left(\nabla_{e_j}V_i^*\left(\mathcal{E}_i^o\right)\right)^T-\frac{1}{2}R_i^{-1}\left(\mathcal{G}_i^i\left(\mathcal{E}_i^o\right)\right)^T\left(\nabla_{x_i}V_i^*\left(\mathcal{E}_i^o\right)\right)^T, \quad \text{(7–19)}$$

for all $\left(\mathcal{E}_i^o\right)\in\mathbb{R}^{n(s_i+1)}$, where $\mathscr{G}_j^i \triangleq \mathscr{G}_j\frac{\partial \mu^*_{\mathcal{S}_j}}{\partial \mu_i^*}$, and $\mathcal{G}_i^i \triangleq \mathcal{G}_i\frac{\partial \mu^*_{\mathcal{S}_i}}{\partial \mu_i^*}$. Since solution of the system of HJ equations in (7–14) is generally infeasible, the feedback-Nash value functions and the feedback-Nash policies are approximated using parametric approximation schemes as $\hat{V}_i\left(\mathcal{E}_i,\hat{W}_{ci}\right)$ and $\hat{\mu}_i\left(\mathcal{E}_i,\hat{W}_{ai}\right)$, respectively where $\hat{W}_{ci}\in\mathbb{R}^{L_i}$ and $\hat{W}_{ai}\in\mathbb{R}^{L_i}$ are parameter estimates. Substitution of the approximations $\hat{V}_i$ and $\hat{\mu}_i$ in (7–14) leads to a set of BEs δ_i defined as

$$\delta_i\left(\mathcal{E}_i,\hat{W}_{ci},\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)\triangleq\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\hat{V}_i\left(\mathcal{E}_i,\hat{W}_{ci}\right)\left(\mathscr{F}_j\left(\mathcal{E}_j\right)+\mathscr{G}_j\left(\mathcal{E}_j\right)\hat{\mu}_{\mathcal{S}_j}\left(\mathcal{E}_j,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)\right)$$
$$+\nabla_{x_i}\hat{V}_i\left(\mathcal{E}_i,\hat{W}_{ci}\right)\left(\mathcal{F}_i\left(\mathcal{E}_i\right)+\mathcal{G}_i\left(\mathcal{E}_i\right)\hat{\mu}_{\mathcal{S}_i}\left(\mathcal{E}_i,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)\right)-\hat{\mu}_i^T\left(\mathcal{E}_i,\hat{W}_{ai}\right)R\hat{\mu}_i\left(\mathcal{E}_i,\hat{W}_{ai}\right)$$

$$- Q_i \left(e_i \right).$$

Approximate feedback-Nash equilibrium control is realized by tuning the estimates $\hat{V}_i$ and $\hat{\mu}_i$ so as to minimize the Bellman errors δ_i. However, computation of δ_i and that of u_{ij} in (7–6) requires exact model knowledge. In the following, a CL-based system identifier is developed to relax the exact model knowledge requirement. In particular, the developed controllers do not require the knowledge of the system drift functions f_i.

7.3 System Identification

On any compact set $\chi \subset \mathbb{R}^n$ the function f_i can be represented using a NN as

$$f_i \left(x \right) = \theta_i^T \sigma_{\theta i} \left(x \right) + \epsilon_{\theta i} \left(x \right), \tag{7–20}$$

for all $x \in \mathbb{R}^n$, where $\theta_i \in \mathbb{R}^{P_i + 1 \times n}$ denote the unknown output-layer NN weights, $\sigma_{\theta i} : \mathbb{R}^n \to \mathbb{R}^{P_i + 1}$ denotes a bounded NN basis function, $\epsilon_{\theta i} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the function reconstruction error, and $P_i \in \mathbb{N}$ denotes the number of NN neurons. Using the universal function approximation property of single layer NNs, provided the rows of $\sigma_{\theta i} \left(x \right)$ form a proper basis, there exist constant ideal weights θ_i and positive constants $\overline{\theta_i} \in \mathbb{R}$ and $\overline{\epsilon_{\theta i}} \in \mathbb{R}$ such that $\left\Vert \theta_i \right\Vert_F \leq \overline{\theta_i} < \infty$ and $\sup_{x \in \chi} \left\Vert \epsilon_{\theta i} \left(x \right) \right\Vert \leq \overline{\epsilon_{\theta i}}$, where $\left\Vert \cdot \right\Vert_F$ denotes the Frobenius norm.

Assumption 7.4. The bounds $\overline{\theta_i}$ and $\overline{\epsilon_{\theta i}}$ are known for all $i \in \mathcal{N}$.

Using an estimate $\hat{\theta}_i \in \mathbb{R}^{P_i + 1 \times n}$ of the weight matrix θ_i, the function f_i can be approximated by the function $\hat{f}_i : \mathbb{R}^n \times \mathbb{R}^{P_i + 1 \times n} \to \mathbb{R}^n$ defined by $\hat{f}_i \left(x, \hat{\theta} \right) \triangleq \hat{\theta}^T \sigma_{\theta i} \left(x \right).$ Based on (7–20), an estimator for online identification of the drift dynamics is developed as

$$\dot{\hat{x}}_i = \hat{\theta}_i^T \sigma_{\theta i} \left(x_i \right) + g_i \left(x_i \right) u_i + k_i \tilde{x}_i, \tag{7–21}$$

where $\tilde{x}_i \triangleq x_i - \hat{x}_i$, and $k_i \in \mathbb{R}$ is a positive constant learning gain. The following assumption facilitates CL-based system identification.

Assumption 7.5. [92] A history stack containing recorded state-action pairs $\left\{x_i^k, u_i^k\right\}_{k=1}^{M_{\theta i}}$ along with numerically computed state derivatives $\left\{\dot{\bar{x}}_i^k\right\}_{k=1}^{M_{\theta i}}$ that satisfies

$$\lambda_{\min}\left(\sum_{k=1}^{M_{\theta i}} \sigma_{\theta i}^k \left(\sigma_{\theta i}^k\right)^T\right) = \underline{\sigma_{\theta i}} > 0,$$

$$\left\|\dot{\bar{x}}_i^k - \dot{x}_i^k\right\| < \overline{d_i}, \, \forall k, \tag{7–22}$$

is available a priori. In (7–22), $\sigma_{\theta i}^k \triangleq \sigma_{\theta i}\left(x_i^k\right)$, $\overline{d_{\theta i}} \in \mathbb{R}$ is a known positive constant, and $\lambda_{\min}\left(\cdot\right)$ denotes the minimum eigenvalue.

The weight estimates $\hat{\theta}_i$ are updated using the following CL-based update law:

$$\dot{\hat{\theta}}_i = k_{\theta i}\Gamma_{\theta i}\sum_{k=1}^{M_{\theta i}} \sigma_{\theta i}^k \left(\dot{\bar{x}}_i^k - g_i^k u_i^k - \hat{\theta}_i^T \sigma_{\theta i}^k\right)^T + \Gamma_{\theta i}\sigma_{\theta i}\left(x_i\right)\tilde{x}_i^T, \tag{7–23}$$

where $g_i^k \triangleq g_i\left(x_i^k\right)$, $k_{\theta i} \in \mathbb{R}$ is a constant positive CL gain, and $\Gamma_{\theta i} \in \mathbb{R}^{P_i+1\times P_i+1}$ is a constant, diagonal, and positive definite adaptation gain matrix.

To facilitate the subsequent stability analysis, a candidate Lyapunov function $V_{0i} : \mathbb{R}^n \times \mathbb{R}^{P_i+1\times n} \to \mathbb{R}$ is selected as

$$V_{0i}\left(\tilde{x}_i, \tilde{\theta}_i\right) \triangleq \frac{1}{2}\tilde{x}_i^T\tilde{x}_i + \frac{1}{2}\text{tr}\left(\tilde{\theta}_i^T\Gamma_{\theta i}^{-1}\tilde{\theta}_i\right), \tag{7–24}$$

where $\tilde{\theta}_i \triangleq \theta_i - \hat{\theta}_i$ and $\text{tr}\left(\cdot\right)$ denotes the trace of a matrix. Using (7–21)-(7–23), the following bound on the time derivative of V_{0i} is established:

$$\dot{V}_{0i} \leq -k_i\left\|\tilde{x}_i\right\|^2 - k_{\theta i}\underline{\sigma_{\theta i}}\left\|\tilde{\theta}_i\right\|_F^2 + \overline{\epsilon_{\theta i}}\left\|\tilde{x}_i\right\| + k_{\theta i}\overline{d_{\theta i}}\left\|\tilde{\theta}_i\right\|_F, \tag{7–25}$$

where $\overline{d_{\theta i}} \triangleq \overline{d}_i\sum_{k=1}^{M_{\theta i}}\left\|\sigma_{\theta i}^k\right\| + \sum_{k=1}^{M_{\theta i}}\left(\left\|\epsilon_{\theta i}^k\right\|\left\|\sigma_{\theta i}^k\right\|\right)$. Using (7–24) and (7–25) a Lyapunov-based stability analysis can be used to show that $\hat{\theta}_i$ converges exponentially to a neighborhood around θ_i.

7.4 Approximation of the BE and the Relative Steady-state Controller

Using the approximations $\hat{f}_i$ for the functions f_i, the BEs can be approximated as

$$\begin{aligned}\hat{\delta}_i\left(\mathcal{E}_i,\hat{W}_{ci},\left(\hat{W}_a\right)_{\mathcal{S}_j},\hat{\theta}_{\mathcal{S}_i}\right)&\triangleq\nabla_{x_i}\hat{V}_i\left(\mathcal{E}_i,\hat{W}_{ci}\right)\left(\hat{\mathcal{F}}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right)+\mathcal{G}_i\left(\mathcal{E}_i\right)\hat{\mu}_{\mathcal{S}_i}\left(\mathcal{E}_i,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)\right)\\&+\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\hat{V}_i\left(\mathcal{E}_i,\hat{W}_{ci}\right)\left(\hat{\mathscr{F}}_j\left(\mathcal{E}_j,\hat{\theta}_{\mathcal{S}_j}\right)+\mathscr{G}_j\left(\mathcal{E}_j\right)\hat{\mu}_{\mathcal{S}_j}\left(\mathcal{E}_j,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)\right)-Q_i\left(e_i\right)\\&-\hat{\mu}_i^T\left(\mathcal{E}_i,\hat{W}_{ai}\right)R\hat{\mu}_i\left(\mathcal{E}_i,\hat{W}_{ai}\right).\end{aligned}\tag{7–26}$$

In (7–26),

$$\begin{gathered}\hat{\mathscr{F}}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right)\triangleq\sum^i a_{ij}\left(\hat{f}_i\left(x_i,\hat{\theta}_i\right)-\hat{f}_j\left(x_j,\hat{\theta}_j\right)\right)+\sum^i a_{ij}\left(g_i\left(x_i\right)\mathscr{L}_{gi}^i-g_j\left(x_j\right)\mathscr{L}_{gi}^j\right)\hat{F}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right),\\ \hat{\mathcal{F}}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right)\triangleq\hat{\theta}_i^T\sigma_{\theta i}\left(x_i\right)+g_i\left(x_i\right)\mathscr{L}_{gi}^i\hat{F}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right),\\ \hat{F}_i\left(\mathcal{E}_i,\hat{\theta}_{\mathcal{S}_i}\right)\triangleq\left[\left(\sum^{\lambda_i^1}a_{\lambda_i^1 j}\hat{f}_{\lambda_i^1 j}\left(x_{\lambda_i^1},\hat{\theta}_{\lambda_i^1},x_j,\hat{\theta}_j\right)\right)^T,\cdots,\left(\sum^{\lambda_i^{s_i}}a_{\lambda_i^{s_i}j}\hat{f}_{\lambda_i^{s_i}j}\left(x_{\lambda_i^{s_i}},\hat{\theta}_{\lambda_i^{s_i}},x_j,\hat{\theta}_j\right)\right)^T\right]^T,\\ \hat{f}_{ij}\left(x_i,\hat{\theta}_i,x_j,\hat{\theta}_j\right)\triangleq g_i^+\left(x_j+x_{dij}\right)\left(\hat{f}_j\left(x_j,\hat{\theta}_j\right)-\hat{f}_i\left(x_j+x_{dij},\hat{\theta}_i\right)\right).\end{gathered}$$

The approximations $\hat{F}_i$, $\hat{\mathscr{F}}_i$, and $\hat{\mathcal{F}}_i$ are related to the original unknown function as $\hat{F}_i\left(\mathcal{E}_i,\theta_{\mathcal{S}_i}\right)+B_i\left(\mathcal{E}_i\right)=F_i\left(\mathcal{E}_i\right)$, $\hat{\mathscr{F}}_i\left(\mathcal{E}_i,\theta_{\mathcal{S}_i}\right)+\mathscr{B}_i\left(\mathcal{E}_i\right)=\mathscr{F}_i\left(\mathcal{E}_i\right)$, and $\hat{\mathcal{F}}_i\left(\mathcal{E}_i,\theta_{\mathcal{S}_i}\right)+\mathcal{B}_i\left(\mathcal{E}_i\right)=\mathcal{F}_i\left(\mathcal{E}_i\right)$, where B_i, $\mathscr{B}_i$, and $\mathcal{B}_i$ are $O\left(\left(\epsilon_\theta\right)_{\mathcal{S}_i}\right)$ terms that denote bounded function approximation errors.

Using the approximations $\hat{f}_i$, an implementable form of the controllers in (7–8) is expressed as

$$u_{\mathcal{S}_i}=\mathscr{L}_{gi}^{-1}\left(\mathcal{E}_i\right)\hat{\mu}_{\mathcal{S}_i}\left(\mathcal{E}_i,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)+\mathscr{L}_{gi}^{-1}\hat{F}_i\left(\mathcal{E}_i,\theta_{\mathcal{S}_i}\right).\tag{7–27}$$

Using (7–7) and (7–27), an unmeasurable form of the virtual controllers implemented on the systems (7–10) and (7–11) is given by

$$\mu_{\mathcal{S}_i}=\hat{\mu}_{\mathcal{S}_i}\left(\mathcal{E}_i,\left(\hat{W}_a\right)_{\mathcal{S}_j}\right)-\hat{F}_i\left(\mathcal{E}_i,\tilde{\theta}_{\mathcal{S}_i}\right)-B_i\left(\mathcal{E}_i\right).\tag{7–28}$$

7.5 Value Function Approximation

On any compact set $\chi\in\mathbb{R}^{n(s_i+1)}$, the value functions can be represented as

$$V_i^*\left(\mathcal{E}_i^o\right)=W_i^T\sigma_i\left(\mathcal{E}_i^o\right)+\epsilon_i\left(\mathcal{E}_i^o\right),\ \forall\mathcal{E}_i^o\in\mathbb{R}^{n(s_i+1)},\tag{7–29}$$

where $W_i \in \mathbb{R}^{L_i}$ are ideal NN weights, $\sigma_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}^{L_i}$ are NN basis functions and $\epsilon_i : \mathbb{R}^{n(s_i+1)} \to \mathbb{R}$ are function approximation errors. Using the universal function approximation property of single layer NNs, provided $\sigma_i\left(\mathcal{E}_i^o\right)$ forms a proper basis, there exist constant ideal weights W_i and positive constants $\overline{W_i} \in \mathbb{R}$ and $\overline{\epsilon_i}, \overline{\nabla\epsilon_i} \in \mathbb{R}$ such that $\|W_i\| \leq \overline{W_i} < \infty$, $\sup_{\mathcal{E}_i^o \in \chi} \left\|\epsilon_i\left(\mathcal{E}_i^o\right)\right\| \leq \overline{\epsilon_i}$, and $\sup_{\mathcal{E}_i^o \in \chi} \left\|\nabla\epsilon_i\left(\mathcal{E}_i^o\right)\right\| \leq \overline{\nabla\epsilon_i}$.

Assumption 7.6. The constants $\overline{\epsilon_i}$, $\overline{\nabla\epsilon_i}$, and $\overline{W_i}$ are known for all $i \in \mathcal{N}$.

Using (7–19) and (7–29), the feedback-Nash equilibrium policies can be represented as

$$\mu_i^*\left(\mathcal{E}_i^o\right) = -\frac{1}{2}R_i^{-1}G_{\sigma i}\left(\mathcal{E}_i^o\right)W_i - \frac{1}{2}R_i^{-1}G_{\epsilon i}\left(\mathcal{E}_i^o\right), \; \forall \mathcal{E}_i^o \in \mathbb{R}^{n(s_i+1)},$$

where

$$G_{\sigma i}\left(\mathcal{E}_i\right) \triangleq \sum_{j \in \mathcal{S}_i} \left(\mathscr{G}_j^i\left(\mathcal{E}_i\right)\right)^T \left(\nabla_{e_j}\sigma_i\left(\mathcal{E}_i\right)\right)^T + \left(\mathcal{G}_i^i\left(\mathcal{E}_i\right)\right)^T \left(\nabla_{x_i}\sigma_i\left(\mathcal{E}_i\right)\right)^T$$

and

$$G_{\epsilon i}\left(\mathcal{E}_i\right) \triangleq \sum_{j \in \mathcal{S}_i} \left(\mathscr{G}_j^i\left(\mathcal{E}_i\right)\right)^T \left(\nabla_{e_j}\epsilon_i\left(\mathcal{E}_i\right)\right)^T + \left(\mathcal{G}_i^i\left(\mathcal{E}_i\right)\right)^T \left(\nabla_{x_i}\epsilon_i\left(\mathcal{E}_i\right)\right)^T.$$

The value functions and the policies are approximated using NNs as

$$\hat{V}_i\left(\mathcal{E}_i, \hat{W}_{ci}\right) \triangleq \hat{W}_{ci}^T\sigma_i\left(\mathcal{E}_i\right), \qquad \hat{\mu}_i\left(\mathcal{E}_i, \hat{W}_{ai}\right) \triangleq -\frac{1}{2}R_i^{-1}G_{\sigma i}\left(\mathcal{E}_i\right)\hat{W}_{ai}. \tag{7–30}$$

7.6 Simulation of Experience via BE Extrapolation

A consequence of Theorem 7.1 is that the BE provides an indirect measure of how close the weights $\hat{W}_{ci}$ and $\hat{W}_{ai}$ are to the ideal weights W_i. From a reinforcement learning perspective, each evaluation of the BE along the system trajectory can be interpreted experience gained by the critic, and each evaluation of the BE at points not yet visited can be interpreted as simulated experience. In previous results such as [95, 112, 119, 128, 157], the critic is restricted to the experience gained (in other words BEs evaluated) along the system state trajectory. The development in [112, 119, 128, 157] can be extended to employ simulated experience; however, the extension requires exact model knowledge. In results such as [95], the formulation of the BE does not allow for simulation of experience. The formulation in (7–26) employs the system identifier developed in section 7.3 to facilitate approximate evaluation of the BE at off-trajectory points. For notational brevity, the arguments to the functions σ_i, $\hat{\mathscr{F}}_i$, $\mathscr{G}_i$, $\mathcal{G}_i$, $\hat{\mathcal{F}}_i$, $\hat{\mu}_i$, $G_{\sigma i}$, $G_{\epsilon i}$, and ϵ_i are suppressed hereafter

To simulate experience, each agent selects a set of points $\left\{\mathcal{E}_i^k\right\}_{k=1}^{M_i}$ and evaluates the instantaneous BE at the current state, denoted by $\hat{\delta}_{ti}$, and the instantaneous state at the selected points, denoted by $\hat{\delta}_{ti}^k$. The BEs $\hat{\delta}_{ti}$ and $\hat{\delta}_{ti}^k$ are defined as

$$\hat{\delta}_{ti}\left(t\right) \triangleq \hat{\delta}_i\left(\mathcal{E}_i\left(t\right), \hat{W}_{ci}\left(t\right), \left(\hat{W}_a\left(t\right)\right)_{\mathcal{S}_j}, \left(\hat{\theta}\left(t\right)\right)_{\mathcal{S}_i}\right),$$
$$\hat{\delta}_{ti}^k\left(t\right) \triangleq \hat{\delta}_i\left(\mathcal{E}_i^k, \hat{W}_{ci}\left(t\right), \left(\hat{W}_a\left(t\right)\right)_{\mathcal{S}_j}, \left(\hat{\theta}\left(t\right)\right)_{\mathcal{S}_i}\right).$$

Note that once $\left\{e_j\right\}_{j\in\mathcal{S}_i}$ and x_i are selected, the i^{th} agent can compute the states of all the remaining agents in the subgraph.

The critic uses simulated experience to update the value function weights using a least squares-based update law

$$\dot{\hat{W}}_{ci} = -\eta_{c1i}\Gamma_i\frac{\omega_i}{\rho_i}\hat{\delta}_{ti} - \frac{\eta_{c2i}\Gamma_i}{M_i}\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}\hat{\delta}_{ti}^k,$$
$$\dot{\Gamma}_i = \left(\beta_i\Gamma_i - \eta_{c1i}\Gamma_i\frac{\omega_i\omega_i^T}{\rho_i^2}\Gamma_i\right)\mathbf{1}_{\left\{\|\Gamma_i\|\leq\overline{\Gamma}_i\right\}},\ \left\|\Gamma_i\left(t_0\right)\right\| \leq \overline{\Gamma}_i, \tag{7–31}$$

where$\rho_i \triangleq 1 + \nu_i\omega_i^T\Gamma_i\omega_i$, $\Gamma_i \in \mathbb{R}^{L_i\times L_i}$ denotes the time-varying least-squares learning gain, $\overline{\Gamma}_i \in \mathbb{R}$ denotes the saturation constant, and $\eta_{c1i}, \eta_{c2i}, \beta_i, \nu_i \in \mathbb{R}$ are constant positive learning gains. In (7–31),

$$\omega_i \triangleq \sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\hat{\mathscr{F}}_j + \mathscr{G}_j\hat{\mu}_{\mathcal{S}_j}\right) + \nabla_{x_i}\sigma_i\left(\hat{\mathcal{F}}_i + \mathcal{G}_i\hat{\mu}_{\mathcal{S}_i}\right),$$
$$\omega_i^k \triangleq \sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i^k\left(\hat{\mathscr{F}}_j^k + \mathscr{G}_j^k\hat{\mu}_{\mathcal{S}_j}^k\right) + \nabla_{x_i}\sigma_i^k\left(\hat{\mathcal{F}}_i^k + \mathcal{G}_i^k\hat{\mu}_{\mathcal{S}_i}^k\right),$$

where for a function $\phi_i\left(\mathcal{E}_i, (\cdot)\right)$, the notation ϕ_i^k indicates evaluation at $\mathcal{E}_i = \mathcal{E}_i^k$; i.e., $\phi_i^k \triangleq \phi_i\left(\mathcal{E}_i^k, (\cdot)\right)$. The actor updates the policy weights using the following update law derived based on a Lyapunov-based stability analysis:

$$\dot{\hat{W}}_{ai} = -\eta_{a2i}\hat{W}_{ai} + \frac{1}{4}\eta_{c1i}G_{\sigma i}R_i^{-1}G_{\sigma i}^T\hat{W}_{ai}^T\frac{\omega_i^T}{\rho_i}\hat{W}_{ci}^T + \frac{1}{4}\sum_{k=1}^{M_i}\frac{\eta_{c2i}}{M_i}G_{\sigma i}^k R_i^{-1}\left(G_{\sigma i}^k\right)^T\hat{W}_{ai}^T\frac{\left(\omega_i^k\right)^T}{\rho_i^k}\hat{W}_{ci}^T - \eta_{a1i}\left(\hat{W}_{ai} - \hat{W}_{ci}\right), \quad (7\text{–}32)$$

where $\eta_{a1i}, \eta_{a2i} \in \mathbb{R}$ are constant positive learning gains. The following assumption facilitates simulation of experience

Assumption 7.7. [97] For each $i \in \mathcal{N}$, there exists a finite set of M_i points $\left\{\mathcal{E}_i^k\right\}_{k=1}^{M_i}$ such that

$$\underline{\rho_i} \triangleq \frac{\left(\inf_{t\in\mathbb{R}_{\geq 0}}\left(\lambda_{\min}\left\{\sum_{k=1}^{M_i}\frac{\omega_i^k(t)\left(\omega_i^k\right)^T(t)}{\rho_i^k(t)}\right\}\right)\right)}{M_i} > 0,$$

where $\lambda_{\min}$ denotes the minimum eigenvalue, and $\underline{\rho_i} \in \mathbb{R}$ is a positive constant.

7.7 Stability Analysis

To facilitate the stability analysis, the left hand side of (7–14) is subtracted from (7–26) to express the BE in terms of weight estimation errors as

$$\hat{\delta}_{ti} = -\tilde{W}_{ci}^T\omega_i - W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i\right)\hat{\mathcal{F}}_i\left(\mathcal{E}_i, \tilde{\theta}_{\mathcal{S}_i}\right) + \frac{1}{4}\tilde{W}_{ai}^T G_{\sigma i}^T R_i^{-1}G_{\sigma i}\tilde{W}_{ai} - \frac{1}{2}W_i^T G_{\sigma i}^T R_i^{-1}G_{\sigma i}\tilde{W}_{ai} - W_i^T\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i\right)\hat{\mathscr{F}}_j\left(\mathcal{E}_j, \tilde{\theta}_{\mathcal{S}_j}\right) + \frac{1}{2}W_i^T\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i\right)\mathscr{G}_j\mathcal{R}_{\mathcal{S}_j}\left(\tilde{W}_a\right)_{\mathcal{S}_j} + \Delta_i + \frac{1}{2}W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i\right)\mathcal{G}_i\mathcal{R}_{\mathcal{S}_i}\left(\tilde{W}_a\right)_{\mathcal{S}_i}, \quad (7\text{–}33)$$

where $\tilde{(\cdot)} \triangleq (\cdot) - \hat{(\cdot)}$, $\Delta_i = O\left(\epsilon_{\mathcal{S}_i}, \nabla_{e_{\mathcal{S}_i}}\epsilon_i, \left(\epsilon_\theta\right)_{\mathcal{S}_i}\right)$, and $\mathcal{R}_{\mathcal{S}_j} \triangleq$ diag $\left(\left[R_{\lambda_j^1}^{-1}G_{\sigma\lambda_j^1}^T, \cdots, R_{\lambda_j^1}^{-1}G_{\sigma\lambda_j^{s_j}}^T\right]\right)$ is a block diagonal matrix. Consider a set of extended neighbors $\mathcal{S}_p$ corresponding to the p^{th} agent. To analyze asymptotic properties of the agents in $\mathcal{S}_p$, consider the following candidate Lyapunov function

$$V_{Lp}\left(Z_p, t\right) \triangleq \sum_{i\in\mathcal{S}_p}V_{ti}\left(e_{\mathcal{S}_i}, t\right) + \sum_{i\in\mathcal{S}_p}\frac{1}{2}\tilde{W}_{ci}^T\Gamma_i^{-1}\tilde{W}_{ci} + \sum_{i\in\mathcal{S}_p}\frac{1}{2}\tilde{W}_{ai}^T\tilde{W}_{ai} + \sum_{i\in\mathcal{S}_p}V_{0i}\left(\tilde{x}_i, \tilde{\theta}_i\right), \quad (7\text{–}34)$$

where $Z_p \in \mathbb{R}^{(2ns_i+2L_is_i+n(P_i+1)s_i)}$ is defined as

$$Z_p \triangleq \left[e_{\mathcal{S}_p}^T, \left(\tilde{W}_c\right)_{\mathcal{S}_p}^T, \left(\tilde{W}_a\right)_{\mathcal{S}_p}^T, \tilde{x}_{\mathcal{S}_p}^T, \text{vec}\left(\tilde{\theta}_{\mathcal{S}_p}\right)^T\right]^T,$$

vec $(\cdot)$ denotes the vectorization operator, and $V_{ti} : \mathbb{R}^{ns_i} \times \mathbb{R} \to \mathbb{R}$ is defined as

$$V_{ti}\left(e_{\mathcal{S}_i}, t\right) \triangleq V_i^*\left(\left[e_{\mathcal{S}_i}^T,\ x_i^T(t)\right]^T\right), \tag{7–35}$$

$\forall e_{\mathcal{S}_i} \in \mathbb{R}^{ns_i}$, $\forall t \in \mathbb{R}$. Since V_{ti}^* depends on t only through uniformly bounded leader trajectories, Lemmas 1 and 2 from [146] can be used to show that V_{ti} is a positive definite and decrescent function. Thus, using Lemma 4.3 from [149], the following bounds on the candidate Lyapunov function in (7–34) are established

$$\underline{v_{lp}}\left(\left\|Z_p^o\right\|\right) \leq V_{Lp}\left(Z_p^o, t\right) \leq \overline{v_{lp}}\left(\left\|Z_p^o\right\|\right), \tag{7–36}$$

for all $Z_p^o \in \mathbb{R}^{(2ns_i+2L_is_i+n(P_i+1)s_i)}$ and for all t, where $\underline{v_{lp}}, \overline{v_{lp}} : \mathbb{R} \to \mathbb{R}$ are class $\mathcal{K}$ functions.

To facilitate the stability analysis, given any compact ball $\chi_p \subset \mathbb{R}^{2ns_i+2L_is_i+n(P_i+1)s_i}$ of radius $r_p \in \mathbb{R}$ centered at the origin, a positive constant $\iota_p \in \mathbb{R}$ is defined as

$$\begin{aligned}\iota_p \triangleq & \sum_{i\in\mathcal{S}_p}\left(\frac{\overline{\epsilon_{\theta i}}^2}{2k_i} + \frac{3\left(k_{\theta i}\overline{d_{\theta i}} + \overline{\left\|A_i^\theta\right\|\left\|B_i^\theta\right\|}\right)^2}{4k_{\theta i}\underline{\sigma_{\theta i}}}\right) + \sum_{i\in\mathcal{S}_p}\frac{5\left(\eta_{c1i}+\eta_{c2i}\right)^2\overline{\left\|\frac{\omega_i}{\rho_i}\Delta_i\right\|}^2}{4\eta_{c2i}\underline{\rho_i}} \\ & + \sum_{i\in\mathcal{S}_p}\frac{1}{2}\overline{\left\|\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_i\mathcal{R}_{\mathcal{S}_i}\epsilon_{\mathcal{S}_i} + \sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_j\mathcal{R}_{\mathcal{S}_j}\epsilon_{\mathcal{S}_j}\right\|} \\ & + \sum_{i\in\mathcal{S}_p}\frac{3\left(\frac{1}{4}\left(\eta_{c1i}+\eta_{c2i}\right)\overline{\left\|W_i^T\frac{\omega_i}{\rho_i}W_i^TG_{\sigma i}^TR_i^{-1}G_{\sigma i}\right\|} + \frac{1}{2}\overline{\left\|A_i^{a1}\right\|} + \eta_{a2i}\overline{W_i}\right)^2}{4\left(\eta_{a1i}+\eta_{a2i}\right)} \\ & + \sum_{i\in\mathcal{S}_p}\overline{\left\|\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_jB_j + \nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_iB_i\right\|},\end{aligned}$$

where for any function $\varpi : \mathbb{R}^l \to \mathbb{R}$, $l \in \mathbb{N}$, the notation $\overline{\|\varpi\|}$, denotes $\sup_{y\in\chi_p\cap\mathbb{R}^l}\|\varpi(y)\|$ and A_i^θ, B_i^θ, and A_i^{a1} are uniformly bounded state-dependent terms. Let $v_{lp} : \mathbb{R} \to \mathbb{R}$ be a class $\mathcal{K}$ function such that

$$v_{lp}\left(\left\|Z_p\right\|\right) \leq +\frac{1}{2}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}\underline{\rho_i}}{5}\left\|\tilde{W}_{ci}\right\|^2+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\frac{\left(\eta_{a1i}+\eta_{a2i}\right)}{3}\left\|\tilde{W}_{ai}\right\|^2+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\frac{k_{\theta i}\underline{\sigma_{\theta i}}}{3}\left\|\tilde{\theta}_i\right\|_F^2$$
$$\frac{1}{2}\sum_{i\in\mathcal{S}_p}\underline{q_i}\left(\left\|e_i\right\|\right)+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\frac{k_i}{2}\left\|\tilde{x}_i\right\|^2,$$

where $\underline{q_i}:\mathbb{R}\to\mathbb{R}$ are class $\mathcal{K}$ functions such that $\underline{q_i}\left(\|e\|\right)\leq Q_i\left(e\right),\ \forall e\in\mathbb{R}^n,\ \forall i\in\mathcal{N}$. The sufficient gain conditions used in subsequent Theorem 7.2 are

$$\frac{\eta_{c2i}\underline{\rho_i}}{5}>\sum_{j\in\mathcal{S}_p}\frac{3s_p\mathbf{1}_{j\in\mathcal{S}_i}\left(\eta_{c1i}+\eta_{c2i}\right)^2\overline{\left\|A_{ij}^{1a\theta}\right\|}^2\overline{\left\|B_{ij}^{1a\theta}\right\|}^2}{4k_{\theta j}\underline{\sigma_{\theta j}}}, \tag{7–37}$$

$$\frac{\left(\eta_{a1i}+\eta_{a2i}\right)}{3}>\sum_{j\in\mathcal{S}_p}\frac{5s_p\mathbf{1}_{i\in\mathcal{S}_j}\left(\eta_{c1j}+\eta_{c2j}\right)^2\overline{\left\|A_{ji}^{1ac}\right\|}^2}{16\eta_{c2j}\underline{\rho_j}}+\frac{5\eta_{a1i}^2}{4\eta_{c2i}\underline{\rho_i}}$$
$$+\frac{\left(\eta_{c1i}+\eta_{c2i}\right)\overline{W_i}\left\|\frac{\omega_i}{\rho_i}\right\|\overline{\left\|G_{\sigma i}^TR_i^{-1}G_{\sigma i}\right\|}}{4},$$

$$v_{lp}^{-1}\left(\iota_p\right)<\overline{v_{lp}}^{-1}\left(\underline{v_{lp}}\left(r_p\right)\right), \tag{7–38}$$

where $A_{ij}^{1a\theta}$, $B_{ij}^{1a\theta}$, and A_{ji}^{1ac} are uniformly bounded state-dependent terms.

Theorem 7.2. *Provided Assumptions 7.1-7.7 hold and the sufficient gain conditions in (7–37)-(7–38) are satisfied, the controller in (7–30) along with the actor and critic update laws in (7–31) and (7–32), and the system identifier in (7–21) along with the weight update laws in (7–23) ensure that the local neighborhood tracking errors e_i are ultimately bounded and that the policies $\hat{\mu}_i$ converge to a neighborhood around the feedback-Nash policies μ_i^* for all $i\in\mathcal{N}$.*

Proof. The time derivative of the candidate Lyapunov function in (7–34) is given by

$$\dot{V}_{Lp}=\sum_{i\in\mathcal{S}_p}\dot{V}_{ti}\left(e_{\mathcal{S}_i},t\right)-\frac{1}{2}\sum_{i\in\mathcal{S}_p}\tilde{W}_{ci}^T\Gamma_i^{-1}\dot{\Gamma}_i\Gamma_i^{-1}\tilde{W}_{ci}-\sum_{i\in\mathcal{S}_p}\tilde{W}_{ci}^T\Gamma_i^{-1}\dot{\hat{W}}_{ci}-\sum_{i\in\mathcal{S}_p}\tilde{W}_{ai}^T\dot{\hat{W}}_{ai}+\dot{V}_{0i}\left(\tilde{x}_i,\tilde{\theta}_i\right). \tag{7–39}$$

Using (7–25), the update laws in (7–31) and (7–32), and the definition of V_{ti} in (7–35), the derivative (7–39) can be bounded as

$$\begin{aligned}
\dot{V}_{Lp} \leq & \sum_{i\in\mathcal{S}_p}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\left(\mathscr{F}_j+\mathscr{G}_j\mu_{\mathcal{S}_j}\right)+\sum_{i\in\mathcal{S}_p}\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\left(\mathcal{F}_i+\mathcal{G}_i\mu_{\mathcal{S}_i}\right)+\sum_{i\in\mathcal{S}_p}k_{\theta i}\overline{d_{\theta i}}\left\|\tilde{\theta}_i\right\|_F \\
& -\frac{1}{2}\sum_{i\in\mathcal{S}_p}\tilde{W}_{ci}^T\Gamma_i^{-1}\left(\beta_i\Gamma_i-\eta_{c1i}\Gamma_i\frac{\omega_i\omega_i^T}{\rho_i^2}\Gamma_i\right)\Gamma_i^{-1}\tilde{W}_{ci}-\sum_{i\in\mathcal{S}_p}\tilde{W}_{ai}^T\left(-\eta_{a1i}\left(\hat{W}_{ai}-\hat{W}_{ci}\right)-\eta_{a2i}\hat{W}_{ai}\right) \\
& -\frac{1}{4}\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ai}^TG_{\sigma i}R_i^{-1}G_{\sigma i}^T\hat{W}_{ai}^T\frac{\omega_i^T}{\rho_i}\hat{W}_{ci}^T+\frac{1}{4}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ai}^T\sum_{k=1}^{M_i}G_{\sigma i}^kR_i^{-1}\left(G_{\sigma i}^k\right)^T\hat{W}_{ai}^T\frac{\left(\omega_i^k\right)^T}{\rho_i^k}\hat{W}_{ci}^T \\
& -\sum_{i\in\mathcal{S}_p}\tilde{W}_{ci}^T\Gamma_i^{-1}\left(-\eta_{c1i}\Gamma_i\frac{\omega_i}{\rho_i}\hat{\delta}_i-\frac{\eta_{c2i}\Gamma_i}{M_i}\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}\hat{\delta}_{ti}^k\right)-\sum_{i\in\mathcal{S}_p}k_i\left\|\tilde{x}_i\right\|^2-\sum_{i\in\mathcal{S}_p}k_{\theta i}\underline{\sigma_{\theta i}}\left\|\tilde{\theta}_i\right\|_F^2+\sum_{i\in\mathcal{S}_p}\overline{\epsilon_{\theta i}}\left\|\tilde{x}_i\right\|.
\end{aligned} \tag{7–40}$$

Using (7–14), (7–28), and (7–33), the derivative in (7–40) can be bounded as

$$\begin{aligned}
\dot{V}_{Lp} \leq & -\sum_{i\in\mathcal{S}_p}Q_i\left(e_i\right)-\frac{1}{2}\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i\omega_i^T}{\rho_i}\tilde{W}_{ci}-\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ci}^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}\omega_i^{kT}\tilde{W}_{ci} \\
& -\sum_{i\in\mathcal{S}_p}\left(\eta_{a1i}+\eta_{a2i}\right)\tilde{W}_{ai}^T\tilde{W}_{ai}+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_j\mathcal{R}_{\mathcal{S}_j}\left(\tilde{W}_a\right)_{\mathcal{S}_j} \\
& +\frac{1}{2}\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{\rho_i}W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i\right)\mathcal{G}_i\mathcal{R}_{\mathcal{S}_i}\left(\tilde{W}_a\right)_{\mathcal{S}_i}+\sum_{i\in\mathcal{S}_p}\eta_{a2i}\tilde{W}_{ai}^TW_i \\
& -\frac{1}{4}\sum_{i\in\mathcal{S}_p}\eta_{c1i}W_i^T\frac{\omega_i}{\rho_i}W_i^TG_{\sigma i}^TR_i^{-1}G_{\sigma i}\tilde{W}_{ai}-\frac{1}{4}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}W_i^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}W_i^TG_{\sigma i}^{kT}R_i^{-1}G_{\sigma i}^k\tilde{W}_{ai} \\
& -\sum_{i\in\mathcal{S}_p}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_j\hat{F}_j\left(\mathcal{E}_j,\tilde{\theta}_{\mathcal{S}_j}\right)-\sum_{i\in\mathcal{S}_p}\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_i\hat{F}_i\left(\mathcal{E}_i,\tilde{\theta}_{\mathcal{S}_i}\right)+\sum_{i\in\mathcal{S}_p}\eta_{a1i}\tilde{W}_{ai}^T\tilde{W}_{ci} \\
& +\frac{1}{2}\sum_{i\in\mathcal{S}_p}\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_i\mathcal{R}_{\mathcal{S}_i}\epsilon_{\mathcal{S}_i}+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_j\mathcal{R}_{\mathcal{S}_j}\epsilon_{\mathcal{S}_j}-\sum_{i\in\mathcal{S}_p}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}V_i^*\left(\mathcal{E}_i\right)\mathscr{G}_jB_j \\
& -\sum_{i\in\mathcal{S}_p}\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_iB_i+\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{\rho_i}\Delta_i-\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ci}^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i^k\right)\hat{\mathcal{F}}_i\left(\mathcal{E}_i^k,\tilde{\theta}_{\mathcal{S}_i}\right) \\
& -\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{\rho_i}W_i^T\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i\right)\hat{\mathscr{F}}_j\left(\mathcal{E}_j,\tilde{\theta}_{\mathcal{S}_j}\right)-\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{\rho_i}W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i\right)\hat{\mathcal{F}}_i\left(\mathcal{E}_i,\tilde{\theta}_{\mathcal{S}_i}\right) \\
& -\sum_{i\in\mathcal{S}_p}\sum_{k=1}^{M_i}\frac{\eta_{c2i}\tilde{W}_{ci}^T\omega_i^k}{M_i\rho_i^k}W_i^T\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i^k\right)\hat{\mathscr{F}}_j\left(\mathcal{E}_j^k,\tilde{\theta}_{\mathcal{S}_j}\right)+\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ci}^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}\Delta_i^k
\end{aligned}$$

$$+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{\rho_i}W_i^T\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i\right)\mathscr{G}_j\mathcal{R}_{\mathcal{S}_j}\left(\tilde{W}_a\right)_{\mathcal{S}_j}+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\nabla_{x_i}V_i^*\left(\mathcal{E}_i\right)\mathcal{G}_i\mathcal{R}_{\mathcal{S}_i}\left(\tilde{W}_a\right)_{\mathcal{S}_i}$$
$$-\sum_{i\in\mathcal{S}_p}\eta_{c1i}\tilde{W}_{ci}^T\frac{\omega_i}{4\rho_i}W_i^TG_{\sigma i}^TR_i^{-1}G_{\sigma i}\tilde{W}_{ai}+\sum_{i\in\mathcal{S}_p}\sum_{k=1}^{M_i}\frac{\eta_{c2i}\tilde{W}_{ci}^T\omega_i^kW_i^T}{2M_i\rho_i^k}\sum_{j\in\mathcal{S}_i}\nabla_{e_j}\sigma_i\left(\mathcal{E}_i^k\right)\mathscr{G}_j^k\mathcal{R}_{\mathcal{S}_j}^k\left(\tilde{W}_a\right)_{\mathcal{S}_j}$$
$$-\frac{1}{4}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ci}^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}W_i^TG_{\sigma i}^{kT}R_i^{-1}G_{\sigma i}^k\tilde{W}_{ai}+\frac{1}{4}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}W_i^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}\tilde{W}_{ai}^TG_{\sigma i}^{kT}R_i^{-1}G_{\sigma i}^k\tilde{W}_{ai}$$
$$+\frac{1}{2}\sum_{i\in\mathcal{S}_p}\frac{\eta_{c2i}}{M_i}\tilde{W}_{ci}^T\sum_{k=1}^{M_i}\frac{\omega_i^k}{\rho_i^k}W_i^T\nabla_{x_i}\sigma_i\left(\mathcal{E}_i^k\right)\mathcal{G}_i^k\mathcal{R}_{\mathcal{S}_i}^k\left(\tilde{W}_a\right)_{\mathcal{S}_i}-\sum_{i\in\mathcal{S}_p}k_i\left\|\tilde{x}_i\right\|^2-\sum_{i\in\mathcal{S}_p}k_{\theta i}\underline{\sigma_{\theta i}}\left\|\tilde{\theta}_i\right\|_F^2$$
$$+\sum_{i\in\mathcal{S}_p}\eta_{c1i}\frac{W_i^T\omega_i}{4\rho_i}\tilde{W}_{ai}^TG_{\sigma i}^TR_i^{-1}G_{\sigma i}\tilde{W}_{ai}+\sum_{i\in\mathcal{S}_p}k_{\theta i}\overline{d_{\theta i}}\left\|\tilde{\theta}_i\right\|_F+\sum_{i\in\mathcal{S}_p}\overline{\epsilon_{\theta i}}\left\|\tilde{x}_i\right\|. \quad (7–41)$$

Using the Cauchy-Schwarz inequality, the Triangle inequality, and completion of squares, the derivative in (7–41) can be bounded as

$$\dot{V}_{Lp}\leq-v_{lp}\left(\left\|Z_p\right\|\right) \quad (7–42)$$

for all $Z_p\in\chi_p$ such that $\left\|Z_p\right\|\geq v_{lp}^{-1}\left(\iota_p\right)$. Using the bounds in (7–36), the sufficient condition in (7–38), and the derivative in (7–42), Theorem 4.18 in [149] can be invoked to conclude that every trajectory $Z_p\left(t\right)$ satisfying $\left\|Z_p\left(t_0\right)\right\|\leq\overline{v_{lp}}^{-1}\left(\underline{v_{lp}}\left(r_p\right)\right)$, is bounded for all $t\in\mathbb{R}$ and satisfies $\limsup_{t\to\infty}\left\|Z_p\left(t\right)\right\|\leq\underline{v_{lp}}^{-1}\left(\overline{v_{lp}}\left(v_{lp}^{-1}\left(\iota_p\right)\right)\right).$

Since the choice of the subgraph $\mathcal{S}_p$ was arbitrary, the neighborhood tracking errors e_i are ultimately bounded for all $i\in\mathcal{N}$. Furthermore, the weight estimates $\hat{W}_{ai}$ converge to a neighborhood of the ideal weights W_i; hence, invoking Theorem 7.1, the policies $\hat{\mu}_i$ converge to a neighborhood of the feedback-Nash equilibrium policies μ_i^* for all $i\in\mathcal{N}$. □

7.8 Simulations

This section provides two simulation examples to demonstrate the applicability of the developed technique. The agents in both the examples are assumed to have the communication topology as shown in Figure 7-1 with unit pinning gains and edge weights. The motion of the agents in the first example is described by identical nonlinear

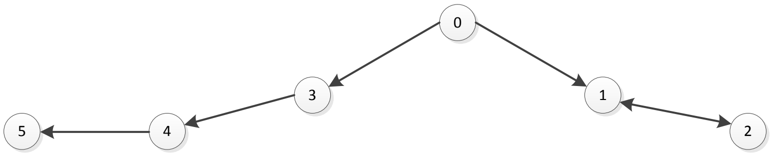

Figure 7-1. Communication topology a network containing five agents.

one-dimensional dynamics, and the motion of the agents in the second example is described by identical nonlinear two-dimensional dynamics.

7.8.1 One-dimensional Example

The dynamics of all the agents are selected to be of the form (7–1) where $f_i\left(x_i\right) = \theta_{i1}x_i + \theta_{i2}x_i^2$, and $g_i\left(x_i\right) = (cos(2x_{i1}) + 2)$ for all $i = 1, \cdots, 5$. The ideal values of the unknown parameters are selected to be $\theta_{i1} = 0$ and $\theta_{i2} = 1$, for all i. The agents start at $x_i = 2$ for all i, and their final desired locations with respect to each other are given by $xd_{12} = 0.5$, $xd_{21} = -0.5$, $xd_{43} = -0.5$, and $xd_{53} = -0.5$. The leader traverses an exponentially decaying trajectory $x_0\left(t\right) = e^{-0.1*t}$. The desired positions of agents 1 and 3 with respect to the leader are $x_{d10} = 0.75$ and $x_{d30} = 1$, respectively.

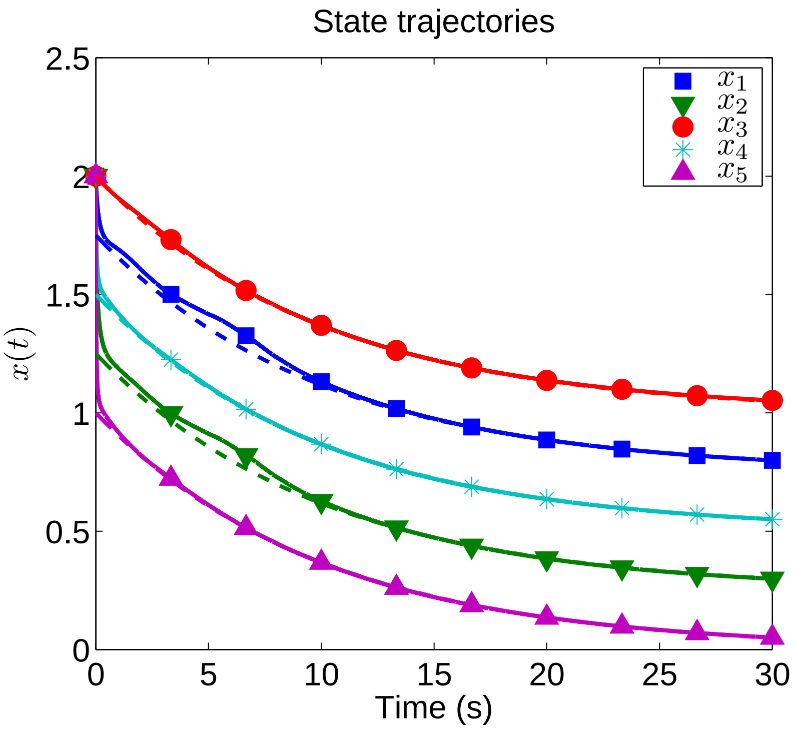

Figure 7-2. State trajectories for the five agents for the one-dimensional example. The dotted lines show the desired state trajectories.

Table 7-1. Simulation parameters for the one-dimensional example.

	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Q_i	10	10	10	10	10
R_i	0.1	0.1	0.1	0.1	0.1
$\sigma_i\left(\mathcal{E}_i\right)$	$\frac{1}{2}[e_1^2,\ \frac{1}{2}e_1^4,\ e_1^2x_1^2,\ e_2^2]^T$	$\frac{1}{2}[e_2^2,\ \frac{1}{2}e_2^4,\ e_2^2x_2^2,\ e_1^2]^T$	$\frac{1}{2}[e_3^2,\ \frac{1}{2}e_3^4,\ e_3^2x_3^2,\ \frac{1}{2}e_3^4x_3^2]^T$	$\dfrac{1}{2}[e_4^2,\ \dfrac{1}{2}e_4^4,\ e_3^2e_4^2,$ $e_4^2x_4^2,\ e_3^2]^T$	$\dfrac{1}{2}[e_5^2,\ \dfrac{1}{2}e_5^4,\ e_4^2e_5^2,\ e_3^2e_5^2,$ $e_5^2x_5^2,\ e_3^2e_4^2,\ e_3^2,\ e_4^2]^T$
$x_i\left(0\right)$	2	2	2	2	2
$\hat{x}_i\left(0\right)$	0	0	0	0	0
$\hat{W}_{ci}\left(0\right)$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{5\times1}$	$3\times\mathbf{1}_{8\times1}$
$\hat{W}_{ai}\left(0\right)$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{4\times1}$	$\mathbf{1}_{5\times1}$	$3\times\mathbf{1}_{8\times1}$
$\hat{\theta}_i\left(0\right)$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$
$\Gamma_i\left(0\right)$	$500I_4$	$500I_4$	$500I_4$	$500I_5$	$500I_8$
η_{c1i}	0.1	0.1	0.1	0.1	0.1
η_{c2i}	10	10	10	10	10
η_{a1i}	5	5	5	5	5
η_{a2i}	0.1	0.1	0.1	0.1	0.1
ν_i	0.005	0.005	0.005	0.005	0.005
$\Gamma_{\theta i}$	I_2	$0.8I_2$	I_2	I_2	I_2
k_i	500	500	500	500	500
$k_{\theta i}$	30	30	25	20	30

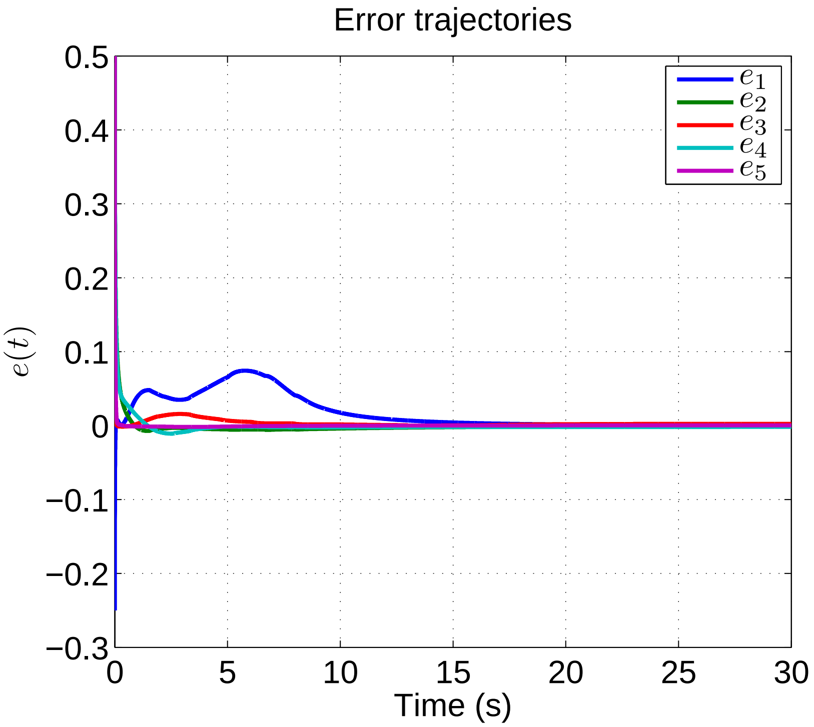

Figure 7-3. Tracking error trajectories for the agents for the one-dimensional example.

Table 7-1 summarizes the optimal control problem parameters, basis functions, and adaptation gains for the agents. For each agent i, five values of e_i, three values of x_i, and three values of errors corresponding to all the extended neighbors are selected for BE extrapolation, resulting in 5×3^{s_i} total values of $\mathcal{E}_i$. All agents estimate the unknown drift parameters using history stacks containing thirty points recorded online using a singular value maximizing algorithm (cf. [93]), and compute the required state derivatives using a fifth order Savitzky-Golay smoothing filter (cf. [150]).

Figures 7-2 - 7-4 show the tracking error, the state trajectories compared with the desired trajectories, and the control inputs for all the agents demonstrating convergence to the desired formation and the desired trajectory. Note that agents 2, 4, and 5 do not have a communication link to the leader, nor do they know their desired relative position from the leader. The convergence to the desired formation is achieved via cooperative control based on decentralized objectives. Figures 7-5 - 7-9 show the evolution and

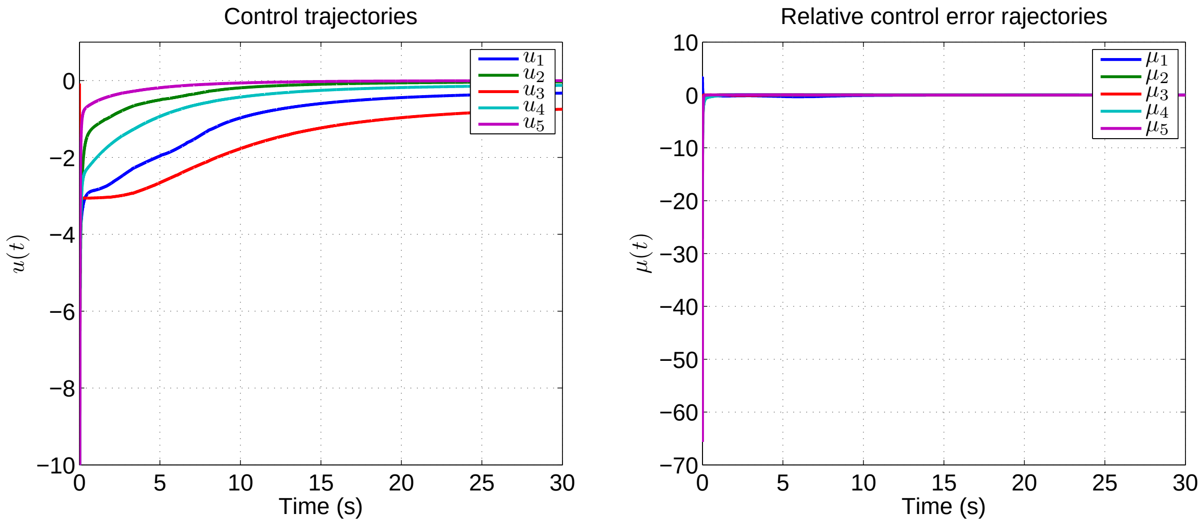

Figure 7-4. Trajectories of the control input and the relative control error for all agents for the one-dimensional example.

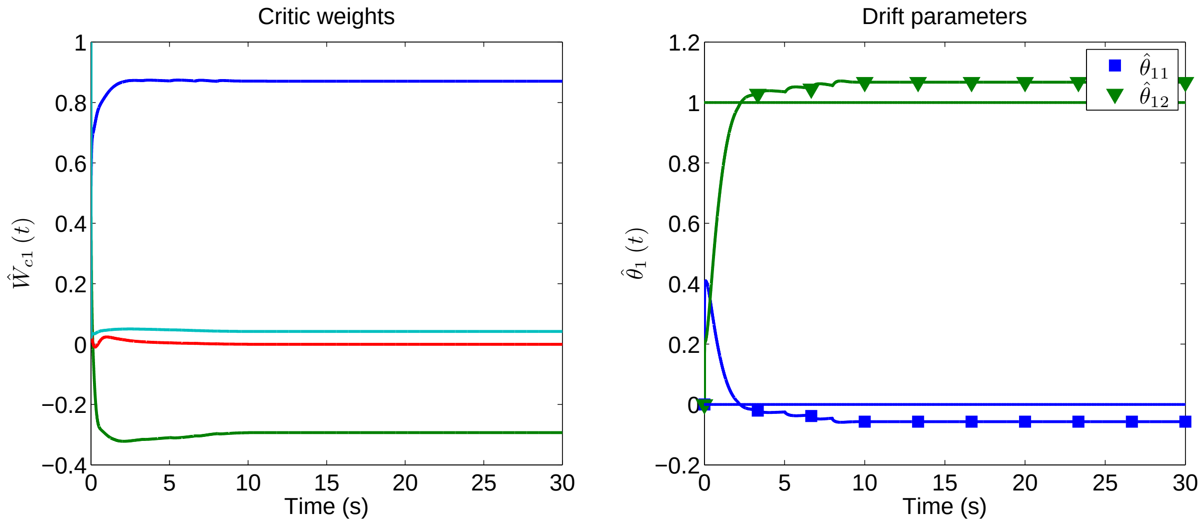

Figure 7-5. Value function weights and drift dynamics parameters estimates for agent 1 for the one-dimensional example. The dotted lines in the drift parameter plot are the ideal values of the drift parameters.

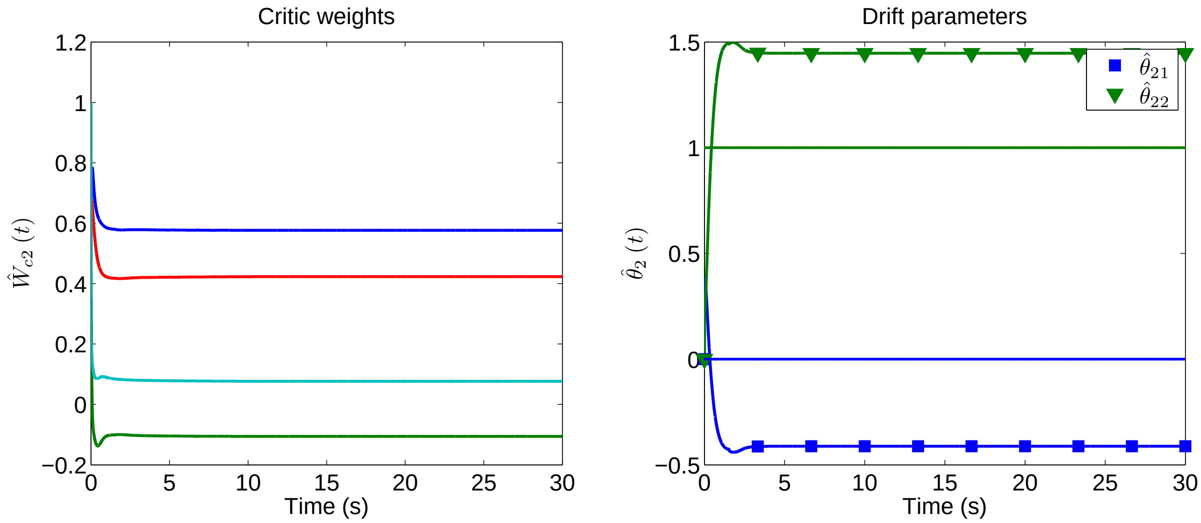

Figure 7-6. Value function weights and drift dynamics parameters estimates for agent 2 for the one-dimensional example. The dotted lines in the drift parameter plot are the ideal values of the drift parameters.

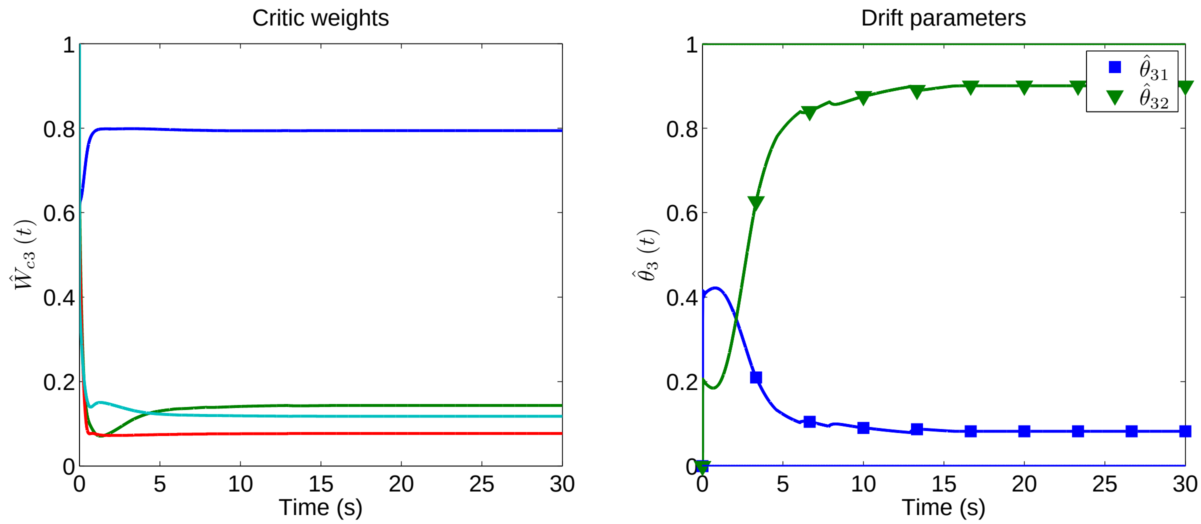

Figure 7-7. Value function weights and drift dynamics parameters estimates for agent 3 for the one-dimensional example. The dotted lines in the drift parameter plot are the ideal values of the drift parameters.

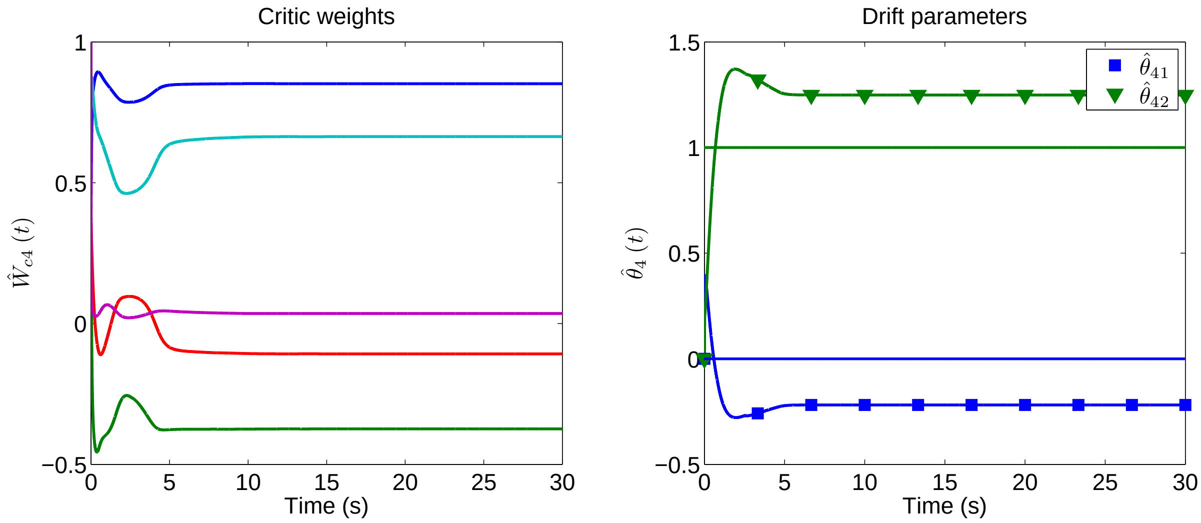

Figure 7-8. Value function weights and drift dynamics parameters estimates for agent 4 for the one-dimensional example. The dotted lines in the drift parameter plot are the ideal values of the drift parameters.

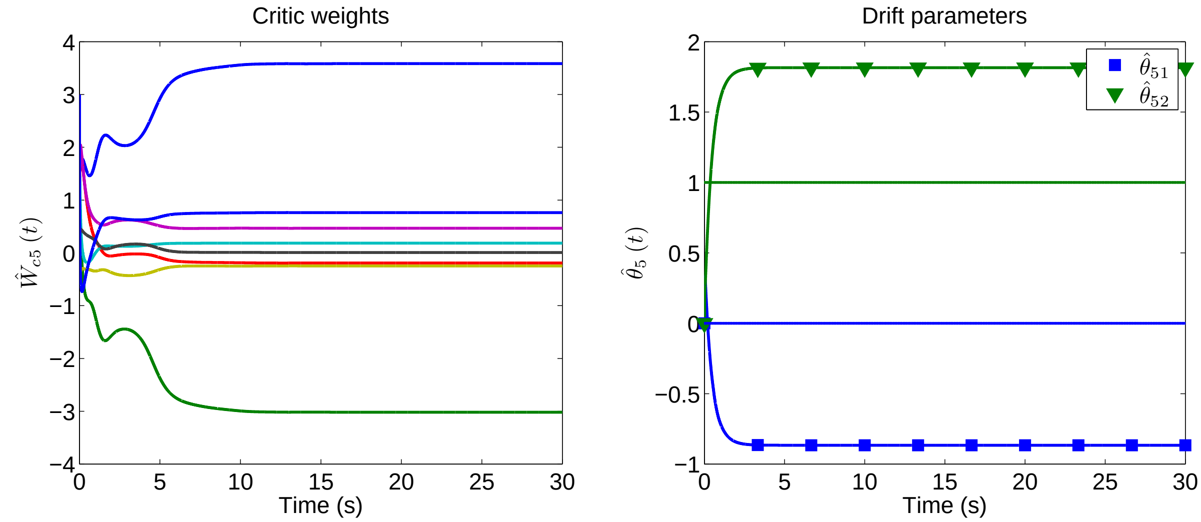

Figure 7-9. Value function weights and drift dynamics parameters estimates for agent 5 for the one-dimensional example. The dotted lines in the drift parameter plot are the ideal values of the drift parameters.

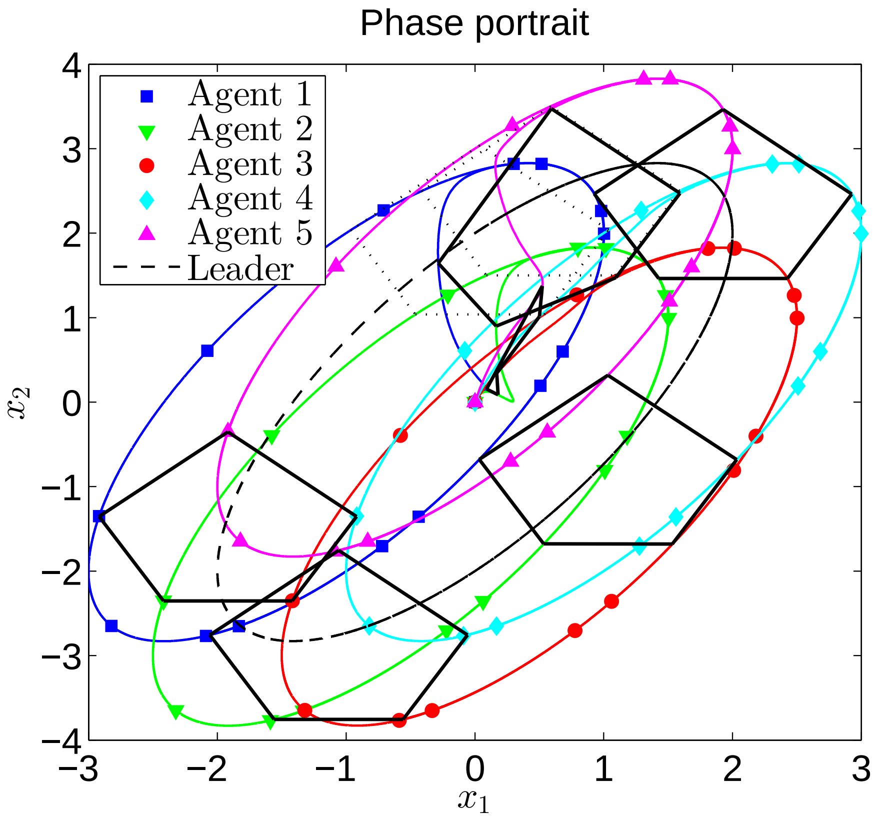

Figure 7-10. Phase portrait in the state-space for the two-dimensional example. The actual pentagonal formation is represented by a solid black pentagon, and the desired desired pentagonal formation around the leader is represented by a dotted black pentagon.

convergence of the value function weights and the unknown parameters in the drift dynamics.

7.8.2 Two-dimensional Example

In this simulation, the dynamics of all the agents are assumed to be exactly known, and are selected to be of the form (7–1) where for all $i = 1, \cdots, 5$,

$$f_i\left(x_i\right) = \begin{bmatrix} -x_{i1} + x_{i2} \\ -0.5x_{i1} - 0.5x_{i2}(1 - (\cos(2x_{i1}) + 2)^2) \end{bmatrix}, g_i\left(x_i\right) = \begin{bmatrix} \sin(2x_{i1}) + 2 & 0 \\ 0 & \cos(2x_{i1}) + 2 \end{bmatrix}.$$

The agents start at the origin, and their final desired relative positions are given by $xd_{12} = [-0.5,\ 1]^T$ $xd_{21} = [0.5,\ -1]^T$, $xd_{43} = [0.5,\ 1]^T$, and $xd_{53} = [-1,\ 1]^T$.

Table 7-2. Simulation parameters for the two-dimensional example

	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Q_i	$10I_2$	$10I_2$	$10I_2$	$10I_2$	$10I_2$
R_i	I_2	I_2	I_2	I_2	I_2
$\sigma_i\left(\mathcal{E}_i\right)$	$\frac{1}{2}[2e_{11}^2,\ 2e_{11}e_{12},\ 2e_{12}^2,\ e_{21}^2,\ 2e_{21}e_{22},\ e_{22}^2,\ e_{11}^2x_{11}^2,\ e_{12}^2x_{11}^2,\ e_{11}^2x_{12}^2,\ e_{12}^2x_{12}^2]^T$	$\frac{1}{2}[2e_{21}^2,\ 2e_{21}e_{22},\ 2e_{22}^2,\ e_{11}^2,\ 2e_{11}e_{12},\ e_{12}^2,\ e_{21}^2x_{21}^2,\ e_{22}^2x_{21}^2,\ e_{21}^2x_{22}^2,\ e_{22}^2x_{22}^2]^T$	$\frac{1}{2}[2e_{31}^2,\ 2e_{31}e_{32},\ 2e_{32}^2,\ e_{31}^2x_{31}^2,\ e_{32}^2x_{31}^2,\ e_{31}^2x_{32}^2,\ e_{32}^2x_{12}^2]^T$	$\frac{1}{2}[2e_{41}^2,\ 2e_{41}e_{42},\ 2e_{42}^2,\ e_{31}^2,\ 2e_{31}e_{32},\ e_{32}^2\ e_{41}^2x_{41}^2,\ e_{42}^2x_{41}^2,\ e_{41}^2x_{42}^2,\ e_{42}^2x_{42}^2]^T$	$\frac{1}{2}[2e_{51}^2,\ 2e_{51}e_{52},\ 2e_{52}^2,\ e_{41}^2,\ 2e_{41}e_{42},\ e_{42}^2,\ e_{31}^2,\ 2e_{31}e_{32},\ e_{32}^2,\ e_{51}^2x_{51}^2,\ e_{52}^2x_{51}^2,\ e_{51}^2x_{52}^2,\ e_{52}^2x_{52}^2]^T$
$x_i\left(0\right)$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$	$\mathbf{0}_{2\times1}$
$\hat{W}_{ci}\left(0\right)$	$\mathbf{1}_{10\times1}$	$\mathbf{1}_{10\times1}$	$2\times\mathbf{1}_{7\times1}$	$5\times\mathbf{1}_{10\times1}$	$3\times\mathbf{1}_{13\times1}$
$\hat{W}_{ai}\left(0\right)$	$\mathbf{1}_{10\times1}$	$\mathbf{1}_{10\times1}$	$2\times\mathbf{1}_{7\times1}$	$5\times\mathbf{1}_{10\times1}$	$3\times\mathbf{1}_{13\times1}$
$\Gamma_i\left(0\right)$	$500I_{10}$	$500I_{10}$	$500I_4$	$500I_5$	$500I_8$
η_{c1i}	0.1	0.1	0.1	0.1	0.1
η_{c2i}	2.5	5	2.5	2.5	2.5
η_{a1i}	2.5	0.5	2.5	2.5	2.5
η_{a2i}	0.01	0.01	0.01	0.01	0.01
ν_i	0.005	0.005	0.005	0.005	0.005

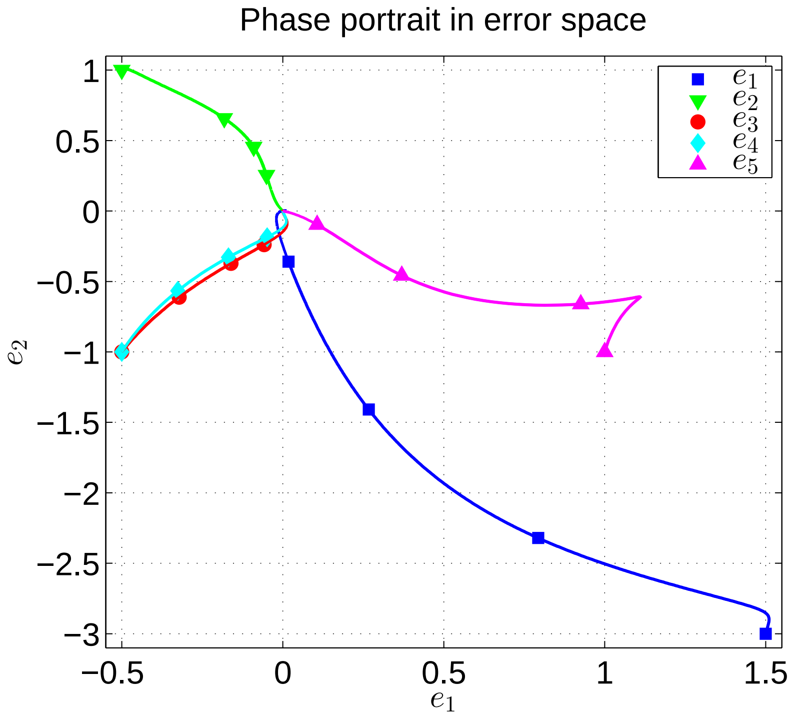

Figure 7-11. Phase portrait of all agents in the error space for the two-dimensional example.

The relative positions are designed such that the final desired formation is a pentagon with the leader node at the center.

The leader traverses a sinusoidal trajectory trajectory $x_0(t) = [2\sin(t),\ 2\sin(t) + 2\cos(t)]^T$. The desired positions of agents 1 and 3 with respect to the leader are $x_{d10} = [-1,\ 0]^T$ and $x_{d30} = [0.5,\ -1]^T$, respectively.

Table 7-2 summarizes the optimal control problem parameters, basis functions, and adaptation gains for the agents. For each agent i, nine values of e_i, x_i, and errors corresponding to all the extended neighbors are selected for BE extrapolation in uniform 3×3 grid in a 1×1 square around the origin, resulting in 9×9^{s_i} total values of $\mathcal{E}_i$.

Figures 7-10 - 7-16 show the tracking error, the state trajectories, and the control inputs for all the agents demonstrating convergence to the desired formation and the desired trajectory. Note that agents 2, 4, and 5 do not have a communication link

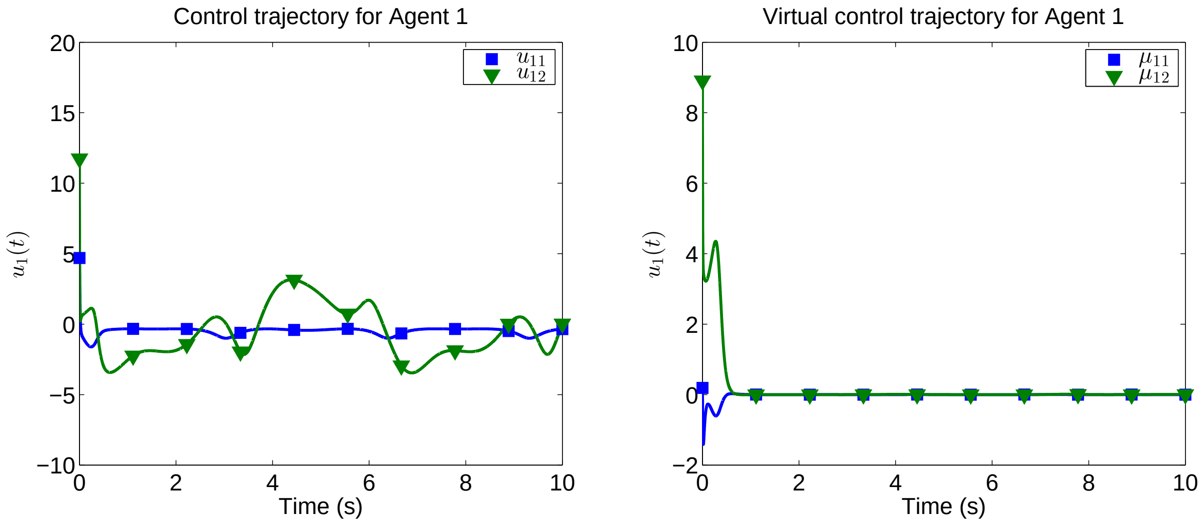

Figure 7-12. Trajectories of the control input and the relative control error for Agent 1 for the two-dimensional example.

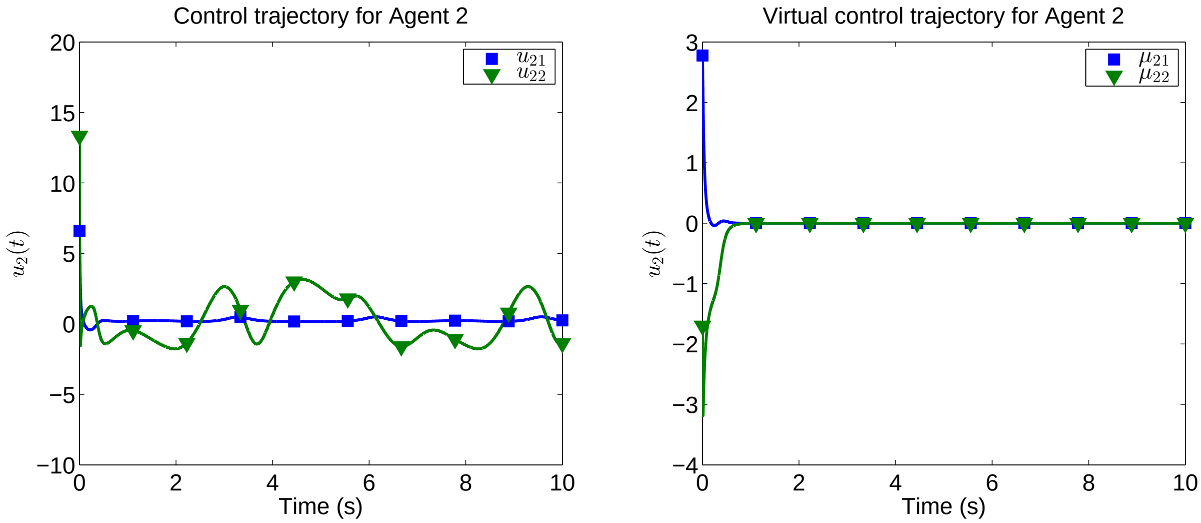

Figure 7-13. Trajectories of the control input and the relative control error for Agent 2 for the two-dimensional example.

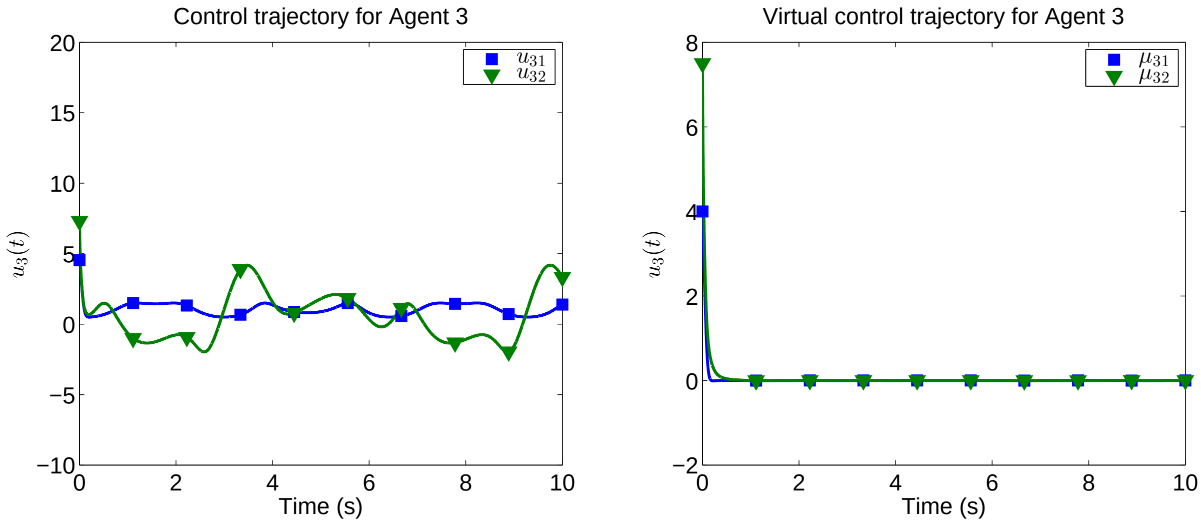

Figure 7-14. Trajectories of the control input and the relative control error for Agent 3 for the two-dimensional example.

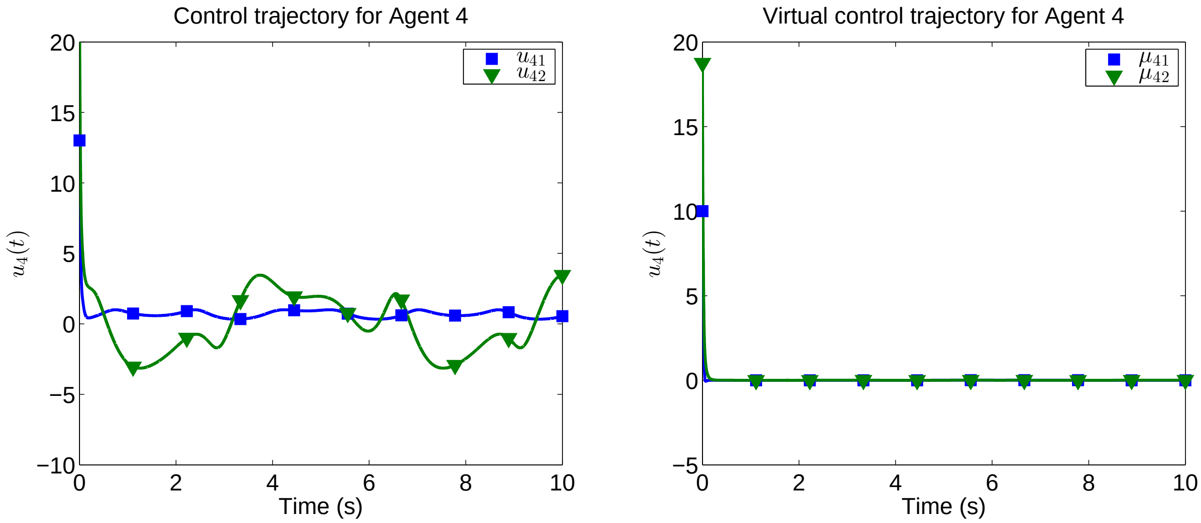

Figure 7-15. Trajectories of the control input and the relative control error for Agent 4 for the two-dimensional example.

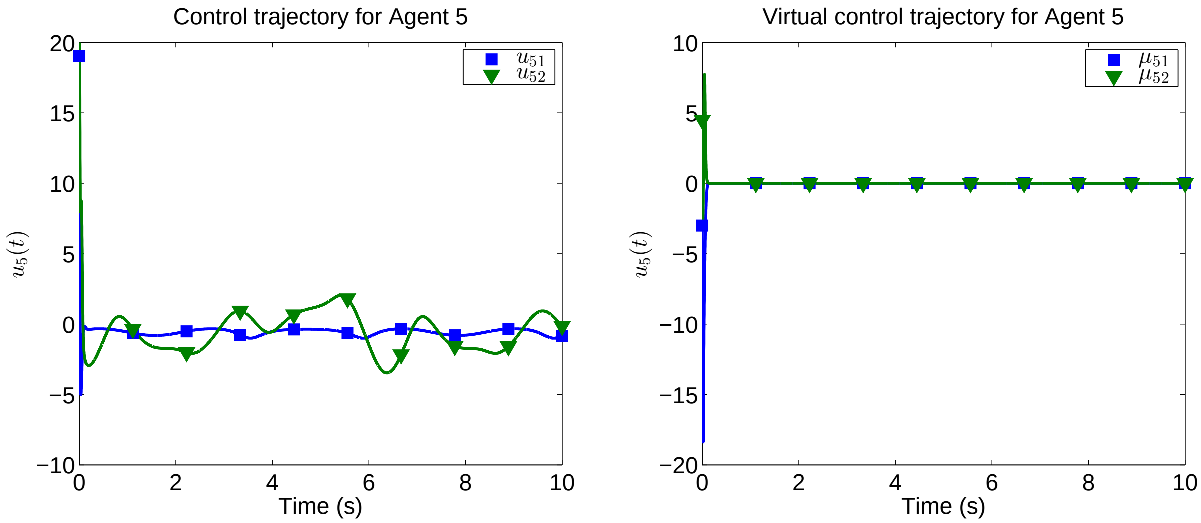

Figure 7-16. Trajectories of the control input and the relative control error for Agent 5 for the two-dimensional example.

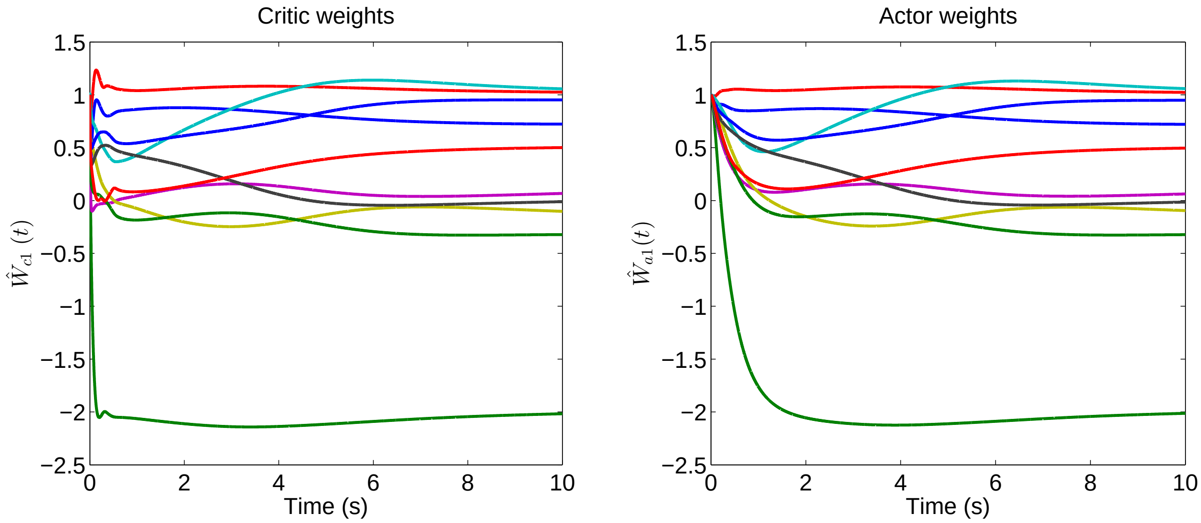

Figure 7-17. Value function weights and policy weights for agent 1 for the two-dimensional example.

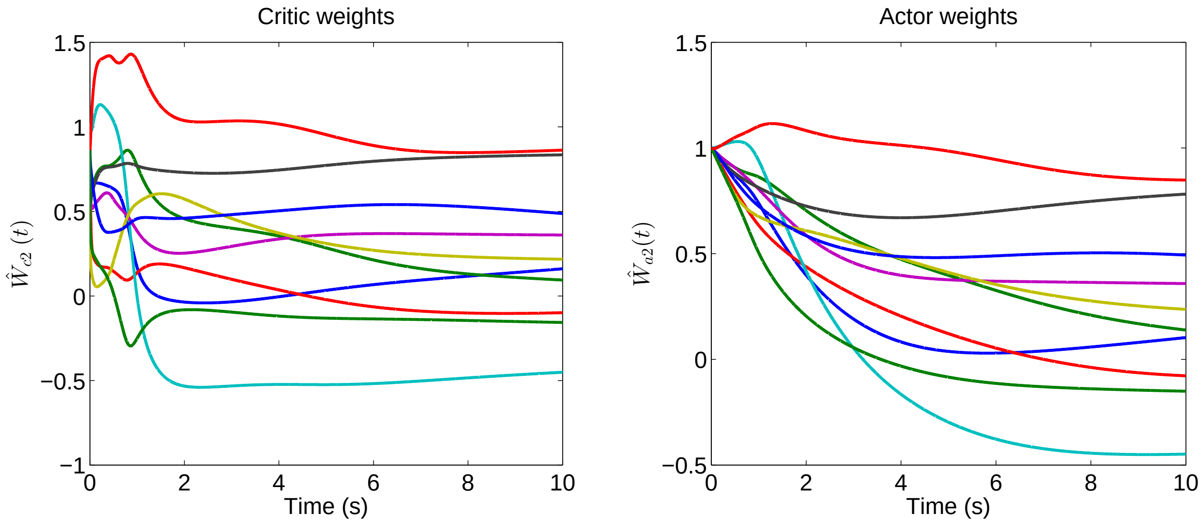

Figure 7-18. Value function weights and policy weights for agent 2 for the two-dimensional example.

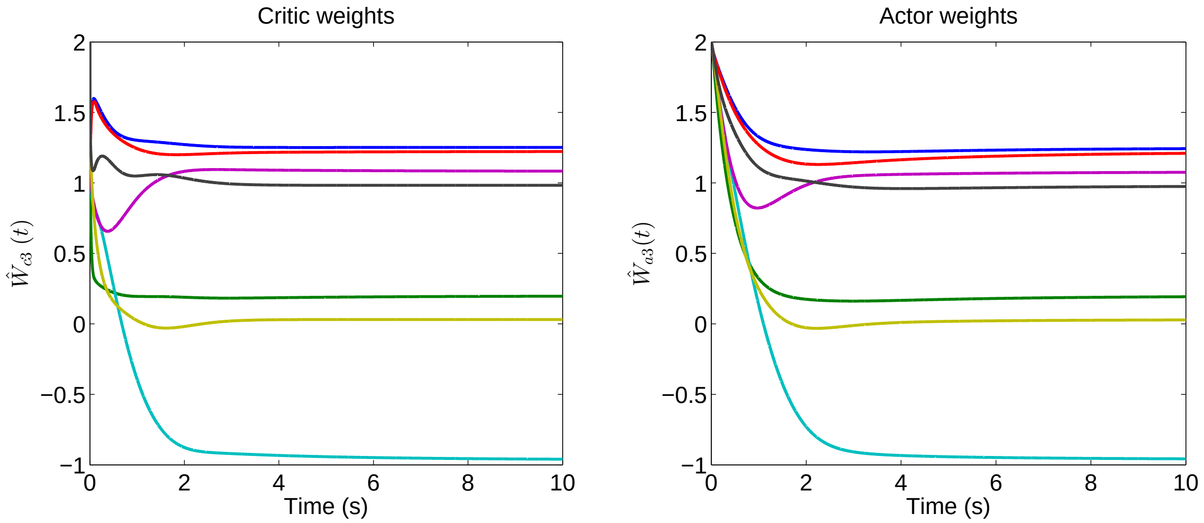

Figure 7-19. Value function weights and policy weights for agent 3 for the two-dimensional example.

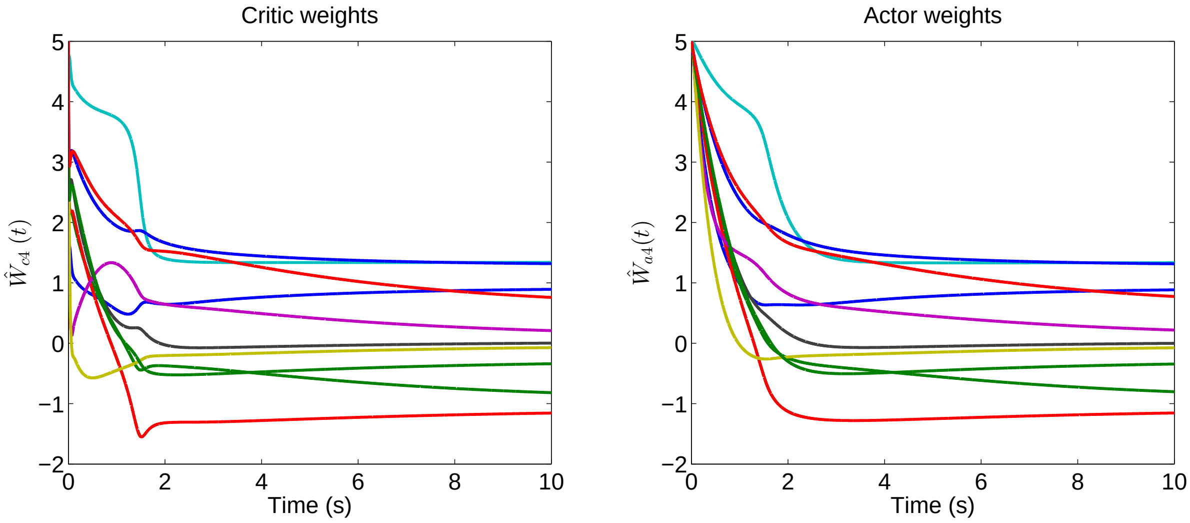

Figure 7-20. Value function weights and policy weights for agent 4 for the two-dimensional example.

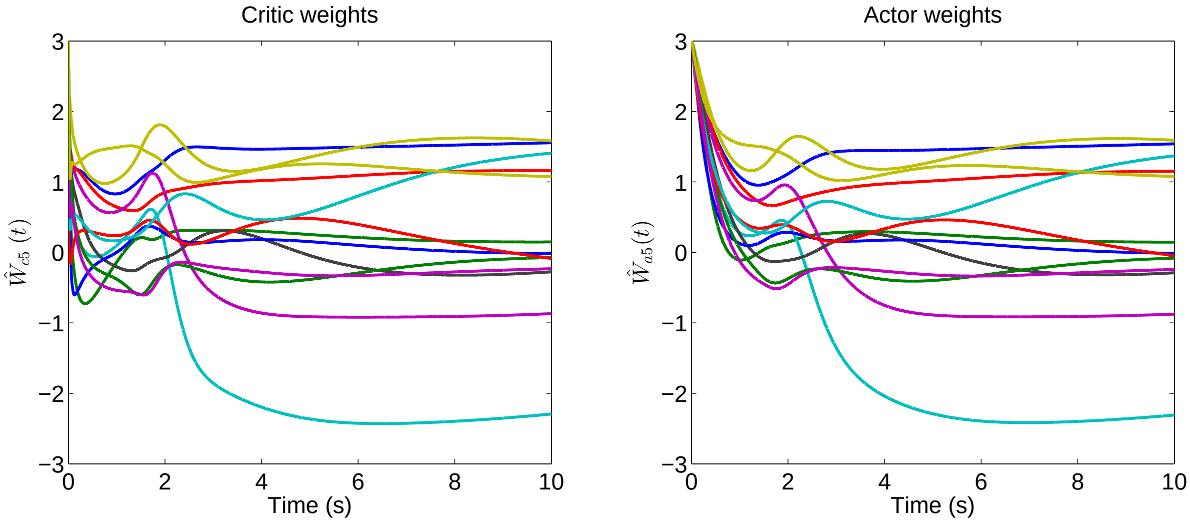

Figure 7-21. Value function weights and policy weights for agent 5 for the two-dimensional example.

to the leader, nor do they know their desired relative position from the leader. The convergence to the desired formation is achieved via cooperative control based on decentralized objectives. Figures 7-17 - 7-21 show the evolution and convergence of the value function weights and the policy weights for all the agents. Since an alternative method to solve this problem is not available to the best of the author's knowledge, a comparative simulation cannot be provided.

7.9 Concluding Remarks

A simulation-based ACI architecture is developed to cooperatively control a group of agents to track a trajectory while maintaining a desired formation. Communication among extended neighbors is needed to implement the developed method. Since an analytical feedback-Nash equilibrium solution is not available, the presented simulation does not demonstrate convergence to feedback-Nash equilibrium solutions. To the best of the author's knowledge, alternative methods to solve differential graphical game problems are not available in the literature; hence, a comparative simulation is infeasible.

CHAPTER 8
CONCLUSIONS

RL is a powerful tool for online learning and optimization, however, the application of RL to dynamical systems is challenging from a control theory perspective. The challenges take three different forms: analysis and design challenges, applicability challenges, and implementation challenges.

Since the controller is simultaneously learned and used online, unique analysis challenges arise in establishing stability during the learning phase. Furthermore, RL-based controllers are hard to design owing to the necessary tradeoffs between exploration and exploitation, which also complicate the stability analysis owing to the fact that in general, the learned controller does not meet the exploration demands, necessitating the addition of an exploration signal. In the case of deterministic nonlinear systems, an explicit characterization of the necessary exploration signals is hard to obtain; hence, the exploration signal is generally left out of the stability analysis, defeating the purpose of the stability analysis.

Applicability challenges spring from the fact that RL in continuous-state systems is usually realized using value function approximation. Since the action that a controller takes in a particular state depends on the value of that state, the control policy depends on the value function; hence, a uniform approximation of the value function over the entire operating domain is vital for the control design. Results that use parametric approximation techniques for value function approximation are ubiquitous in literature. Since parametric approximators can only generate uniform approximations over compact domains, approximation becomes challenging if the value function is time-varying and if the time horizon is infinite. Hence, traditional RL methods are not applicable for trajectory tracking applications, network control applications, and other applications that exhibit time-varying value functions.

The results of this dissertation partially address the aforementioned challenges via the development new innovative model-based RL methods and rigorous Lyapunov-based methods for stability analysis. In Chapter 3, a data-driven model-based RL technique that does not require an added exploration signal is developed to solve infinite-horizon total-cost optimal regulation problems for uncertain control-affine nonlinear systems. In Chapter 4, the data-driven model-based RL technique is extended to obtain feedback-Nash equilibrium solutions to $N-$player nonzero-sum differential games, without external ad-hoc application of an exploration signal. In chapters 3 and 4, sufficient exploration is simulated by using an estimate of the system dynamics obtained using a data-driven system identifier to extrapolate the BE to unexplored areas of the state-space. A set of points in the state space is selected a priori for BE extrapolation, and the value function is approximated using a time-varying regressor matrix computed based on the selected points. The developed result relies on a sufficient condition on the minimum eigenvalue of a time-varying regressor matrix. While this condition can be heuristically satisfied by choosing enough points, and can be easily verified online, it cannot, in general, be guaranteed a priori. Further research is required to investigate the existence of a set of points that guarantees that the resulting regressor matrix has a uniform a positive minimum singular value. The fact that the convergence rate of the value function approximation depends on the aforementioned minimum singular value motivates further research into a priori selection of and online adjustments to the set of points used for BE extrapolation. For example, threshold-based algorithms can be employed to ensure sufficient exploration by selecting new points if the minimum singular value of the regressor falls below a certain threshold.

In Chapter 5, RL-based methods are extended to a class of infinite-horizon optimal trajectory tracking problems where the value function is time-varying. Provided that the desired trajectory is the output of an autonomous dynamical system, the optimal control problem can be formulated so that the vale function depends on time only through the desired trajectory. Value function approximation is then achieved by using the desired trajectory along with the tracking error as training inputs. A Lyapunov-based stability analysis is developed based by proving that the time-varying value function is a Lyapunov function for the optimal closed-loop error system. The developed result relies on the assumption that a steady-state controller that can make the system exactly track the desired trajectory exists, and that it can be computed by inversion of the system dynamics. Inversion of system dynamics requires exact model knowledge. Motivated by the need to obtain an optimal tracking solution for uncertain systems, a data-driven system identifier is developed for approximate model inversion in Chapter 6. The data-driven system identifier is also used to extrapolate the BE, thereby removing the need for an added exploration signal from the tracking controller developed in Chapter 5. The developed technique requires knowledge of the dynamics of the desired trajectory. The fact that in many real world control applications, the desired trajectory is generated online using a trajectory planner module, motivates the development of an optimal tracking controller robust to uncertainties in the dynamics of the desired trajectory. Further research is required to apply RL-based methods to time-varying systems that cannot be transformed into stationary systems on compact domains using state augmentation. In adaptive control, it is generally possible to formulate the control problem such that PE along the desired trajectory is sufficient to achieve parameter convergence. In the ADP-based tracking problem, PE along the desired trajectory would be sufficient to achieve parameter convergence if the BE can be formulated in terms of the desired trajectories. Achieving such a formulation is not trivial, and is a subject for future research.

In Chapter 7, the RL-based methods are extended to obtain feedback-Nash equilibrium solutions to a class of differential graphical games using ideas from chapters *3* - *6.* It is established that in a cooperative game based on minimization of the local neighborhood tracking errors, the value function corresponding to the agents depends on information obtained from all their extended neighbors. A set of coupled HJ equations are developed that serve as necessary and sufficient conditions for feedback-Nash equilibrium, and closed-form expressions for the feedback-Nash equilibrium policies are developed based on the HJ equations. The fact that the developed technique requires each agent to communicate with all of its extended neighbors motivates the search for a decentralized method to generate feedback-Nash equilibrium policies.

In all the chapters of this dissertation, parametric approximation techniques are used to approximate the value functions. Parametric approximation of the value function requires selection of appropriate basis functions. Selection of basis functions for general nonlinear systems is a nontrivial open problem, even if the system dynamics are known. Implementation of RL-based controllers for general nonlinear systems is difficult because the basis functions and the exploration signal needs to be selected using trial-and-error, with very little insights to be gained from domain knowledge about the system. Note that a uniform approximation of the value function over the entire domain is required only if an optimal controller is desired. For real-time sub-optimal control, a good approximation of the value function over a small neighborhood of the current state is sufficient. This motivates the development of basis functions that follow the system state, and are capable of approximating the value function over a small domain. Analysis of convergence and stability issues arising from the use of moving basis functions is a subject for future research.

APPENDIX A
ONLINE DATA COLLECTION (CH 3)

The history stack $\mathcal{H}_{id}$ that satisfies conditions in (3–4) can be collected online provided the controller in (2–15) results in the system states being sufficiently exciting over a finite time interval $\left[t_0, t_0 + \overline{t}\right] \subset \mathbb{R}$.[1] During this finite time interval, since a history stack is not available, an adaptive update law that ensures fast convergence of $\tilde{\theta}$ to zero without PE cannot be developed. Hence, the system dynamics cannot be directly estimated without PE. Since extrapolation of the BE to unexplored areas of the state space requires estimates of the system dynamics, without PE, such extrapolation is infeasible during the time interval $\left[t_0, t_0 + \overline{t}\right]$.

However, evaluation of the BE along the system trajectories does not explicitly depend on the parameters θ. Estimation of the state derivative is enough to evaluate the BE along system trajectories. This motivates the development of the following state derivative estimator.

$$\begin{aligned} \dot{\hat{x}}_f &= gu + k_f \tilde{x}_f + \mu_f, \\ \dot{\mu}_f &= \left(k_f \alpha_f + 1\right) \tilde{x}_f, \end{aligned} \tag{A–1}$$

where $\hat{x}_f \in \mathbb{R}^n$ is an estimate of the state x, $\tilde{x}_f \triangleq x - \hat{x}_f$, and $k_f, \alpha_f, \gamma_f \in \mathbb{R}_{>0}$ are constant estimation gains. To facilitate the stability analysis, define a filtered error signal $r \in \mathbb{R}^n$ as $r \triangleq \dot{\tilde{x}}_f + \alpha_f \tilde{x}_f$, where $\dot{\tilde{x}}_f \triangleq \dot{x} - \dot{\hat{x}}_f$. Using (2–1) and (A–1) the dynamics of the filtered error signal can be expressed as $\dot{r} = -k_f r + \tilde{x}_f + \nabla_x f f + \nabla_x f g \hat{u} + \alpha \dot{\tilde{x}}_f$. The instantaneous BE in (2–12) can be approximated along the state trajectory using the

[1] To collect the history stack, the first M values of the state, the control, and the corresponding numerically computed state derivative are added to the history stack. Then, the existing values are progressively replaced with new values using a singular value maximization algorithm.

state derivative estimate as

$$\hat{\delta}_f = \omega_f^T \hat{W}_{cf} + x^T Q x + \hat{u}^T \left(x, \hat{W}_{af} \right) R \hat{u} \left(x, \hat{W}_{af} \right), \tag{A–2}$$

where $\omega_f \in \mathbb{R}^L$ is the regressor vector defined as $\omega_f \triangleq \nabla \sigma (x) \dot{\hat{x}}_f$. During the interval $\left[t_0, t_0 + \overline{t}\right]$, the value function and the policy weights can be learned based on the approximate BE in (A–2) provided the system states are exciting, i.e., if the following assumption is satisfied.

Assumption A.1. There exists a time interval $\left[t_0, t_0 + \overline{t}\right] \subset \mathbb{R}$ and positive constants $\underline{\psi}, T \in \mathbb{R}$ such that closed-loop trajectories of the system in (2–1) with the controller in (2–15) along with the weight update laws

$$\dot{\hat{W}}_{cf} = -\eta_{cf} \Gamma_f \frac{\omega_f}{\rho_f} \delta_f, \; \dot{\Gamma}_f = \lambda_f \Gamma_f - \eta_{cf} \Gamma_f \frac{\omega_f \omega_f^T}{\rho_f} \Gamma_f,$$

$$\dot{\hat{W}}_{af} = -\eta_{a1f} \left(\hat{W}_a - \hat{W}_c \right) - \eta_{a2f} \hat{W}_a, \tag{A–3}$$

where $\rho_f \triangleq 1 + \nu_f \omega_f^T \Gamma_f \omega_f$ is the normalization term, $\eta_{a1f}, \eta_{a2f}, \eta_{cf}, \nu_f \in \mathbb{R}$ are constant positive gains and $\Gamma_f \in \mathbb{R}^{L \times L}$ is the least-squares gain matrix, and the state derivative estimator in (A–1) satisfies

$$\underline{\psi} I_L \leq \int\limits_t^{t+T} \psi_f (\tau) \psi_f (\tau)^T d\tau, \; \forall t \in \left[t_0, t_0 + \overline{t}\right], \tag{A–4}$$

where $\psi_f \triangleq \frac{\omega_f}{\sqrt{1 + \nu_f \omega_f^T \Gamma_f \omega_f}} \in \mathbb{R}^N$ is the regressor vector. Furthermore, there exists a set of time instances $\{t_1 \cdots t_M\} \subset \left[t_0, t_0 + \overline{t}\right]$ such that the history stack $\mathcal{H}_{id}$ containing the values of state-action pairs and the corresponding numerical derivatives recorded at $\{t_1 \cdots t_M\}$ satisfies the conditions in Assumption (3.1).

Conditions similar to (A–4) are ubiquitous in online approximate optimal control literature. In fact, Assumption A.1 requires the regressor ψ_f to be exciting over a finite time interval, whereas the PE conditions used in related results such as [57–59, 114, 158] require similar regressor vectors to be exciting over all $t \in \mathbb{R}_{\geq t_0}$.

On any compact set $\chi \subset \mathbb{R}^n$ the function f is Lipschitz continuous; hence, there exist positive constants $L_f, L_{df} \in \mathbb{R}$ such that

$$\left\| f\left(x\right)\right\| \leq L_f \left\| x\right\| \text{ and } \left\| \nabla_x f\left(x\right)\right\| \leq L_{df},$$

for all $x \in \chi$. The update laws in (A–3) along with the excitation condition in (A–4) ensure that the adaptation gain matrix is bounded such that

$$\underline{\Gamma}_f \leq \left\| \Gamma_f\left(t\right)\right\| \leq \overline{\Gamma}_f, \; \forall t \in \mathbb{R}_{\geq t_0}, \tag{A–5}$$

where (cf. [91, Proof of Corollary 4.3.2])

$$\underline{\Gamma}_f = \min\left\{\eta_{cf}\underline{\psi}T, \lambda_{\min}\left(\Gamma_f\left(t_0\right)\right)\right\} e^{-\lambda_f T}.$$

The following positive constants are defined for brevity of notation.

$$\vartheta_8 \triangleq \frac{L_{df}}{2}\overline{\left\| gR^{-1}g^T\nabla\sigma^T\right\|}, \quad \vartheta_9 \triangleq \frac{\overline{\left\| W^T G_\sigma + \frac{1}{2}\nabla\epsilon G^T \nabla\sigma^T\right\|}}{2} + \eta_{a2f}\overline{W},$$

$$\vartheta_{10} \triangleq \frac{\overline{\left\| 2W^T\nabla\sigma G\nabla\epsilon^T + G_\epsilon\right\|}}{4}, \quad \iota_f \triangleq 2\eta_{cf}\vartheta_{10} + \frac{3\vartheta_9}{4\left(\eta_{a1f}+\eta_{a2f}\right)} + \vartheta_4 + \frac{5\vartheta_8^2\overline{W}^2}{4k_f},$$

$$v_{lf} = \frac{1}{2}\min\left(\frac{\underline{q}}{2}, \frac{\beta\underline{\Gamma}_f}{4}, \frac{\left(\eta_{a1f}+\eta_{a2f}\right)}{3}, \frac{\alpha_f}{3}, \frac{k_f}{5}\right).$$

To facilitate the stability analysis, Let $V_{Lf} : \mathbb{R}^{3n+2L} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a continuously differentiable positive definite candidate Lyapunov function defined as

$$V_{Lf}\left(Z_f, t\right) \triangleq V^*\left(x\right) + \frac{1}{2}\tilde{W}_{cf}^T\Gamma_f^{-1}\left(t\right)\tilde{W}_{cf} + \frac{1}{2}\tilde{W}_{af}^T\tilde{W}_{af} + \frac{1}{2}\tilde{x}_f^T\tilde{x}_f + \frac{1}{2}r^T r. \tag{A–6}$$

Using the fact that V^* is positive definite, (A–5) and Lemma 4.3 from [149] yield

$$\underline{v_{lf}}\left(\left\| Z_f\right\|\right) \leq V_{Lf}\left(Z_f, t\right) \leq \overline{v_{lf}}\left(\left\| Z_f\right\|\right), \tag{A–7}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Z_f \in \mathbb{R}^{3n+2L}$. In (A–7), $\underline{\upsilon_{lf}}, \overline{\upsilon_{lf}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions and $Z \triangleq \left[x^T,\, \tilde{W}_{cf}^T,\, \tilde{W}_{af}^T,\, \tilde{x}_f^T,\, r^T\right]^T$. The sufficient conditions for UUB convergence are derived based on the subsequent stability analysis as

$$(\eta_{a1f} + \eta_{a2f}) > \frac{3\eta_{a1f}}{2\zeta_4} - \frac{3\vartheta_8\zeta_5}{2} - \frac{3\eta_{cf}\overline{\|G_\sigma\|}}{4\sqrt{\nu_f\Gamma_f}}\overline{Z}_f, \quad \underline{q} > 2L_f^2\left(2\eta_{cf}\overline{\nabla\epsilon}^2 + \frac{5L_{df}^2}{4k_f}\right)$$

$$k_f > 5\max\left(\frac{\vartheta_8}{2\zeta_5} + \alpha_f + 2\eta_{cf}\overline{W}^2\overline{\|\nabla\sigma\|}^2, \frac{3\alpha_f^3}{4}\right), \quad \frac{1}{\alpha_f} > 6\eta_{cf}\overline{W}^2\overline{\|\nabla\sigma\|}^2, \quad \beta\underline{\Gamma}_f > 2\eta_{a1f}\zeta_4, \tag{A–8}$$

where $\overline{Z}_f \triangleq \underline{\upsilon}_f^{-1}\left(\overline{\upsilon}_f\left(\max\left(\|Z_f(t_0)\|, \sqrt{\frac{\iota_f}{\upsilon_{lf}}}\right)\right)\right)$, and $\zeta_4, \zeta_5 \in \mathbb{R}$ are known positive adjustable constants. An algorithm similar to Algorithm 3.1 is employed to select the gains and a compact set $\mathcal{Z}_f \subset \mathbb{R}^{3n+2L}$ such that

$$\sqrt{\frac{\iota_f}{\upsilon_{lf}}} \leq \frac{1}{2}\text{diam}(\mathcal{Z}_f). \tag{A–9}$$

Theorem A.1. *Provided the gains are selected to satisfy the sufficient conditions in* (A–8) *based on an algorithm similar to Algorithm 3.1, the controller in* (2–15)*, the weight update laws in* (A–3)*, the state derivative estimator in* (A–1)*, and the excitation condition in* (A–4) *ensure that the state trajectory x, the state estimation error $\tilde{x}_f$, and the parameter estimation errors $\tilde{W}_{cf}$, and $\tilde{W}_{af}$ remain bounded such that*

$$\|Z_f(t)\| \leq \overline{Z}_f,\ \forall t \in \left[t_0, t_0 + \overline{t}\right].$$

Proof. Using techniques similar to the proof of Theorem 3.1, the time derivative of the candidate Lyapunov function in (A–6) can be bounded as

$$\dot{V}_{Lf} \leq -\upsilon_{lf}\|Z_f\|^2,\ \forall\|Z_f\| \geq \sqrt{\frac{\iota_f}{\upsilon_{lf}}}, \tag{A–10}$$

in the domain $\mathcal{Z}_f$. Using (A–7), (A–9), and (A–10), Theorem 4.18 in [149] is used to show that Z_f is UUB, and that $\|Z_f(t)\| \leq \overline{Z}_f,\ \forall t \in \left[t_0, t_0 + \overline{t}\right]$. □

During the interval $\left[t_0, t_0 + \overline{t}\right]$, the controller in (2–15) is used along with the weight update laws in Assumption A.1. When enough data is collected in the history stack to satisfy the rank condition in (3–4), the update laws from Section (3.3) are used. The bound $\overline{Z}_f$ is used to compute gains for Theorem 3.1 using Algorithm 3.1. Theorems 1 and 2 establish UUB regulation of the system state and the parameter estimation errors for the overall switched system.

APPENDIX B
PROOF OF SUPPORTING LEMMAS (CH 5)

B.1 Proof of Lemma 5.1

The following supporting technical lemma is used to prove Lemma 5.1.

Lemma B.1. *Let $D \subseteq \mathbb{R}^n$ contain the origin and let $\Xi : D \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be positive definite. If $\Xi(x,t)$ is bounded, uniformly in t, for all bounded x and if $x \longmapsto \Xi(x,t)$ is continuous, uniformly in t, then Ξ is decrescent in D.*

Proof. Since $\Xi(x,t)$ is bounded, uniformly in t, $\sup_{t\in\mathbb{R}_{\geq 0}} \{\Xi(x,t)\}$ exists and is unique for all bounded x. Let the function $\alpha : D \to \mathbb{R}_{\geq 0}$ be defined as

$$\alpha(x) \triangleq \sup_{t\in\mathbb{R}_{\geq 0}} \{\Xi(x,t)\}. \tag{B–1}$$

Since $x \to \Xi(x,t)$ is continuous, uniformly in t, $\forall\varepsilon > 0$, $\exists\varsigma(x) > 0$ such that $\forall y \in D$,

$$d_{D\times\mathbb{R}_{\geq 0}}((x,t),(y,t)) < \varsigma(x) \implies d_{\mathbb{R}_{\geq 0}}(\Xi(x,t),\Xi(y,t)) < \varepsilon, \tag{B–2}$$

where $d_M(\cdot,\cdot)$ denotes the standard Euclidean metric on the metric space M. By the definition of $d_M(\cdot,\cdot)$, $d_{D\times\mathbb{R}_{\geq 0}}((x,t),(y,t)) = d_D(x,y)$. Using (B–2),

$$d_D(x,y) < \varsigma(x) \implies |\Xi(x,t) - \Xi(y,t)| < \varepsilon. \tag{B–3}$$

Given the fact that Ξ is positive, (B–3) implies $\Xi(x,t) < \Xi(y,t) + \varepsilon$ and $\Xi(y,t) < \Xi(x,t) + \varepsilon$ which from (B–1) implies $\alpha(x) < \alpha(y) + \varepsilon$ and $\alpha(y) < \alpha(x) + \varepsilon$, and hence, from (B–3), $d_D(x,y) < \varsigma(x) \implies |\alpha(x) - \alpha(y)| < \varepsilon$. Since Ξ is positive definite, (B–1) can be used to conclude $\alpha(0) = 0$. Thus, Ξ is bounded above by a continuous positive definite function; hence, Ξ is decrescent in D. □

Based on the definitions in (5–3)-(5–6) and (5–21),

$$V_t^*(e,t) = \int\limits_t^\infty \left(e^T(\rho)\,Qe(\rho) + \mu^{*T}(\rho)\,R\mu^*(\rho)\right)d\rho \quad \geq V_e(e), \forall t \in \mathbb{R}_{\geq 0}, \tag{B–4}$$

where $V_e\left(e\right) \triangleq \int_t^\infty \left(e^T\left(\rho\right)Qe\left(\rho\right)\right)d\rho$ is a positive definite function. Lemma 4.3 in [149] can be invoked to conclude that there exists a class $\mathcal{K}$ function $\underline{v} : [0,a] \to \mathbb{R}_{\geq 0}$ such that $\underline{v}\left(\|e\|\right) \leq V_e\left(e\right)$, which along with (B–4), implies (5–22a).

From (B–4), $V^*\left(\left[0,x_d^T\right]^T\right) = \int_t^\infty \left(\mu^{*T}\left(\rho\right)R\mu^*\left(\rho\right)\right)d\rho$, with the minimizer $\mu^*\left(t\right) = 0, \forall t \in \mathbb{R}_{\geq 0}$. Furthermore, $V^*\left(\left[0,x_d^T\right]^T\right)$ is the cost incurred when starting with $e = 0$ and following the optimal policy thereafter for any arbitrary desired trajectory x_d (cf. Section 3.7 of [43]). Substituting $x\left(t_0\right) = x_d\left(t_0\right)$, $\mu\left(t_0\right) = 0$ and (5–2) in (5–4) indicates that $\dot{e}\left(t_0\right) = 0$. Thus, when starting from $e = 0$, the zero policy satisfies the dynamic constraints in (5–4). Furthermore, the optimal cost is $V^*\left(\left[0,x_d^T\right]^T\right) = 0,\ \forall \left\|x_d\right\| < d$ which, from (5–21), implies (5–22b).

Admissibility of the optimal policy implies that $V^*\left(\zeta\right)$ is bounded for all bounded ζ. Since the desired trajectory is bounded, $V_t^*\left(e,t\right)$ is bounded, uniformly in t, for all bounded e. To establish that $e \longmapsto V_t^*\left(e,t\right)$ is continuous, uniformly in t, let $\chi_{e_o} \subset \mathbb{R}^n$ be a compact set containing e_o. Since x_d is bounded, $x_d \in \chi_{x_d}$, where $\chi_{x_d} \subset \mathbb{R}^n$ is compact. Since $V^* : \mathbb{R}^{2n} \to \mathbb{R}_{\geq 0}$ is continuous, and $\chi_{e_o} \times \chi_{x_d} \subset \mathbb{R}^{2n}$ is compact, V^* is uniformly continuous on $\chi_{e_o} \times \chi_{x_d}$. Thus, $\forall \varepsilon > 0$, $\exists \varsigma > 0$, such that $\forall \left[e_o^T, x_d^T\right]^T, \left[e_1^T, x_d^T\right]^T \in \chi_{e_o} \times \chi_{x_d}$, $d_{\chi_{e_o} \times \chi_{x_d}}\left(\left[e_o^T, x_d^T\right]^T, \left[e_1^T, x_d^T\right]^T\right) < \varsigma \implies d_{\mathbb{R}}\left(V^*\left(\left[e_o^T, x_d^T\right]^T\right), V^*\left(\left[e_1^T, x_d^T\right]^T\right)\right) < \varepsilon$. Thus, for each $e_o \in \mathbb{R}^n$, there exists a $\varsigma > 0$ independent of x_d, that establishes the continuity of $e \longmapsto V^*\left(\left[e^T, x_d^T\right]^T\right)$ at e_o. Thus, $e \longmapsto V^*\left(\left[e^T, x_d^T\right]^T\right)$ is continuous, uniformly in x_d, and hence, using (5–21) $e \longmapsto V_t^*\left(e,t\right)$ is continuous, uniformly in t. Using Lemma B.1 and (5–22a) and (5–22b), there exists a positive definite function $\alpha : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that $V_t^*\left(e,t\right) < \alpha\left(e\right), \ \forall \left(e,t\right) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0}$. Lemma 4.3 in [149] indicates that there exists a class $\mathcal{K}$ function $\overline{v} : [0,a] \to \mathbb{R}_{\geq 0}$ such that $\alpha\left(e\right) \leq \overline{v}\left(\|e\|\right)$, which implies (5–22c).

B.2 Proof of Lemma 5.2

Using the definition of the controller in (14), the tracking error dynamics can be expressed as

$$\dot{e} = f + \frac{1}{2} g R^{-1} G^T \sigma'^T \tilde{W}_a + g g_d^+ \left(h_d - f_d\right) - \frac{1}{2} g R^{-1} G^T \sigma'^T W - h_d.$$

On any compact set, the tracking error derivative can be bounded above as

$$\left\| \dot{e} \right\| \leq L_F \left\| e \right\| + L_W \left\| \tilde{W}_a \right\| + L_e,$$

where $L_e = L_F \left\| x_d \right\| + \left\| g g_d^+ \left(h_d - f_d\right) - \frac{1}{2} g R^{-1} G^T \sigma'^T W - h_d \right\|$ and $L_W = \frac{1}{2} \left\| g R^{-1} G^T \sigma'^T \right\|$. Using the fact that e and $\tilde{W}_a$ are continuous functions of time, on the interval $[t, t+T]$, the time derivative of e can be bounded as

$$\left\| \dot{e} \right\| \leq L_F \sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\| + L_W \sup_{\tau \in [t, t+T]} \left\| \tilde{W}_a\left(\tau\right) \right\| + L_e.$$

Since the infinity norm is less than the 2-norm, the derivative of the j^{th} component of $\dot{e}$ is bounded as

$$\dot{e}_j \leq L_F \sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\| + L_W \sup_{\tau \in [t, t+T]} \left\| \tilde{W}_a\left(\tau\right) \right\| + L_e.$$

Thus, the maximum and the minimum value of e_j are related as

$$\sup_{\tau \in [t, t+T]} \left| e_j\left(\tau\right) \right| \leq \inf_{\tau \in [t, t+T]} \left| e_j\left(\tau\right) \right| + \left(L_F \sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\| + L_W \sup_{\tau \in [t, t+T]} \left\| \tilde{W}_a\left(\tau\right) \right\| + L_e \right) T.$$

Squaring the above expression and using the inequality $\left(x + y\right)^2 \leq 2x^2 + 2y^2$

$$\sup_{\tau \in [t, t+T]} \left| e_j\left(\tau\right) \right|^2 \leq 2 \inf_{\tau \in [t, t+T]} \left| e_j\left(\tau\right) \right|^2 + 2 \left(L_F \sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\| + L_W \sup_{\tau \in [t, t+T]} \left\| \tilde{W}_a\left(\tau\right) \right\| + L_e \right)^2 T^2.$$

Summing over j, and using the the facts that $\sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\|^2 \leq \sum_{j=1}^{n} \sup_{\tau \in [t, t+T]} \left| e_j\left(\tau\right) \right|^2$ and $\inf_{\tau \in [t, t+T]} \sum_{j=1}^{n} \left| e_j\left(\tau\right) \right|^2 \leq \inf_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\|^2$,

$$\sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\|^2 \leq 2 \inf_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\|^2 + 2 \left(L_F \sup_{\tau \in [t, t+T]} \left\| e\left(\tau\right) \right\|^2 + L_W \sup_{\tau \in [t, t+T]} \left\| \tilde{W}_a\left(\tau\right) \right\|^2 + L_e \right)^2 n T^2.$$

Using the inequality $(x+y+z)^2 \leq 3x^2+3y^2+3z^2$, (5–23) can be obtained. Using a similar procedure on the update law for $\tilde{W}_a$,

$$-\inf_{\tau\in[t,t+T]}\left\|\tilde{W}_a(\tau)\right\|^2 \leq -\frac{\left(1-6N\left(\eta_{a1}+\eta_{a2}\right)^2T^2\right)}{2}\sup_{\tau\in[t,t+T]}\left\|\tilde{W}_a(\tau)\right\|^2 + 3N\eta_{a2}^2W^2T^2$$
$$+3N\eta_{a1}^2\sup_{\tau\in[t,t+T]}\left\|\tilde{W}_c(\tau)\right\|^2T^2. \quad \text{(B–5)}$$

Similarly, the dynamics for $\tilde{W}_c$ yield

$$\sup_{\tau\in[t,t+T]}\left\|\tilde{W}_c(\tau)\right\|^2 \leq \frac{2}{\left(1-\frac{6N\eta_c^2\overline{\varphi}^2T^2}{\nu^2\underline{\varphi}^2}\right)}\inf_{\tau\in[t,t+T]}\left\|\tilde{W}_c(\tau)\right\|^2 + \frac{6NT^2\eta_c^2\overline{\varphi}^2\left(\bar{\epsilon'}L_Fd+\iota_5\right)^2}{\nu\underline{\varphi}\left(1-\frac{6N\eta_c^2\overline{\varphi}^2T^2}{\nu^2\underline{\varphi}^2}\right)}$$
$$+\frac{6NT^2\eta_c^2\overline{\varphi}^2\bar{\epsilon'}^2L_F^2}{\nu\underline{\varphi}\left(1-\frac{6N\eta_c^2\overline{\varphi}^2T^2}{\nu^2\underline{\varphi}^2}\right)}\sup_{\tau\in[t,t+T]}\left\|e(\tau)\right\|^2. \quad \text{(B–6)}$$

Substituting (B–6) into (B–5), (5–24) can be obtained.

B.3 Proof of Lemma 5.3

The integrand on the LHS can be written as

$$\tilde{W}_c^T(\tau)\psi(\tau) = \tilde{W}_c^T(t)\psi(\tau) + \left(\tilde{W}_c^T(\tau)-\tilde{W}_c^T(t)\right)\psi(\tau).$$

Using the inequality $(x+y)^2 \geq \frac{1}{2}x^2 - y^2$ and integrating,

$$\int_t^{t+T}\left(\tilde{W}_c^T(\tau)\psi(\tau)\right)^2d\tau \geq \frac{1}{2}\tilde{W}_c^T(t)\left(\int_t^{t+T}\left(\psi(\tau)\psi(\tau)^T\right)d\tau\right)\tilde{W}_c(t)$$
$$-\int_t^{t+T}\left(\left(\int_t^{\tau}\dot{\tilde{W}}_c(\sigma)\,d\tau\right)^T\psi(\tau)\right)^2d\tau.$$

Substituting the dynamics for $\tilde{W}_c$ from (20) and using the PE condition in Assumption 3,

$$\int_t^{t+T}\left(\tilde{W}_c^T(\tau)\psi(\tau)\right)^2d\tau \geq \frac{1}{2}\underline{\psi}\tilde{W}_c^T(t)\tilde{W}_c(t) - \int_t^{t+T}\Bigg(\Bigg(\Bigg(\int_t^{\tau}\Bigg(-\eta_c\Gamma(\sigma)\psi(\sigma)\psi^T(\sigma)\tilde{W}_c(\sigma)$$
$$+\frac{\eta_c\Gamma(\sigma)\psi(\sigma)\Delta(\sigma)}{\sqrt{1+\nu\omega(\sigma)^T\Gamma(\sigma)\omega(\sigma)}} + \frac{\eta_c\Gamma(\sigma)\psi(\sigma)\tilde{W}_a^T\mathcal{G}_\sigma\tilde{W}_a}{4\sqrt{1+\nu\omega(\sigma)^T\Gamma(\sigma)\omega(\sigma)}}$$

$$- \frac{\eta_c \Gamma\left(\sigma\right) \psi\left(\sigma\right) \epsilon^{'}\left(\sigma\right) F\left(\sigma\right)}{\sqrt{1 + \nu\omega\left(\sigma\right)^T \Gamma\left(\sigma\right)\omega\left(\sigma\right)}}\Bigg) d\sigma\Bigg)^T \psi\left(\tau\right)\Bigg)^2,$$

where $\Delta \triangleq \frac{1}{4}\epsilon' \mathcal{G} \epsilon'^T + \frac{1}{2} W^T \sigma' \mathcal{G} \epsilon'^T$. Using the inequality $\left(x + y + w - z\right)^2 \leq 2x^2 + 6y^2 + 6w^2 + 6z^2$,

$$\begin{aligned}
\int\limits_t^{t+T} \left(\tilde{W}_c^T\left(\tau\right)\psi\left(\tau\right)\right)^2 d\tau \geq \frac{1}{2}\underline{\psi}\tilde{W}_c^T\left(t\right)\tilde{W}_c\left(t\right) \\
- \int\limits_t^{t+T} 2\left(\int\limits_t^{\tau} \eta_c \tilde{W}_c^T\left(\sigma\right)\psi\left(\sigma\right)\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right) d\sigma\right)^2 d\tau \\
- 6\int\limits_t^{t+T}\left(\int\limits_t^{\tau} \frac{\eta_c \Delta^T\left(\sigma\right)\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)}{\sqrt{1 + \nu\omega\left(\sigma\right)^T\Gamma\left(\sigma\right)\omega\left(\sigma\right)}} d\sigma\right)^2 d\tau \\
- 6\int\limits_t^{t+T}\left(\int\limits_t^{\tau} \frac{\eta_c F^T\left(\sigma\right)\epsilon^{'T}\left(\sigma\right)\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)}{\sqrt{1 + \nu\omega\left(\sigma\right)^T\Gamma\left(\sigma\right)\omega\left(\sigma\right)}} d\sigma\right)^2 d\tau \\
- 6\int\limits_t^{t+T}\left(\int\limits_t^{\tau} \frac{\eta_c \tilde{W}_a^T\left(\sigma\right)\mathcal{G}_\sigma\left(\sigma\right)\tilde{W}_a\left(\sigma\right)\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)}{\sqrt{1 + \nu\omega\left(\sigma\right)^T\Gamma\left(\sigma\right)\omega\left(\sigma\right)}} d\sigma\right)^2 d\tau.
\end{aligned}$$

Using the Cauchy-Schwarz inequality, the Lipschitz property, the fact that $\frac{1}{\sqrt{1+\nu\omega^T\Gamma\omega}} \leq 1$, and the bounds in (23),

$$\begin{aligned}
\int\limits_t^{t+T} \left(\tilde{W}_c^T\left(\tau\right)\psi\left(\tau\right)\right)^2 d\tau \geq \frac{1}{2}\underline{\psi}\tilde{W}_c^T\left(t\right)\tilde{W}_c\left(t\right) - 6\int\limits_t^{t+T}\left(\int\limits_t^{\tau} \frac{\eta_c \iota_5 \overline{\varphi}}{\nu \underline{\varphi}} d\sigma\right)^2 d\tau \\
- \int\limits_t^{t+T} 2\eta_c^2 \left(\int\limits_t^{\tau} \left(\tilde{W}_c^T\left(\sigma\right)\psi\left(\sigma\right)\right)^2 d\sigma \int\limits_t^{\tau} \left(\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)\right)^2 d\sigma\right) d\tau \\
- \int\limits_t^{t+T} 6\eta_c^2\iota_2^2 \left(\int\limits_t^{\tau} \left\|\tilde{W}_a\left(\sigma\right)\right\|^4 d\sigma \int\limits_t^{\tau} \left(\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)\right)^2 d\sigma\right) d\tau \\
- \int\limits_t^{t+T} 6\eta_c^2\bar{\epsilon}'^2 \left(\int\limits_t^{\tau} \left\|F\left(\sigma\right)\right\|^2 d\sigma \int\limits_t^{\tau} \left(\psi^T\left(\sigma\right)\Gamma^T\left(\sigma\right)\psi\left(\tau\right)\right)^2 d\sigma\right) d\tau
\end{aligned}$$

Thus,

$$
\begin{aligned}
\int\limits_{t}^{t+T} \left(\tilde{W}_c^T\left(\tau\right)\psi\left(\tau\right)\right)^2 d\tau \geq -2\eta_c^2 A^4 \overline{\varphi}^2 \int\limits_{t}^{t+T} \left(\tau - t\right) \int\limits_{t}^{\tau} \left(\tilde{W}_c^T\left(\sigma\right)\psi\left(\sigma\right)\right)^2 d\sigma d\tau \\
+ \frac{1}{2}\underline{\psi}\tilde{W}_c^T\left(t\right)\tilde{W}_c\left(t\right) - 3\eta_c^2 A^4 \overline{\varphi}^2 \iota_5^2 T^3 - 6\eta_c^2 \iota_2^2 A^4 \overline{\varphi}^2 \int\limits_{t}^{t+T} \left(\tau - t\right) \int\limits_{t}^{\tau} \left\|\tilde{W}_a\left(\sigma\right)\right\|^4 d\sigma d\tau \\
- 6\eta_c^2 \bar{\epsilon}'^2 L_F^2 A^4 \overline{\varphi}^2 \int\limits_{t}^{t+T} \left(\tau - t\right) \int\limits_{t}^{\tau} \left\|e\right\|^2 d\sigma d\tau - 3\eta_c^2 A^4 \overline{\varphi}^2 \bar{\epsilon}'^2 L_F^2 d^2 T^3,
\end{aligned}
$$

where $A = \frac{1}{\sqrt{\nu\underline{\varphi}}}$. Changing the order of integration,

$$
\begin{aligned}
\int\limits_{t}^{t+T} \left(\tilde{W}_c^T\left(\tau\right)\psi\left(\tau\right)\right)^2 d\tau \geq \frac{1}{2}\underline{\psi}\tilde{W}_c^T\left(t\right)\tilde{W}_c\left(t\right) - \eta_c^2 A^4 \overline{\varphi}^2 T^2 \int\limits_{t}^{t+T} \left(\tilde{W}_c^T\left(\sigma\right)\psi\left(\sigma\right)\right)^2 d\sigma \\
- 3\eta_c^2 A^4 \overline{\varphi}^2 \bar{\epsilon}'^2 L_F^2 T^2 \int\limits_{t}^{t+T} \left\|e\left(\sigma\right)\right\|^2 d\sigma - 3\eta_c^2 \iota_2^2 A^4 \overline{\varphi}^2 T^2 \int\limits_{t}^{t+T} \left\|\tilde{W}_a\left(\sigma\right)\right\|^4 d\sigma \\
- 2\eta_c^2 A^4 \overline{\varphi}^2 T^3 \left(\iota_5^2 + \bar{\epsilon}'^2 L_F^2 d^2\right).
\end{aligned}
$$

Reordering the terms, (5–25) is obtained.

REFERENCES

[1] D. Kirk, *Optimal Control Theory: An Introduction*. Dover, 2004.

[2] O. von Stryk and R. Bulirsch, “Direct and indirect methods for trajectory optimization,” *Ann. Oper. Res.*, vol. 37, no. 1, pp. 357–373, 1992.

[3] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *J. Guid. Contr. Dynam.*, vol. 21, no. 2, pp. 193–207, 1998.

[4] C. R. Hargraves and S. Paris, “Direct trajectory optimization using nonlinear programming and collocation,” *J. Guid. Contr. Dynam.*, vol. 10, no. 4, pp. 338–342, 1987.

[5] G. T. Huntington, “Advancement and analysis of a gauss pseudospectral transcription for optimal control,” Ph.D. dissertation, Department of Aeronautics and Astronautics, MIT, May 2007.

[6] F. Fahroo and I. M. Ross, “Pseudospectral methods for infinite-horizon nonlinear optimal control problems,” *J. Guid. Contr. Dynam.*, vol. 31, no. 4, pp. 927–936, 2008.

[7] 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.

[8] C. L. Darby, W. W. Hager, and A. V. Rao, “An hp-adaptive pseudospectral method for solving optimal control problems,” *Optim. Control Appl. Methods*, vol. 32, no. 4, pp. 476–502, 2011. [Online]. Available: http://dx.doi.org/10.1002/oca.957

[9] D. Garg, W. W. Hager, and A. V. Rao, “Pseudospectral methods for solving infinite-horizon optimal control problems,” *Automatica*, vol. 47, no. 4, pp. 829 – 837, 2011.

[10] R. Freeman and P. Kokotovic, “Optimal nonlinear controllers for feedback linearizable systems,” in *Proc. Am. Control Conf.*, Jun. 1995, pp. 2722–2726.

[11] Q. Lu, Y. Sun, Z. Xu, and T. Mochizuki, “Decentralized nonlinear optimal excitation control,” *IEEE Trans. Power Syst.*, vol. 11, no. 4, pp. 1957–1962, Nov. 1996.

[12] V. Nevistic and J. A. Primbs, “Constrained nonlinear optimal control: a converse HJB approach,” California Institute of Technology, Pasadena, CA 91125, Tech. Rep. CIT-CDS 96-021, 1996.

[13] J. A. Primbs and V. Nevistic, “Optimality of nonlinear design techniques: A converse HJB approach,” California Institute of Technology, Pasadena, CA 91125, Tech. Rep. CIT-CDS 96-022, 1996.

[14] M. Sekoguchi, H. Konishi, M. Goto, A. Yokoyama, and Q. Lu, "Nonlinear optimal control applied to STATCOM for power system stabilization," in *Proc. IEEE/PES Transm. Distrib. Conf. Exhib.*, Oct. 2002, pp. 342–347.

[15] Y. Kim and F. Lewis, "Optimal design of CMAC neural-network controller for robot manipulators," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 30, no. 1, pp. 22 –31, feb 2000.

[16] Y. Kim, F. Lewis, and D. Dawson, "Intelligent optimal control of robotic manipulator using neural networks," *Automatica*, vol. 36, no. 9, pp. 1355–1364, 2000.

[17] K. Dupree, C. Liang, G. Hu, , and W. E. Dixon, "Global adaptive lyapunov-based control of a robot and mass-spring system undergoing an impact collision," *IEEE Trans. Syst. Man Cybern.*, vol. 38, pp. 1050–1061, 2008. [Online]. Available: http://ncr.mae.ufl.edu/papers/tcyber08.pdf

[18] K. Dupree, C. Liang, G. Hu, and W. E. Dixon, "Lyapunov-based control of a robot and mass-spring system undergoing an impact collision," *Int. J. Robot. Autom.*, vol. 206, no. 4, pp. 3166–3174, 2009.

[19] R. A. Freeman and P. V. Kokotovic, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Boston, MA: Birkhäuser, 1996.

[20] J. Fausz, V.-S. Chellaboina, and W. Haddad, "Inverse optimal adaptive control for nonlinear uncertain systems with exogenous disturbances," in *Proc. IEEE Conf. Decis. Control*, Dec. 1997, pp. 2654–2659.

[21] Z. H. Li and M. Krstic, "Optimal design of adaptive tracking controllers for nonlinear systems," *Automatica*, vol. 33, pp. 1459–1473, 1997.

[22] M. Krstic and Z.-H. Li, "Inverse optimal design of input-to-state stabilizing nonlinear controllers," *IEEE Trans. Autom. Control*, vol. 43, no. 3, pp. 336–350, March 1998.

[23] M. Krstic and P. Tsiotras, "Inverse optimal stabilization of a rigid spacecraft," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 1042–1049, May 1999.

[24] W. Luo, Y.-C. Chu, and K.-V. Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1639–1654, Nov. 2005.

[25] J. R. Cloutier, "State-dependent riccati equation techniques: an overview," in *Proc. Am. Control Conf.*, vol. 2, 1997, pp. 932–936.

[26] T. Çimen, "State-dependent riccati equation (SDRE) control: a survey," in *Proc. IFAC World Congr.*, 2008, pp. 6–11.

[27] ——, "Systematic and effective design of nonlinear feedback controllers via the state-dependent riccati equation (sdre) method," *Annu. Rev. Control*, vol. 34, no. 1, pp. 32–51, 2010.

[28] T. Yucelen, A. S. Sadahalli, and F. Pourboghrat, “Online solution of state dependent riccati equation for nonlinear system stabilization,” in *Proc. Am. Control Conf.*, 2010, pp. 6336–6341.

[29] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice - a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[30] D. Mayne and H. Michalska, “Receding horizon control of nonlinear systems,” *IEEE Trans. Autom. Contr.*, vol. 35, no. 7, pp. 814–824, 1990.

[31] M. Morari and J. Lee, “Model predictive control: past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[32] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Springer, 2000, vol. 26.

[33] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, pp. 789–814, 2000.

[34] E. F. Camacho and C. Bordons, *Model predictive control*. Springer, 2004, vol. 2.

[35] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer, 2011.

[36] R. Bellman, “The theory of dynamic programming,” DTIC Document, Tech. Rep., 1954.

[37] A. Barto, R. Sutton, and C. Anderson, “Neuron-like adaptive elements that can solve difficult learning control problems,” *IEEE Trans. Syst. Man Cybern.*, vol. 13, no. 5, pp. 834–846, 1983.

[38] R. Sutton, “Learning to predict by the methods of temporal differences,” *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[39] P. Werbos, “A menu of designs for reinforcement learning over time,” *Neural Netw. for Control*, pp. 67–95, 1990.

[40] C. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.

[41] R. Bellman, *Dynamic Programming*. Dover Publications, Inc., 2003.

[42] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2007.

[43] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[44] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[45] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” *IEEE Trans. Automat. Contr.*, vol. 42, no. 5, pp. 674–690, 1997.

[46] J. N. Tsitsiklis and B. V. Roy, “Average cost temporal-difference learning,” *Automatica*, vol. 35, no. 11, pp. 1799 – 1808, 1999.

[47] J. Tsitsiklis, “On the convergence of optimistic policy iteration,” *J. Mach. Learn. Res.*, vol. 3, pp. 59–72, 2003.

[48] V. Konda and J. Tsitsiklis, “On actor-critic algorithms,” *SIAM J. Contr. Optim.*, vol. 42, no. 4, pp. 1143–1166, 2004.

[49] P. Mehta and S. Meyn, “Q-learning and pontryagin’s minimum principle,” in *Proc. IEEE Conf. Decis. Control*, Dec. 2009, pp. 3598 –3605.

[50] S. Balakrishnan, “Adaptive-critic-based neural networks for aircraft optimal control,” *J. Guid. Contr. Dynam.*, vol. 19, no. 4, pp. 893–898, 1996.

[51] M. Abu-Khalaf and F. Lewis, “Nearly optimal HJB solution for constrained input systems using a neural network least-squares approach,” in *Proc. IEEE Conf. Decis. Control*, Las Vegas, NV, 2002, pp. 943–948.

[52] ——, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[53] R. Padhi, N. Unnikrishnan, X. Wang, and S. Balakrishnan, “A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems,” *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

[54] D. Vrabie, M. Abu-Khalaf, F. Lewis, and Y. Wang, “Continuous-time ADP for linear systems with partially unknown dynamics,” in *Proc. IEEE Int. Symp. Approx. Dyn. Program. Reinf. Learn.*, 2007, pp. 247–253.

[55] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, “Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, pp. 943–949, 2008.

[56] K. Vamvoudakis and F. Lewis, “Online synchronous policy iteration method for optimal control,” in *Recent Advances in Intelligent Control Systems*, W. Yu, Ed. Springer, 2009, pp. 357–374.

[57] ——, “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[58] D. Vrabie and F. Lewis, “Integral reinforcement learning for online computation of feedback nash strategies of nonzero-sum differential games,” in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 3066–3071.

[59] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. L. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.

[60] G. Lendaris, L. Schultz, and T. Shannon, "Adaptive critic design for intelligent steering and speed control of a 2-axle vehicle," in *Int. Joint Conf. Neural Netw.*, 2000, pp. 73–78.

[61] S. Ferrari and R. Stengel, "An adaptive critic global controller," in *Proc. Am. Control Conf.*, vol. 4, 2002, pp. 2665–2670.

[62] D. Han and S. Balakrishnan, "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 4, pp. 481–489, 2002.

[63] P. He and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 37, no. 2, pp. 425–436, 2007.

[64] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation -based neural network control of affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 90–106, Jan. 2008.

[65] 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.

[66] A. Heydari and S. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 145–157, 2013.

[67] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[68] D. Prokhorov, R. Santiago, and D. Wunsch, "Adaptive critic designs: A case study for neurocontrol," *Neural Netw.*, vol. 8, no. 9, pp. 1367–1372, 1995.

[69] X. Liu and S. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Am. Control Conf.*, vol. 3, 2000.

[70] J. Murray, C. Cox, G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 32, no. 2, pp. 140–153, 2002.

[71] R. Leake and R. Liu, "Construction of suboptimal control sequences," *SIAM J. Control*, vol. 5, p. 54, 1967.

[72] L. Baird, “Advantage updating,” Wright Lab, Wright-Patterson Air Force Base, OH, Tech. Rep., 1993.

[73] R. Beard, G. Saridis, and J. Wen, “Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation,” *Automatica*, vol. 33, pp. 2159–2178, 1997.

[74] K. Doya, “Reinforcement learning in continuous time and space,” *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[75] T. Hanselmann, L. Noakes, and A. Zaknich, “Continuous-time adaptive critics,” *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 631–647, 2007.

[76] D. Vrabie and F. 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.

[77] S. Bhasin, N. Sharma, P. Patre, and W. E. Dixon, “Robust asymptotic tracking of a class of nonlinear systems using an adaptive critic based controller,” in *Proc. Am. Control Conf.*, Baltimore, MD, 2010, pp. 3223–3228.

[78] Y. Jiang and Z.-P. Jiang, “Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics,” *Automatica*, vol. 48, no. 10, pp. 2699 – 2704, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109812003664

[79] X. Yang, D. Liu, and D. Wang, “Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints,” *Int. J. Control*, vol. 87, no. 3, pp. 553–566, 2014.

[80] L.-J. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” *Mach. Learn.*, vol. 8, no. 3-4, pp. 293–321, 1992.

[81] P. Cichosz, “An analysis of experience replay in temporal difference learning,” *Cybern. Syst.*, vol. 30, no. 5, pp. 341–363, 1999.

[82] S. Kalyanakrishnan and P. Stone, “Batch reinforcement learning in a complex domain,” in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, Honolulu, HI, 2007, pp. 650–657.

[83] L. Dung, T. Komeda, and M. Takagi, “Efficient experience reuse in non-markovian environments,” in *Proc. Int. Conf. Instrum. Control Inf. Technol.*, Tokyo, Japan, 2008, pp. 3327–3332.

[84] P. Wawrzyński, “Real-time reinforcement learning by sequential actor-critics and experience replay,” *Neural Netw.*, vol. 22, no. 10, pp. 1484–1497, 2009.

[85] S. Adam, L. Busoniu, and R. Babuska, “Experience replay for real-time reinforcement learning control,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 2, pp. 201–212, 2012.

[86] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy hdp iteration algorithm," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 4, pp. 937–942, 2008.

[87] 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.

[88] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptive control without persistent excitation," *IEEE Trans. Autom. Control*, vol. 32, pp. 134–145, 1987.

[89] K. Narendra and A. Annaswamy, *Stable Adaptive Systems*. Prentice-Hall, Inc., 1989.

[90] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Upper Saddle River, NJ: Prentice-Hall, 1989.

[91] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[92] G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *J. Guid. Contr. Dynam.*, vol. 34, no. 2, pp. 592–607, March 2011.

[93] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.

[94] X. Zhang and Y. Luo, "Data-based on-line optimal control for unknown nonlinear systems via adaptive dynamic programming approach," in *Proc. Chin. Control Conf.* IEEE, 2013, pp. 2256–2261.

[95] 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.

[96] L. K. R. Sutton, "Model-based reinforcement learning with an approximate, learned model," in *Proc.Yale Workshop Adapt. Learn. Syst.*, 1996, pp. 101–105.

[97] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Concurrent learning-based approximate optimal regulation," in *Proc. IEEE Conf. Decis. Control*, Florence, IT, Dec. 2013, pp. 6256–6261.

[98] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, ser. Dover Books on Mathematics. Dover Publications, 1999.

[99] S. Tijs, *Introduction to Game Theory*. Hindustan Book Agency, 2003.

[100] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory: Second Edition*, ser. Classics in Applied Mathematics. SIAM, 1999.

[101] J. Nash, “Non-cooperative games,” *Annals of Math.*, vol. 2, pp. 286–295, 1951.

[102] J. Case, “Toward a theory of many player differential games,” *SIAM J. Control*, vol. 7, pp. 179–197, 1969.

[103] A. Starr and C.-Y. Ho, “Nonzero-sum differential games,” *J. Optim. Theory App.*, vol. 3, no. 3, pp. 184–206, 1969.

[104] A. Starr and Ho, “Further properties of nonzero-sum differential games,” *J. Optim. Theory App.*, vol. 4, pp. 207–219, 1969.

[105] A. Friedman, *Differential games*. Wiley, 1971.

[106] A. Bressan and F. S. Priuli, “Infinite horizon noncooperative differential games,” *J. Differ. Equ.*, vol. 227, no. 1, pp. 230 – 257, 2006.

[107] A. Bressan, “Noncooperative differential games,” *Milan J. Math.*, vol. 79, no. 2, pp. 357–427, December 2011.

[108] M. Littman, “Value-function reinforcement learning in markov games,” *Cogn. Syst. Res.*, vol. 2, no. 1, pp. 55–66, 2001.

[109] Q. Wei and H. Zhang, “A new approach to solve a class of continuous-time nonlinear quadratic zero-sum game using adp,” in *IEEE Int. Conf. Netw. Sens. Control*, 2008, pp. 507–512.

[110] H. Zhang, Q. Wei, and D. Liu, “An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games,” *Automatica*, vol. 47, pp. 207–214, 2010.

[111] X. Zhang, H. Zhang, Y. Luo, and M. Dong, “Iteration algorithm for solving the optimal strategies of a class of nonaffine nonlinear quadratic zero-sum games,” in *Proc. IEEE Conf. Decis. Control*, May 2010, pp. 1359–1364.

[112] K. Vamvoudakis and F. Lewis, “Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations,” *Automatica*, vol. 47, pp. 1556–1569, 2011.

[113] Y. M. Park, M. S. Choi, and K. Y. Lee, “An optimal tracking neuro-controller for nonlinear dynamic systems,” *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1099–1110, 1996.

[114] T. Dierks and S. Jagannathan, “Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics,” in *Proc. IEEE Conf. Decis. Control*, 2009, pp. 6750–6755.

[115] ——, “Optimal control of affine nonlinear continuous-time systems,” in *Proc. Am. Control Conf.*, 2010, pp. 1568–1573.

[116] 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, 2011.

[117] M. Johnson, T. Hiramatsu, N. Fitz-Coy, and W. E. Dixon, “Asymptotic stackelberg optimal control design for an uncertain Euler-Lagrange system,” in *Proc. IEEE Conf. Decis. Control*, Atlanta, GA, 2010, pp. 6686–6691.

[118] K. Vamvoudakis and F. Lewis, “Online neural network solution of nonlinear two-player zero-sum games using synchronous policy iteration,” in *Proc. IEEE Conf. Decis. Control*, 2010.

[119] M. Johnson, S. Bhasin, and W. E. Dixon, “Nonlinear two-player zero-sum game approximate solution using a policy iteration algorithm,” in *Proc. IEEE Conf. Decis. Control*, 2011, pp. 142–147.

[120] K. Vamvoudakis, F. L. Lewis, M. Johnson, and W. E. Dixon, “Online learning algorithm for stackelberg games in problems with hierarchy,” in *Proc. IEEE Conf. Decis. Control*, Maui, HI, Dec. 2012, pp. 1883–1889.

[121] M. Lewis and K. Tan, “High precision formation control of mobile robots using virtual structures,” *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[122] T. Balch and R. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec 1998.

[123] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, “A vision-based formation control framework,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct 2002.

[124] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sept. 2004.

[125] R. Murray, “Recent research in cooperative control of multivehicle systems,” *J. Dyn. Syst. Meas. Control*, vol. 129, pp. 571–583, 2007.

[126] D. H. Shim, H. J. Kim, and S. Sastry, “Decentralized nonlinear model predictive control of multiple flying robots,” in *Proc. IEEE Conf. Decis. Control*, vol. 4, 2003, pp. 3621–3626.

[127] L. Magni and R. Scattolini, “Stabilizing decentralized model predictive control of nonlinear systems,” *Automatica*, vol. 42, no. 7, pp. 1231 – 1236, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000510980600104X

[128] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, “Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality,” *Automatica*, vol. 48, no. 8, pp. 1598 – 1611, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109812002476

[129] A. Heydari and S. N. Balakrishnan, “An optimal tracking approach to formation control of nonlinear multi-agent systems,” in *Proc. AIAA Guid. Navig. Control Conf.*, 2012.

[130] D. Vrabie, “Online adaptive optimal control for continuous-time systems,” Ph.D. dissertation, University of Texas at Arlington, 2010.

[131] S. P. Singh, “Reinforcement learning with a hierarchy of abstract models,” in *AAAI Natl. Conf. Artif. Intell.*, vol. 92, 1992, pp. 202–207.

[132] C. G. Atkeson and S. Schaal, “Robot learning from demonstration,” in *Int. Conf. Mach. Learn.*, vol. 97, 1997, pp. 12–20.

[133] P. Abbeel, M. Quigley, and A. Y. Ng, “Using inaccurate models in reinforcement learning,” in *Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2006, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143845

[134] M. P. Deisenroth, *Efficient reinforcement learning using Gaussian processes*. KIT Scientific Publishing, 2010.

[135] D. Mitrovic, S. Klanke, and S. Vijayakumar, “Adaptive optimal feedback control with learned internal dynamics models,” in *From Motor Learning to Interaction Learning in Robots*, ser. Studies in Computational Intelligence, O. Sigaud and J. Peters, Eds. Springer Berlin Heidelberg, 2010, vol. 264, pp. 65–84.

[136] M. P. Deisenroth and C. E. Rasmussen, “Pilco: A model-based and data-efficient approach to policy search,” in *Int. Conf. Mach. Learn.*, 2011, pp. 465–472.

[137] Y. Luo and M. Liang, “Approximate optimal tracking control for a class of discrete-time non-affine systems based on gdhp algorithm,” in *IWACI Int. Workshop Adv. Comput. Intell.*, 2011, pp. 143–149.

[138] D. Wang, D. Liu, and Q. Wei, “Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach,” *Neurocomputing*, vol. 78, no. 1, pp. 14 – 22, 2012.

[139] J. Wang and M. Xin, “Multi-agent consensus algorithm with obstacle avoidance via optimal control approach,” *Int. J. Control*, vol. 83, no. 12, pp. 2606–2621, 2010. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00207179.2010.535174

[140] ——, “Distributed optimal cooperative tracking control of multiple autonomous robots,” *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 572 – 583,

2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889011002211

[141] ——, “Integrated optimal formation control of multiple unmanned aerial vehicles,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1731–1744, 2013.

[142] W. Lin, “Distributed uav formation control using differential game approach,” *Aerosp. Sci. Technol.*, vol. 35, pp. 54–62, 2014.

[143] E. Semsar-Kazerooni and K. Khorasani, “Optimal consensus algorithms for cooperative team of agents subject to partial information,” *Automatica*, vol. 44, no. 11, pp. 2766 – 2777, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109808002719

[144] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.

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

[146] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. Dixon. (2013) Approximately optimal trajectory tracking for continuous time nonlinear systems. arXiv:1301.7664.

[147] G. Chowdhary, “Concurrent learning adaptive control for convergence without persistencey of excitation,” Ph.D. dissertation, Georgia Institute of Technology, December 2010.

[148] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[149] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[150] A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures.” *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.

[151] K. Hornik, M. Stinchcombe, and H. White, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural Netw.*, vol. 3, no. 5, pp. 551 – 560, 1990.

[152] 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.

[153] K. M. Misovec, “Friction compensation using adaptive non-linear control with persistent excitation,” *Int. J. Control*, vol. 72, no. 5, pp. 457–479, 1999.

[154] K. Narendra and A. Annaswamy, “Robust adaptive control in the presence of bounded disturbances,” *IEEE Trans. Automat. Contr.*, vol. 31, no. 4, pp. 306–315, 1986.

[155] E. Panteley, A. Loria, and A. Teel, “Relaxed persistency of excitation for uniform asymptotic stability,” *IEEE Trans. Autom. Contr.*, vol. 46, no. 12, pp. 1874–1886, 2001.

[156] A. Loría and E. Panteley, “Uniform exponential stability of linear time-varying systems: revisited,” *Syst. Control Lett.*, vol. 47, no. 1, pp. 13 – 24, 2002.

[157] R. Kamalapurkar, H. T. Dinh, P. Walters, and W. E. Dixon, “Approximate optimal cooperative decentralized control for consensus in a topological network of agents with uncertain nonlinear dynamics,” in *Proc. Am. Control Conf.*, Washington, DC, June 2013, pp. 1322–1327.

[158] 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.

BIOGRAPHICAL SKETCH

Rushikesh Kamalapurkar received his Bachelor of Technology degree in mechanical engineering from Visvesvaraya National Institute of Technology, Nagpur, India. He worked for two years as a Design Engineer at Larsen and Toubro Ltd., Mumbai, India. He received his Master of Science degree and his Doctor of Philosophy degree from the Department of Mechanical and Aerospace Engineering at the University of Florida under the supervision of Dr. Warren E. Dixon. His research interests include dynamic programming, optimal control, reinforcement learning, and data-driven adaptive control for uncertain nonlinear dynamical systems.