OPTIMAL TRACKING CONTROL OF UNCERTAIN SYSTEMS: ON-POLICY AND OFF-POLICY REINFORCEMENT LEARNING APPROACHES

by

HAMIDREZA MODARES

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2015

This work is dedicated to my wife Bahare for her understanding and patience and being part of my life during the production of this work.

Acknowledgements

I would like to express my sincere thanks to my advisor Dr. Frank L. Lewis for his guidance, support and motivation during my doctoral research. I would always be indebted to him for introducing me in the fields of reinforcement learning for control and distributed control of multi-agent systems. In addition to science, I have learnt several things from him in my life.

I am thankful to my committee members, Dr. William E. Dillan, Dr. Dan Popa, Dr. Manfred Huber and Dr. Mickael T. Manry, for their insightful comments. I thank my collaborators, Dr. Zhong-ping Jiang for giving me a new perspective about off-policy reinforcement learning, Dr. Ali Davoudi for introducing me to the filed of resilent control of cyber-physical multi-agent systems, and Subramanya Nageshrao for his contribution to Chapter 6 of my dissertation, during his stay in UTA as a visiting scholar. I also thank Gail Paniuski for her assistance with finding the answer to every question I had.

I thank my wife for her support and sacrifice. I was fortunate to have her with me during my PhD. Her unconditional encouragement saw me through to the end. I cannot thank her enough.

Finally, I would want to thank my parents for their unconditional support and encouragement during my student time at the University of Texas at Arlington.

June 26, 2015

Abstract

OPTIMAL TRACKING CONTROL OF UNCERTAIN SYSTEMS: ON-POLICY AND OFF-POLICY REINFORCEMENT LEARNING APPROACHES

HAMIDREZA MODARES, PhD

The University of Texas at Arlington, 2015

Supervising Professor: FRANK L. LEWIS

Over the last few decades, strong connections between reinforcement learning (RL) and optimal control have prompted a major effort towards developing online and model-free RL algorithms to learn the solution to optimal control problems. Although RL algorithms have been widely used to solve the optimal regulation problems, few results considered solving the optimal tracking control problem (OTCP), despite the fact that most real-world control applications are tracking problems. On the other hand, existing methods for solving OTCP require complete knowledge of the system dynamics.

This research begins with developing an adaptive optimal algorithm for linear quadratic tracking problem (LQT). A discounted performance function is introduced for the LQT problem. A discounted algebraic Riccati equation (ARE) is then derived which gives the solution to the LQT problem. The integral reinforcement learning (IRL) technique and off-policy RL technique are used to learn the solution to the discounted ARE online and without requiring complete knowledge of the system dynamics. The proposed idea is then extended to solve optimal tracking control for nonlinear systems. The input constraints are also taken into account for nonlinear systems.

In the next step, the proposed method is extended to solve the CT two-player zero-sum game arising in the H_∞ tracking control problem. An off-policy RL algorithm is developed which enables us to find the solution to the H∞ tracking control problem online in real time and without requiring the disturbance being adjustable, which is usually impractical for most of real systems.

The next results show how to design dynamic OPFP controllers for CT linear systems with unknown dynamics. To this end, it is first shown that the system state can be constructed using some limited observations on the system output over a period of the history of the system. A Bellman equation is then developed to evaluate a control policy and find an improved policy simultaneously using only some limited observations on the system output. Then, using this Bellman equation, a model-free IRL-based OPFB controller is developed.

Next, a model-free approach is developed for solving output synchronization of heterogeneous multi-agent systems. Both the leader's and the follower's dynamics is assumed to be unknown. First, a distributed adaptive observer is designed to estimate the leader's state for each agent. A model-free off-policy RL algorithm is then developed to solve the optimal output synchronization problem online in real time. It is shown that this distributed RL approach implicitly solves the output regulation equations without actually doing so and without requiring knowledge of the leader or of agent's dynamics.

Finally, a model-free RL based method is design for the human-robot interaction system to help the robot adapt itself to the level of the human skills. This assists the human operator to perform a given task with minimum workload demands and optimize the overall human-robot system performance. First, a robot-specific neuro-adaptive controller is designed to make the unknown nonlinear robot behave like a prescribed robot impedance model. Then, a task-specific outer-loop controller is designed to find the optimal parameters of the prescribed robot impedance model, online in real time.

Table of Contents

List of Illustrations

Chapter 1

INTRODUCTION

This introductory chapter discusses motivation, background and contribution.

1.1. Adaptive optimal control using reinforcement learning

Optimal control involves the design of a control policy that satisfies a tracking or regulation control objective while simultaneously minimizes a performance function. A sufficient condition to find a feedback solution to an optimal regulation problem is to solve the Hamilton-Jacobi-Bellman (HJB) equation. For linear systems with quadratic performance function, the HJB equation reduces to the algebraic Riccati equation (ARE). For the case of optimal tracking problem, however, traditional solutions are composed of two components; a feedback term obtained by solving an HJB equation and a feedforward term obtained a priori by either solving a differential equation [60] or applying inverse dynamic concept [80]. The feedback term tries to stabilize the tracking error dynamics and the feedforward term tries to guarantee perfect tracking. Procedures for computing the feedback and feedforward terms are traditionally based on offline solution methods which require complete knowledge of the system dynamics.

Motivated by the desire to eliminate the requirement for exact knowledge of the system dynamics, reinforcement learning (RL) [9], [15], [40], [61], [63], [89], [95], [100], [120], [124], [117], [125], [126], [129], [141], has been extensively used to solve optimal control problems. RL technique, inspired by learning mechanisms observed in mammals, is a computational approach to learning from interactions with the surrounding environment and concerned with how an agent or actor ought to take actions so as to optimize a cost of its long-term interactions with the environment. In the context of control, the environment is the dynamic system, the agent corresponds to the controller, and actions correspond to control signals. The RL objective is to find a strategy that minimizes an expected long-term cost. Unlike traditional optimal control solutions, RL does not require the exact knowledge of the system dynamics. Instead, RL largely relies upon experience gathered from tacking actions and directly interacting with the system dynamics.

During the last few years, RL methods have been successfully used to solve the optimal regulation problems by learning the solution to the HJB equation. RL algorithms for solving optimal control problems are usually based on policy iteration (PI) and value iteration (VI). PI algorithms have two steps, namely, policy evaluation and policy improvement. In the policy evaluation step, the value function related to a control policy is evaluated. An improved control policy is then obtained in the policy improvement step based on the assessment of this value function. PI algorithms must start from an admissible control policy, which requires that the initial control policy be stabilizing. On the other hand, VI algorithms do not require an initial stabilizing control policy. Werbos [125], [126] defined a family of VI algorithms implemented on actor-critic structures to solve optimal control problems online for discrete-time systems. In these structures, the actor learns to select an action based on evaluative feedback from the critic to minimize a performance index. Both PI and VI algorithms use the state value function (or V-function) to update their policies. V-functions only describe the quality of the system states. In order to obviate the need to have knowledge of the system dynamics, Werbos introduced action-dependent heuristic dynamic programming and Watkins [119] used the state-action value function (or Q-function) and presented a Q-learning algorithm for linear discrete-time systems.

1.2. Background and Motivation

Reinforcment learning has been widely used to solve optimal control problems [2], [5], [6], [7], [16], [17], [23], [59], [62], [68], [70], [73], [74], [75], [76], [82], [97]

, [98], [106], [107], [108], [109], [114], [116], [121], [122], [123], [130], [131], [132]. For continuous-time (CT) systems, which are the focus of this work, Vrabie and Lewis [112], [113] proposed a promising RL algorithm, called integral reinforcement learning (IRL), to learn the solution to the HJB equation using only partial knowledge about the system dynamics. They used an iterative online PI procedure to implement their IRL algorithm. The IRL algorithm is an on-policy algorithm. That is, the algorithm must follow the policy which it is learning about and so it learns only about the executing policy. In an off-policy RL algorithm, on the other hand, the algorithm learns about a policy or policies different from the one which it is executing. Off-policy RL algorithms were presented in [51], [52] to solve the optimal regulation problem for completely unknown CT systems. In these algorithms, both value function and policy are updated at the same time by evaluation of a Bellman equation. Moreover, these algorithms take into account the effect of the probing noise to avoid any bias in solving the Bellman equation. An off-policy RL algorithm was proposed in [78] to solve the H_∞ control problem for partially-unknown systems. Other than the IRL and off-policy based PI algorithms, efficient synchronous PI algorithms with guaranteed closed-loop stability were proposed in [16], [106] to learn the solution to the HJB equation. Synchronous IRL algorithms were also presented for solving the HJI equation in [108]. The interested reader is referred to [63] and the references therein for details of the existing RL methods for solving optimal control problems.

Although IRL and off-policy RL algorithms have been successfully used to solve the optimal regulation problems, few results considered solving the optimal tracking control problems (OTCPs) for both discrete-time [24], [43], [55], [143], [138] and CT systems [25], [136]. Moreover, existing methods require the exact knowledge of the system dynamics a priori. In order to attain the required knowledge of the system dynamics, in [138], a plant model was first identified and then an RL-based optimal tracking controller was synthesized using the identified model. To our knowledge, there has been no attempt to develop RL-based techniques to solve the OTCP for CT systems with unknown or partially-unknown dynamics using only measured data in real time. While the importance of the IRL algorithm and off-policy RL algorithm are well understood for solving optimal regulation problems for partially or completely unknown systems, the requirement of the exact knowledge of the system dynamics for finding the steady-state part of the control input in the existing OTCP formulation does not allow extending the IRL algorithm or the off-policy RL algorithm for solving the OTCP.

Another important issue which is ignored in the existing RL based solutions to the OTCP is the amplitude limitation on the control inputs. In fact, in the existing formulation for the OTCP, it is not possible to encode the input constraints into the optimization problem a priori, as only the cost of the feedback part of the control input is considered in the performance function. Therefore, the existing RL-based solutions to the OTCP offer no guarantee on the remaining control inputs on their permitted bounds during and after learning. This may result in performance degradation or even system instability. In the context of the constrained optimal regulation problem, however, an offline PI algorithm [3] was presented to find the solution to the constrained HJB equation.

Moreover, existing model-free RL algorithms for CT systems require measurement of the system states. However, it is not possible to measure the full states of the systems in many practical situations. OPFB-based controllers are more desirable than state-feedback controllers in these applications. For discrete-time systems, in [62] an RL-based method was developed which used only measured input/output data from the system to learn the optimal control policy. However, developing OPFB controllers using past measured data for CT is considerably more complicated and needs more math development and proofs and therefore has been not considered yet.

Finally, the design of model-free optimal output syhcnronization for heterogeneous multi-agent systems, in which a distributed control protocol is designed to make all agents output follow the leader output, has not been considered in the literature. Existing solutions to this problem [20], [42], [41], [42], [77], [128], [134], [135], however, require complete knowledge of the agent and leader dynamics, which is not available in many real-world applications. This is because these methods require the explicit solution to the output regulation equations.

This work attempts to address these mentioned issues and provide efficient RL-based methods for optimal tracking control of uncertain systems.

1.3. Contribution and Outline

The key contributions of the dissertation are listed as follows.

- Online RL algorithms are developed for learning the solution to OTCP of CT systems with partially-unknown or completely unknown dynamics.
    - ✓ In Chapter 2, the linear quadratic tracking (LQT) problem for uncertain CT systems is solved using RL algorithms. The LQT problem is first transformed into minimizing a discounted performance function subject to an augmented system, composed of the original system and the command generator system. An LQT ARE equation is then developed which gives both feedforward and feedback parts of optimal control solution simultaneously. Then, IRL and off-policy RL algorithms are used to learn the solution to the LQT ARE for systems with partially-unknown and completely unknown dynamics.

  - ✓ In chapter 3, an RL-based solution for solving OTCP of uncertain constrained-input nonlinear systems is presented. In contrast to existing methods for OTCP, input constraints are taken into account into the optimization problem a priori. A tracking constrained HJB equation is developed and rigorous proofs of stability and optimality of the HJB solution are provided. An online IRL algorithm with guaranteed stability is provided to learn the solution to the tracking constrained HJB equation for partially-unknown systems.
  - ✓ In Chapter 4, an off-policy RL algorithm is presented to solve the H_∞ tracking control of nonlinear CT systems with completely unknown dynamics. A tracking Hamilton-Jacobi-Isaac (HJI) equation is developed to give the solution to the optimization problem in hand. An iterative off-policy RL algorithm is used to learn the solution to the tracking HJI equation without requiring any knowledge of the system dynamics. Convergence of the proposed algorithm to the solution to the tracking HJI equation is verified.
- OPFB controllers are designed to learn the solution to both LQR and LQT problems for systems with partially-unknown and completely unknown dynamics.
  - ✓ In Chapter 5, a dynamic OPFB controller is designed for completely unknown dynamics using off-policy algorithm. A discounted performance function is employed such that the proposed off-policy algorithm can be used to solve both LQR and LQT problems. A novel Bellman equation is then developed to evaluate a control policy and find an improved policy simultaneously using only some limited observations on the system output over a period of the history of the system. Then, using this

Bellman equation, an off-policy RL algorithm is developed to find an optimal policy based on only measured outputs and without requiring the knowledge of the system state or the system dynamics. Convergence to the optimal solution is verified.

- Reinforcement learning is used to solve optimal output synchronization problem for heterogeneous multi-agent linear systems.
  - ✓ In Chapter 6, a novel distributed model-free controller is designed to solve the output synchronization problem for heterogeneous multi-agent systems. A distributed adaptive observer is first designed to estimate the leader state for each agent, without requiring the knowledge of the leader's dynamics. The estimated leader state along with the local state of each agent is then used by the agent to design a model-free optimal local controller. Therefoe, the optimal output synchronization problem is cast into a set of optimal output tracking problems for a set of decoupled systems. It is shown that solving a set of decoupled discounted AREs solves the output synchronization problem. Online model-free solution to these decoupled AREs is then found by using an off-policy RL algorithm. It is shown that this distributed reinforcement learning approach implicitly solves the output regulation equations without actually doing so and without requiring any knowledge of the leader's dynamics or of the agent's dynamics.
- Reinforcement learning is used to design an iintelligent human-robot interaction (HRI) system with adjustable robot behavior.
  - ✓ In Chapter 7, an HRI system is designed to assist the human operator to perform a given task with minimum workload demands and optimize the

overall HRI system performance. First, a robot-specific neuro-adaptive controller is designed in the inner loop to make the unknown nonlinear robot behave like a prescribed robot impedance model as perceived by a human operator. Then, a task-specific outer-loop controller is designed to find the optimal parameters of the prescribed robot impedance model to adjust the robot's dynamics to the operator skills and minimize the tracking error. IRL algorithm is used to find the optimal parameters of the prescribed robot impedance model without the requirement of the knowledge of the human model.

1.4. Publications resulted form this work

1- H. Modares, F. L. Lewis, and Z. P. Jiang, "H_∞ Tracking Control of Completely-unknown Continuous-time Systems," accepted for publication in *IEEE Transactions on Neural Networks and Learning Systems, 2015*.

2- H. Modares, and F. L. Lewis, "Linear Quadratic Tracking Control of Partially-Unknown Continuous-time Systems using Reinforcement Learning," *IEEE Transactions on Automatic control*, *vol. 59, pp.3051-3056, 2014.*

3- H. Modares, and F. L. Lewis, "Optimal Tracking Control of Nonlinear Partially-unknown Constrained-input Systems using Integral Reinforcement Learning," *Automatica*, *Vol. 50, no. 7, pp. 1780-1792, 2014.*

4- H. Modares, F. L. Lewis, and D. Popa, "Optimized Assistive Human-robot Interaction using Reinforcement Learning," *accepted for publication in IEEE Transaction on Cybernetics.*

5- H. Modares, B. Kiumarsi, F. L. Lewis, Z. P. Jiang, "Optimal Output-feedback Control of Unknown Continuous-time Linear Systemsusing Reinforcement Learning," *Conditionally accepted for publication in IEEE Transactions on Cybernetics, 2015*.

6- S. P. Neshrao, H. Modares, G. Lopes, R. Babuska, F. L. Lewis, "Optimal Model-free Output Synchronization of Heterogeneous Systems Using Off-policy Reinforcement Learning," *Submitted to Automatica*, 2015.

Chapter 2

LINEAR QUDRATIC TRACKING CONTROL OF PARTIALLY-UNKNOWN AND COMPLETELY UNKNOWN SYSTEMS

2.1. Introduction

This chapter is concerned with developing online IRL and off-policy RL algorithms to solve the LQT problem for partially-unknown and completely unknown CT systems.

Traditional solutions to the LQT problem are composed of two components; a feedback term obtained by solving an ARE and a feedforward term obtained by either solving a differential equation [60] or calculating a desired control input a priori using knowledge of the system dynamics [81]. The feedback term tries to stabilize the tracking error dynamics and the feedforward term tries to guarantee perfect tracking. Procedures for computing the feedback and feedforward terms are traditionally based on offline solution methods which must be done in a noncausal manner backwards in time and require complete knowledge of the system dynamics.

RL algorithms has been mainly used to solve optimal regulator problems, and only few results considered solving optimal tracking problems. This is mainly because of the additional computational burden created by computing the feedforward control term that is not presented in optimal regulator problems. Existing RL solutions to the optimal tracking problem employ the dynamic inversion concept or solve output regulator equations to obtain the feedforward control term a priori and then find the optimal feedback control term using RL techniques. However, these methods require complete knowledge of the system dynamics.

In this chapter, online adaptive controllers based on IRL and off-policy RL algorithms are developed which converge to the optimal solution of the LQT problem without requiring complete knowledge of the system dynamics or the command generator dynamics. The algorithm starts with an admissible nonoptimal control policy and learns an optimal control policy using only measured data from the system and the command generator in real time. To achieve this goal, it is first shown that the value function is quadratic in terms of the system state and the reference trajectory and an augmented system is constructed from the original system and the command generator. Using the quadratic structure of the value function, a novel Bellman equation and an augmented LQT ARE equation are derived for the LQT problem. This formulation allows extending the IRL and off-policy RL techniques to learn the solution to the LQT ARE without requiring complete knowledge of the system dynamics. Convergence of the proposed learning algorithm to an optimal control solution is verified.

The reminder of this chapter is organized as follows. In the next section, the LQT problem and its standard solution are discussed. In Section 2.3 it is shown that solving the LQT problem is equivalent to solving an augmented ARE. Section 2.4. presents IRL and off-policy RL algorithms to solve the LQT problem without the need for complete knowledge of the system dynamics or the command generator dynamics. Simulation results and conclusion are discussed in Sections 2.5 and 2.6, respectively.

2.2. LQT problem and its standard solution

In this section, the infinite-horizon LQT problem and its standard solution are presented for CT systems. It is assumed in this section that the reference trajectory is generated by an asymptotically stable system. That is, the reference trajectory goes to zero as time goes to infinity.

Consider the linear CT system

$$\begin{aligned} \dot{x} &= A\,x + B\,u \\ y &= C\,x \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^{n\times 1}$ is a measurable system state vector, $y \in \mathbb{R}^{p\times 1}$ is the system output, $u \in \mathbb{R}^{m\times 1}$ is the control input, $A \in \mathbb{R}^{n\times n}$ gives the drift dynamics of the system, $B \in \mathbb{R}^{n\times m}$ is the input matrix and $C \in \mathbb{R}^{p\times n}$ is the output matrix.

Assumption 2.1. The pair (A, B) is stabilizable and the pair $(A, \sqrt{Q}\,C)$ is observable.

The goal of the optimal tracking problem is to find the optimal control policy u^* so as to make the system (2.1) track a desired (reference) trajectory $y_d(t) \in \mathbb{R}^{p\times 1}$ in an optimal manner by minimizing a predefined performance index. In the infinite-horizon LQT problem, the performance index is usually considered as

$$J(x, \overline{y}_d) = \frac{1}{2}\int_t^{\infty} \left[(Cx - y_d)^T Q (Cx - y_d) + u^T R\, u\right] d\tau \tag{2.2}$$

where $\overline{y}_d = \{y_d(\tau),\, t \le \tau\}$, $Q > 0$ and $R > 0$ are symmetric matrices, and $(Cx - y_d)^T Q\,(Cx - y_d) + u^T R\, u$ is the utility function.

The standard solution to the LQT problem is given as [60]

$$u = -R^{-1}B^T S\ x + R^{-1}B^T\ v_{SS} \tag{2.3}$$

where *S* is obtained by solving the ARE

$$0 = A^T S + SA - S\,B\,R^{-1}B^T S + C^T Q\,C \tag{2.4}$$

and the limiting function v_{SS} is given by $v_{SS} = \lim_{T\to\infty} v$, with the auxiliary time signal v satisfies

$$-\dot{v} = (A - B\,R^{-1}B^T S)^T v + C^T Q\,y_d, \quad v(T) = 0 \tag{2.5}$$

The first term of the control input (2.3) is a feedback control part that depends linearly on the system state, and the second term is a feedforward control part that depends on the reference trajectory. The feedforward part of the control input is time varying in general and thus a theoretical difficulty arises in the solution of the infinite-horizon LQT problem. In [10], [11], methods for real-time computation of v_{ss} are provided**.**

Remark 2.1. Note that the performance function (2.2) is unbounded if the reference trajectory does not approach zero as time goes to infinity. This is because the feedforward part of the control input and consequently the second term under the integral of the performance function (2.2) depends on the reference trajectory. Therefore, standard methods can only be used if the reference trajectory is generated by an asymptotically stable system.

2.3. Augmented ARE for causal solution of the infinite-horizon LQT problem

In this section, a causal solution to the LQT problem is presented. It is assumed that the reference trajectory is generated by a linear command generator and it is then shown that the value function for the LQT problem is quadratic in the system state and the reference trajectory. An augmented LQT ARE for this system is derived to solve the LQT problem in a causal manner.

Assumption 2.2. Assume that the reference trajectory $y_d(t)$ is generated by the command generator system

$$\dot{y}_d = F\,y_d \tag{2.6}$$

where F is a constant matrix of appropriate dimension.

Remark 2.2. Matrix F is not assumed stable. The command generator dynamics given in (2.6) can generate a large class of useful command trajectories, including unit step (useful, e.g., in position command), sinusoidal waveforms (useful, e.g., in hard disk drive control), damped sinusoids (useful, e.g., in vibration quenching in flexible beams), the ramp (useful in velocity tracking systems, e.g., satellite antenna pointing), and more.

As was discussed in Section 2.2, the use of the performance function (2.2) for the LQT problem requires the command generator be asymptotically stable, i.e., F in (2.6) must be Hurwitz. In order to relax this restrictive assumption, a discounted performance function is introduced for the LQR problem as follows

$$J(x,\overline{y}_d) = \frac{1}{2}\int_t^{\infty} e^{-\gamma(\tau-t)}[(Cx-y_d)^T Q(Cx-y_d) + u^T R u]d\tau \tag{2.7}$$

where $\gamma > 0$ is the discount factor.

Definition 2.1. Admissible control. A control policy $\mu(x)$ is said to be admissible with respect to (2.2), if $\mu(x)$ is continuous, $\mu(0)=0$, $u(x)=\mu(x)$ stabilizes (2.1), and $J(x(t),\overline{y}_d)$ is finite $\forall\, x(t) \text{ and } \overline{y}_d$.

Lemma 2.1. Quadratic form of the LQT value function. Consider the LQT problem with the system dynamics and the reference trajectory dynamics given as (2.1) and (2.6), respectively. Consider the admissible fixed control policy

$$u = Kx + K'y_d \tag{2.8}$$

Then, the value function (2.7) for control policy (2.8) can be written as the quadratic form

$$J(x(t),\overline{y}_d) = V(x(t),y_d(t)) = \frac{1}{2}[x(t)^T\, y_d(t)^T]\,P\,[x(t)^T\, y_d(t)^T]^T \tag{2.9}$$

for some symmetric $P>0$.

Proof: Putting (2.8) in the value function (2.2) and performing some manipulations yields

$$\begin{aligned} V(x(t),y_d(t)) = \frac{1}{2}\int_0^{\infty} e^{-\gamma\tau}\Big[&x(\tau+t)^T(C^TQC+K^TRK)x(\tau+t) \\ &+ 2x(\tau+t)^T(-C^TQ+K^TRK')y_d(\tau+t) + y_d(\tau+t)^T(Q+K'^TRK')y_d(\tau+t)\Big]d\tau \end{aligned} \tag{2.10}$$

Using (2.8), the solutions for the linear differential equations (2.1) and (2.6) become

$$x(\tau + t) = e^{(A+BK)\tau}\, x(t) + (\int_0^{\tau} e^{(A+BK)\tau'} B K' e^{F\tau'} d\tau')\, y_d(t) \equiv L_1(\tau)\, x(t) + L_2(\tau)\, y_d(t) \quad (2.11)$$

$$y_d(\tau + t) = e^{F\tau}\ y_d(t) \equiv L_3(\tau)\, y_d(t) \quad (2.12)$$

Substituting (2.11) and (2.12) in (2.10) results in

$$V(x(t), y_d(t)) = \frac{1}{2}\Big[x(t)^T y_d(t)^T\Big] P\ \Big[x(t)^T y_d(t)^T\Big]^T \quad (2.13)$$

where $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ with

$$P_{11} = \int_0^{\infty} e^{-\gamma\tau} L_1(\tau)^T\ \ C^T Q C + K^T R K\ \ L_1(\tau)\, d\tau \quad (2.14)$$

$$P_{12} = \int_0^{\infty} e^{-\gamma\tau} \Big[L_1(\tau)^T\ \ C^T Q C + K^T R K\ \ L_2(\tau) + L_1(\tau)^T\ \ -C^T Q + K^T R K'\ \ L_3(\tau)\Big] d\tau \quad (2.15)$$

$$P_{21} = \int_0^{\infty} e^{-\gamma\tau} \Big[L_2(\tau)^T\ \ C^T Q C + K^T R K\ \ L_1(\tau) + L_3(\tau)^T\ \ -Q C + K'^T R K\ \ L_1(\tau)\Big] d\tau \quad (2.16)$$

$$\begin{aligned} P_{22} = \int_0^{\infty} e^{-\gamma\tau} \Big[& L_3(\tau)^T (Q + K'^T R K') L_3(\tau) + L_2(\tau)^T (C^T Q C + K^T R K) L_2(\tau) \\ & + 2 L_2(\tau)^T (-C^T Q + K^T R K') L_3(\tau)\Big] d\tau \end{aligned} \quad (2.17)$$

This completes the proof. □

Note that equation (2.9) is valid because Assumptions 2.2 is imposed. Also, note that because the closed-loop system is stable for an admissible policy, L_1 and L_2 in (2.14)-(2.17) are bounded. The boundness of L_3 and consequently the existence of a solution to the LQT problem is discussed in the following remark.

Remark 2.3. If the reference trajectory is bounded (i.e., if F is stable or marginally stable, e.g., tracking a step or sinusoidal waveform), then L_3 and is bounded for every $\gamma > 0$. However, if the command generator dynamics F in (2.6) is unstable, then the first and last terms of P_{22} in (2.17) can be unbounded for some values of γ. More specifically, one can conclude form (2.17) that P_{22} is bounded if $(F-0.5\gamma I)$ has all its poles in the left-hand side of the complex plane. Therefore, if F is unstable, we need to know an upper bound of the real part of unstable poles of the F to choose γ large enough to make sure P_{22} is bounded and thus a solution to the LQT exists.

Now define the augmented system state as

$$X(t) = \left[x(t)^T \;\; y_d(t)^T\right]^T \tag{2.18}$$

Putting (2.1) and (2.6) together construct the augmented system as

$$\dot{X} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} X + \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} u \equiv T\,X + B_1\,u \tag{2.19}$$

The value function (2.9) in terms of the augmented system state becomes

$$V\;\; X(t) \;\; = \frac{1}{2} X(t)^T P\;\; X(t) \tag{2.20}$$

Using value function (2.20) for the left-hand side of (2.7) and differentiating (2.7) along with the trajectories of the augmented system (2.19) gives the augmented LQT Bellman equation

$$0 = (T + B_1\,K_1)^T P\,X + X^T P(T + B_1\,K_1) - \gamma X^T P X + X^T C_1^{\;T} Q\,C_1\,X + u^T R\,u \tag{2.21}$$

where

$$C_1 = [C \;\text{-}\, I] \tag{2.22}$$

Consider the fixed control input (2.8) as

$$u = K\,x + K' y_d = K_1\,X \tag{2.23}$$

where $K_1 = [K \;\; K']$. Putting (2.20) and (2.23) into (2.21), the LQT Bellman equation gives the augmented LQT Lyapunov equation

$$(T + B_1 K_1)^T P + P(T + B_1 K_1) - \gamma P + C_1^T Q C_1 + K_1^T R K_1 = 0 \tag{2.24}$$

Based on (2.21), define the Hamiltonian

$$H(X,u,P) = (TX + B_1 u)^T PX + X^T P(TX + B_1 u) - \gamma X^T PX + X^T C_1^T Q C_1 X + u^T R u \tag{2.25}$$

Theorem 2.1. Causal solution for the LQT problem. The optimal control solution for the infinite-horizon LQT problem is given by

$$u = K_1 X \tag{2.26}$$

where

$$K_1 = -R^{-1} B_1^T P \tag{2.27}$$

and P satisfies the augmented LQT ARE

$$0 = T^T P + PT - \gamma P - PB_1 R^{-1} B_1^T P + C_1^T Q C_1 \tag{2.28}$$

Proof: A necessary condition for optimality is stationarity condition

$$\frac{\partial H}{\partial u} = B_1^T P X + R u = 0 \tag{2.29}$$

which results in control input (2.26). Substituting (2.20) and (2.26) in the LQT Bellman equation (2.21) yields (2.28). This completes the proof. □

Lemma 2.2. Existence of the solution to the LQT ARE. The LQT ARE (2.28) has a unique positive semi-definite solution if (A,B) is stabilizable and the discount factor $\gamma > 0$ is chosen such that $F - 0.5\gamma I$ is stable.

Proof. Note that the LQT ARE (2.28) can be written as

$$0 = (T - 0.5\gamma I)^T P + P(T - 0.5\gamma I) - PB_1 R^{-1} B_1^T P + C_1^T Q C_1 \tag{2.30}$$

This amounts to an ARE without discount factor and with the system dynamics given by $T - 0.5\gamma I$ and B_1. Therefore, a unique solution to the LQT ARE (2.30) and consequently the LQT ARE (2.28) exists if $(T-0.5\gamma I, B_1)$ is stabilizable. This requires that $(A-0.5\gamma I, B)$ be stabilizable and $F-0.5\gamma I$ be stable. However, since (A,B) is stabilizable, then $(A-0.5\gamma I, B)$ is also stabilizable for any $\gamma > 0$. This completes the proof.

Remark 2.4. The fact that $F-0.5\gamma I$ should be stable to have a solution to the LQT ARE supports the conclusion of Remark 2.3 for the existence of a solution to the LQT problem. In Remark 2.3, it is further elaborated how to choose the discount factor to make sure the LQT problem has a solution.

Remark 2.5. The optimal control input (2.26) can be written in form of $u = K\,x + K' y_d$, as in (2.23). Therefore, similar to the standard solution given in Section 2.2, the proposed control solution (2.26) has both feedback feedforward control parts. However, in the proposed method, both control parts are obtained simultaneously by solving an LQT ARE in a causal manner. This causal formulation is a consequence of Assumption 2.2 and the quadratic form (2.9), (2.20).

Now a formal proof is given to show that the LQT ARE solution makes the tracking error $e_d = Cx - y_d$ bounded and it asymptotically stabilizes $\overline{e}_d(t) \equiv e^{-(\gamma/2)t} e_d(t)$. The following key fact is instrumental.

Lemma 2.3. For any admissible control policy $u(X)$, let P be the corresponding solution to the Bellman equation (2.21). Define $u^*(X) = -R^{-1} B_1^T P\,X$. Then

$$H(X,u,P) = H(X,u^*,P) + (u-u^*)^T R (u-u^*) \qquad (2.31)$$

where H is the Hamiltonian function defined in (2.25).

Theorem 2.2. Stability of the LQT ARE solution. Consider the LQT problem for the system (2.1) with performance function (2.7). Suppose that P^* is a smooth positive-definite solution to the tracking LQT ARE (2.28) and define the control input $u^* = -R^{-1}B_1^T P^* X$. Then, u^* makes $\bar{e}_d(t) \equiv e^{-(\gamma/2)t} e_d(t)$ asymptotically stable.

Proof. For any continuous value function $V(X) = X^T P X$, by differentiating $V(X)$ along the augmented system trajectories, one has

$$\frac{dV(X)}{dt} = (TX + B_1 u)^T PX + X^T P(TX + B_1 u) \tag{2.32}$$

so that

$$H(X,u,P) = \frac{dV(X)}{dt} - \gamma V(X) + X^T C_1^T Q C_1 X + u^T R u \tag{2.33}$$

Suppose now that P^* satisfies the LQT ARE (2.28). Then, using (2.31) and since $H(X^*, u^*, P^*) = 0$, one has

$$\frac{dV(X)}{dt} - \gamma V(X) + X^T C_1^T Q C_1 X + u^T R u = (u - u^*)^T R (u - u^*) \tag{2.34}$$

Selecting $u = u^* = K_1 X$ gives

$$\frac{dV(X)}{dt} - \gamma V(X) + X^T (C_1^T Q C_1 + K_1^T R K_1) X = 0 \tag{2.35}$$

where K_1 is the control gain obtained by solving the LQT ARE and it is given in (2.27). Multiplying $e^{-\gamma t}$ to the both sides of (2.35) and using $V(X) = X^T P X$ gives

$$\frac{d}{dt}(e^{-\gamma t} X^T P X) = -e^{-\gamma t} X^T (C_1^T Q C_1 + K_1^T R K_1) X \le 0 \tag{2.36}$$

Now define the new state $\bar{X}(t) = e^{-(\gamma/2)t} X(t)$ and consider the Lyapunov function $V(\bar{X}) = \bar{X}^T P \bar{X}$. Then using (2.36) one has

$$\dot{V}(\bar{X}) = -\bar{X}^T ({C_1}^T Q C_1 + K_1^T R K_1)\bar{X} < 0 \tag{2.37}$$

Therefore $\bar{X}(t)$ is asymptotically stable. On the other hand, since $\bar{e}_d = C_1 \bar{X}$ and $C_1 \neq 0$, thus $\bar{e}$ is also asymptotically stable. □

Remark 2.6. Note that a discounted performance function is used in [27], section 3.6, for optimal tracking control of N-player differential games. However, it does not consider developing a value function in terms of both the state and the desired trajectory and consequently obtaining both feedback and feedforward control inputs simultaneously by solving a LQT ARE.

Remark 2.7. The discount factor γ and the weight matrix Q in (2.7) are design parameters and they can be chosen appropriately to make the system state goes to a very small region around zero. The larger the Q is, the more negative the Lyapunov function (2.37) is and consequently the faster the tracking error decreases. Also, the smaller the discount factor is, the faster the tracking error decreases.

2.4. Reinforcement learning algorithms for finding the solution to the LQT ARE

In this section, first an offline solution to the LQT ARE is presented. Then, a CT Bellman equation is developed based on the IRL idea. Based on this, IRL algorithm is employed to solve the LQT problem online in real time and without the need for the knowledge of drift dynamics of the system Ax and command generator dynamics $F y_d$. Finally, off-policy RL algorithm is employed to learn the solution to the LQT ARE without requiring any knowledge of the system dynamics and the command generator.

The LQT Lyapunov equation (2.24), which can be solved to evaluate a fixed control policy, is linear in P and is easier to solve than the LQT ARE (2.28). This is the motivation for introducing an iterative technique to solve the LQT problem. An iterative Lyapunov method for solving the LQT problem is given as follows.

Algorithm 2.1. Offline policy iteration for solving the LQT problem

Initialization: Start with an admissible control input $u = K_1^0 X$

Policy evaluation: Given a control gain K_1^i, find P^i using the LQT Lyapunov equation

$$(T - 0.5\gamma I + B_1 K_1^i)^T P^i + P^i (T - 0.5\gamma I + B_1 K_1^i) + C_1^T Q C_1 + (K_1^i)^T R\, (K_1^i) = 0 \tag{2.38}$$

Policy improvement: update the control gain using

$$K_1^{i+1} = -R^{-1} B_1^T P^i \tag{2.39}$$

Algorithm 2.1 is an offline algorithm which extends Kleinman's algorithm [56] to the LQT problem. It is shown in [56] that if the initial control policy is stabilizing, then all subsequent control policies will also be stabilizing. Convergence of Kleinman's algorithm to the solution of the ARE is also shown.

To obviate the need for complete knowledge of the system dynamics, the IRL algorithm [112], [113] can be extended to the LQT problem. The IRL is a PI algorithm which uses an equivalent formulation of the Lyapunov equation that does not involve the system dynamics. Hence, it is central to the development of model-free RL algorithms for CT systems. To obtain the IRL Bellman equation for the LQT problem, note that for time interval $\Delta t > 0$, the value function (2.7) satisfies

$$V\ X(t)\ = \frac{1}{2} \int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q\ C_1\, X(t) + u^T R\, u \right] d\tau + e^{-\gamma \Delta t}\, V\ X(t+\Delta t) \tag{2.40}$$

where C_1 is defined in (2.22). Using (2.20) in (2.40) yields the LQT IRL Bellman equation

$$X(t)^T P X(t) = \int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + u^T R\, u \right] d\tau + e^{-\gamma \Delta t} X(t+\Delta t)^T P X(t+\Delta t) \tag{2.41}$$

The first term of (2.41) is known as the integral reinforcement.

Lemma 2.4. Equivalence of the Lyapunov equation (2.24) and the IRL Bellman equation (2.41). The LQT IRL Bellman equation (2.41) and the LQT Lyapunov equation (2.24) have the same positive semi-definite solution for value function.

Proof. Dividing both sides of (2.41) by Δt and taking limit yields

$$\begin{aligned}&\lim_{\Delta t\to 0}\frac{e^{-\gamma\Delta t}X(t+\Delta t)^T P\,X(t+\Delta t)-X(t)^T P\;X(t)}{\Delta t}+\\ &\lim_{\Delta t\to 0}\frac{\int_t^{t+\Delta t} e^{-\gamma(\tau-t)}\left[X(t)^T C_1^{\,T}Q\;C_1\,X(t)+u^T R\,u\right]d\tau}{\Delta t}=0\end{aligned}\tag{2.42}$$

By L'Hopital's rule, then

$$\lim_{\Delta t\to 0}\frac{\int_t^{t+\Delta t} e^{-\gamma\;\tau-t}\left[X(t)^T C_1^{\,T}Q\;C_1\,X(t)+u^T R\;u\right]d\tau}{\Delta t}=X(t)^T C_1^{\,T}Q\;C_1\,X(t)+u^T R\;u\tag{2.43}$$

and also

$$\begin{aligned}&\lim_{\Delta t\to 0}\frac{X(t)^T P\;X(t)-e^{-\gamma\Delta t}X(t+\Delta t)^T P\,X(t+\Delta t)}{\Delta t}=\\ &\lim_{\Delta t\to 0}\;-\gamma e^{-\gamma\Delta t}X(t+\Delta t)^T P\,X(t+\Delta t)+e^{-\gamma\Delta t}\dot{X}(t+\Delta t)^T P\,X(t+\Delta t)+\\ &e^{-\gamma\Delta t}X(t+\Delta t)^T P\,\dot{X}(t+\Delta t)\;\;=-\gamma X(t)^T P\,X(t)+\dot{X}(t)^T P\,X(t)+X(t)^T P\,\dot{X}(t)\end{aligned}\tag{2.44}$$

Using the system dynamics (2.19) in (2.44) and putting (2.43) and (2.44) in (2.42) gives the Bellman equation (2.21). On the other hand, the Bellman equation (2.21) has the same value function solution as the Lyapunov equation (2.24) and this completes the proof. □

Using (2.41) instead of (2.24) in policy evaluation step of Algorithm 2.1, the following IRL-based algorithm is obtained.

Algorithm 2.2. Online IRL algorithm for solving the LQT problem

Initialization: Start with an admissible control input $u^0 = K_1^0 X$

Policy evaluation: Given a control policy u^i, find P^i using the Bellman equation

$$X(t)^T P^i X(t) = \frac{1}{2}\int_t^{t+\Delta t} e^{-\gamma\,\tau - t}\,\left[X(t)^T C_1^T Q\, C_1\, X(t) + (u^i)^T R\,(u^i)\right] d\tau + e^{-\gamma\Delta t} X(t+\Delta t)^T P^i\, X(t+\Delta t) \tag{2.45}$$

Policy improvement: update the control input using

$$u^{i+1} = -R^{-1} B_1^T P^i X \tag{2.46}$$

The policy evaluation and improvement steps (2.45) and (2.46) are repeated until the policy improvement step no longer changes the present policy, thus convergence to the optimal controller is achieved. That is, until $\left\|P^{i+1} - P^i\right\| \le \varepsilon$ is satisfied, where ε is a small constant. Algorithm 2.2 does not require knowledge of A and F.

According to Lemma 2.4, the IRL Bellman equation (2.45) in Algorithm 2.2 has the same value function solution as the Lyapunov equation (2.38) in Algorithm 2.1. Therefore, iterating between (2.45) and (2.46) in Algorithm 2.2 is equivalent to iterating between (2.38) and (2.39) in Algorithm 2.1. Thus, similar to Algorithm 2.1, if the initial control policy is stabilizing in Algorithm 2.2, then all subsequent control policies will be stabilizing and the algorithm converges to the optimal policy, provided that the unique solution to the IRL Bellman equation (2.45) is obtained at each iteration. This unique solution can be uniquely determined using the least squares technique under some PE condition, as shown in [113].

Remark 2.8. The PE condition can be satisfied by injecting a probing noise into the control input. This can cause biased results. However, it was shown in [62] that discounting the performance function can significantly reduce the deleterious effects of probing noise. Moreover, since the probing noise is known a priori, one can consider its effect into the IRL Bellman equation, as in [59], to avoid affecting the convergence of the learning process.

Remark 2.9. The proposed IRL Algorithm 2.2 has the same structure as the IRL algorithm in [113] for solving the LQR problem. However, in the proposed algorithm, the augmented system state involves the reference trajectory in it and also a discount factor is used in the IRL Bellman equation of Algorithm 2.2. In fact, using Assumption 2.2 and developing Lemmas 2.1 and 2.4 and Theorem 2.1 allows us to extend the IRL algorithm to the LQT problem.

Remark 2.10. The solution for P^i in the policy evaluation step (2.45) is generally carried out in a least squares (LS) sense. In fact (2.45) is a scalar equation and P is a symmetric $n \times n$ matrix with $n(n+2)/2$ independent elements and therefore at least $n(n+2)/2$ data sets are required before (2.45) can be solved using LS. Both batch LS and recursive LS methods can be used to perform policy evaluation step (2.45).

Remark 2.11. The proposed policy iteration Algorithm 2.2. requires an initial admissible policy. If one knows that the system to be control is itself stable, which is true for many cases, then the initial policy can be chosen as $u = 0$ and the admissibility of the initial policy is guaranteed without requiring any knowledge of A. Moreover, if the reference trajectory is bounded, which is true for most real-world applications, no knowledge of F is needed. Otherwise, the initial admissible policy can be obtained by using some knowledge of T. Suppose the system (2.1) has a nominal model T_N satisfying $T = T_N + \Delta T$, where ΔT is unknown part of T. In this case, one can use robust control methods such as H_∞ control with the nominal model T_N to yield an admissible initial policy. Note that the learning process does not require any knowledge of T. Finally, Algorithm 2.2 is a policy iteration algorithm and IRL value iteration can be used to avoid the need for an initial admissible policy.

The IRL Algorithm 2.2 requires knowledge of the input dynamics B. In the following, we extend this method for discounted performance function such that it can be used for solving LQT problem. To this end, the system dynamics is first written as

$$\dot{x} = A_i\, x + B(K^i x + u) \tag{2.47}$$

with $A_i = A + BK^i$. Then, one has the following Bellman equation

$$\begin{aligned} &e^{-\gamma T} x(t+T)^T P^i\, x(t+T) - x(t)^T P^i\, x(t) = \int_t^{t+T} e^{-\gamma(\tau - t)} [x^T (A_i^T P^i + P^i A_i - \gamma P^i) x \\ &+ 2(u + K^i x)^T B^T P^i x]\, d\tau = -\int_t^{t+T} e^{-\gamma(\tau - t)} x^T Q_i\, x\, d\tau + \\ &2\int_t^{t+T} e^{-\gamma(\tau - t)} (u + K^i x)^T R\, K^{i+1} x\, d\tau \end{aligned} \tag{2.48}$$

where $Q_i = C^T Q C + (K^i)^T R (K^i)$. For a fixed control gain K^i, the Bellman equation (2.48) can be solved for both the value function kernel matrix P^i and the updated improved gain K^{i+1}, simultaneously. The following Algorithm 2.3 uses the above Bellman equation to iteratively solve the ARE equation.

Algorithm 2.3. Online Off-policy RL algorithm for solving LQT problem

Initialization: Start with a control policy $u^0 = K^0\, x + e$, where K^0 is stabilizing and e is the probing noise.

Policy evaluation and improvement: Solve the following Bellman equation for P^i and K^{i+1} simultaneously

$$e^{-\gamma T} x(t+T)^T P^i\, x(t+T) - x(t)^T P^i\, x(t) = -\int_t^{t+T} e^{-\gamma(\tau-t)} x^T Q_i\, x\, d\tau + \\ 2\int_t^{t+T} e^{-\gamma(\tau-t)} (u + K^i x)^T R\, K^{i+1}\, x\, d\tau \tag{2.49}$$

Stop if a stopping criterion is met, otherwise set $i = i + 1$ and got to 2.

2.5. Simulation results

In this section, an example is provided to verify the correct performance of Algorithm 2.2 for solving the LQT problem.

Consider the unstable continuous-time linear system

$$\dot{x}(t) = \begin{bmatrix} 0.5 & 1.5 \\ 2.0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 5 \\ 1 \end{bmatrix} u(t), \qquad y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \tag{2.50}$$

and suppose that the desired trajectory is generated by the command generator system

$$\dot{y}_d = 0 \tag{2.51}$$

with the initial value $y_d(0) = 3$. So, the reference trajectory is a step input with amplitude 3. The performance index is given as (2.2) with $Q = 10$ and $R = 1$, and the discount factor is chosen as $\gamma = 0.1$.

The solution obtained by directly solving the LQT ARE (2.28) using known dynamics (T, B_1) is given by

$$P^* = \begin{bmatrix} 0.6465 & 0.0524 & -0.6221 \\ 0.0524 & 0.0191 & -0.0244 \\ -0.6221 & -0.0244 & 1.7360 \end{bmatrix} \tag{2.52}$$

and hence using (2.27) the optimal control gain becomes

$$K_1^* = \begin{bmatrix} -3.2851 & -0.2813 & 3.1347 \end{bmatrix} \tag{2.53}$$

It is now assumed that the system drift dynamics and the command generator dynamics are unknown and Algorithm 2.2 is implemented online to solve the LQT problem for the system. The simulation was conducted using data obtained from the augmented system at every 0.05 s. A batch least squares problem is solved after 6 data samples and thus the controller is updated every 0.3s. The initial control policy is chosen as $K_1 = [-5.0 \quad -1.0 \quad -0.5]$. Fig. 2.1 shows how the norm of the difference between the optimal P matrix and the P matrix obtained by the online learning algorithm converges to zero. Also, Fig 2.2 depicts the norm of the difference between the optimal control gain and the control gain obtained by the learning algorithm. From Figs. 2.1 and 2.2, it is clear that the value function and control gain parameters converge to their optimal values in (2.52) and (2.53) after four iterations. Thus, the solution of the LQT ARE is obtained at time *t*=1.2s. Fig. 2.3 shows the output and the desired trajectory during simulation. It can be seen that the output tracks the desired trajectory after the optimal control is found.

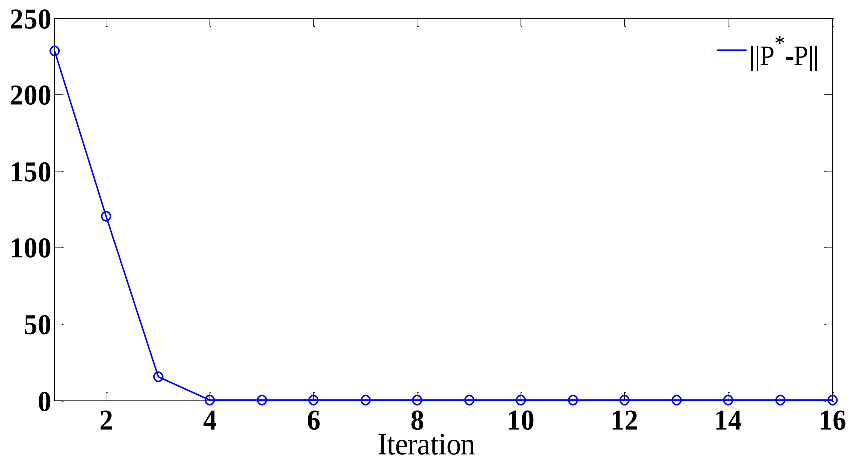

Fig. 2.1. Convergence of the P matrix parameters to their optimal values

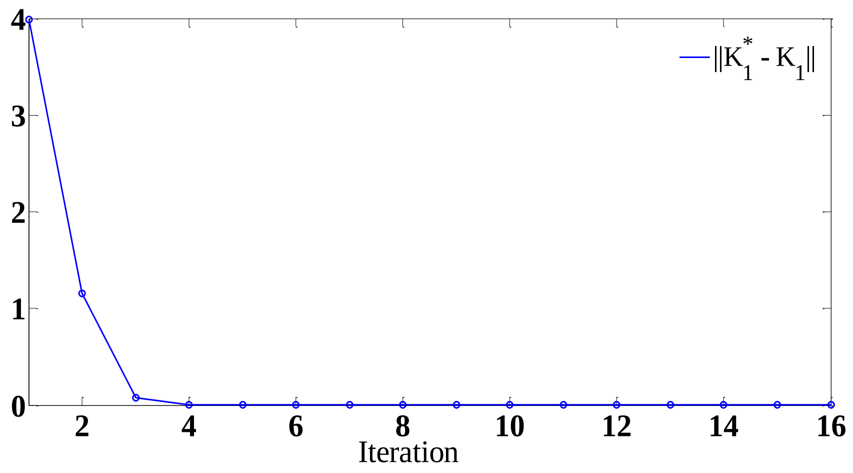

Fig. 2.2. Convergence of the control gain parameters to their optimal values

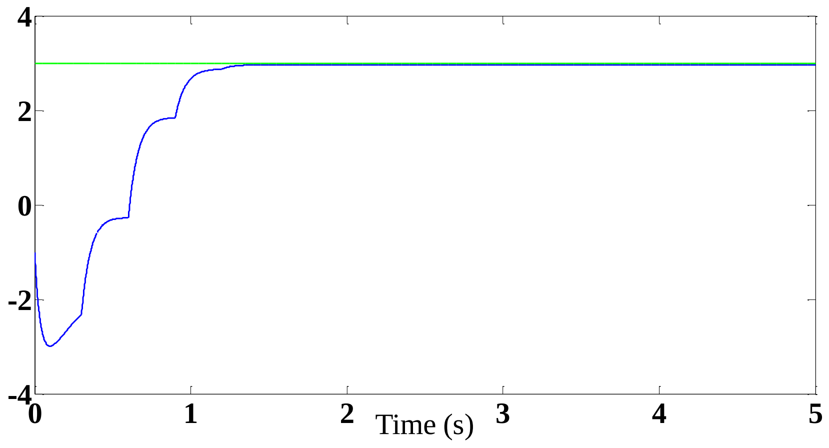

Fig. 2.3. System output versus reference trajectory

2.6. Conclusion

Online learning algorithms based on reinforcement learning were presented to find the solution to the LQT problem without requiring the knowledge of the system dynamics as well as the command generator dynamics. No preceding identification procedure was used to identify the unknown dynamics and only measured data using the system and the command generator were used to learn the optimal policy. It was shown that the proposed algorithm converges to the optimal solution of the LQT problem. A simulation example was provided to justify our claim.

Chapter 3

OPTIMAL TRACKING CONTROL OF PARTIALLY-UNKNOWN NONLINEAR CONSTRAINED-INPUT SYSTEMS

3.1. Introduction

This chapter extends the results of the previous chapter to nonlinear systems. In this case the theoretical development becomes a bit more complicated since finding the optimal solution requires the solution to a tracking HJB equation, a nonlinear partial differential equation which is in general impossible to be solved analytically. Moreover, the amplitude limitation on the control inputs is taken into account. In fact, in the existing formulation for the optimal tracking control problem (OTCP), it is not possible to encode the input constraints into the optimization problem a priori, as only the cost of the feedback part of the control input is considered in the performance function. Therefore, the existing RL-based solutions to the OTCP offer no guarantee on the remaining control inputs on their permitted bounds during and after learning. This may result in performance degradation or even system instability.

In this chapter, an online adaptive controller is developed based on the IRL technique to learn the OTCP solution for nonlinear continuous-time systems without knowing the system drift dynamics or the command generator dynamics. The input constraints are encoded into the optimization problem a priori by employing a suitable nonquadratic performance function. A tracking HJB equation related to this nonquadratic performance function is derived which gives both feedforward and feedback parts of the control input simultaneously. An IRL algorithm, implemented on an actor-critic structure, is used to find the solution to the tracking HJB equation online using only partial knowledge about the system dynamics. In contrast to the existing work, a preceding identification procedure is not needed and the optimal policy is learned using only measured data from the system. Convergence of the proposed learning algorithm to a near-optimal control solution and the boundness of the tracking error and the actor and critic NNs weights during learning are also shown.

The remainder of this chaper is organized as follows. The next section formulates the optimal tracking problem and provides the standard solution to it. A new formulation for the OTCP is presented in Section 3.3 and the traking Bellman and HJB equations corresponding to this formulation are found in Section 3.4.Section 3.5 shows how to find the solution to the tracking HJB equation online in real time and using only partial knowledge about the system dynamics. Sections 3.6 and 3.7 provide the simulation results and conclusion, respectively.

3.2. Optimal tracking control for nonlinear systems

In this section, a review of the OTCP for continuous-time nonlinear systems is given. It is pointed out that the standard solution to the given problem requires complete knowledge of the system dynamics. It is also pointed out that the input constraints caused by the actuator saturation cannot be encoded into the standard performance function a priori. A new formulation of the OTCP problem is given in the next section to overcome these shortcomings.

3.2.1. Problem Formulation

Consider the affine CT dynamical system describe by

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

where $x \in \mathbb{R}^n$ is the measurable system state vector, $f(x) \in \mathbb{R}^n$ is the drift dynamics of the system, $g(x) \in \mathbb{R}^{n\times m}$ is the input dynamics of the system, and $u(t) \in \mathbb{R}^m$ is the control input. The elements of $u(t)$ are defined by $u_i(t),\, i = 1,...,m$.

Assumption 3.1. It is assumed that $f(0)=0$ and $f(x)$ and $g(x)$ are lipschitz, and that the system (3.1) is controllable in the sense that there exists a continuous control on a set $\Omega \subseteq \mathbb{R}^n$ which stabilizes the system.

Assumption 3.2 . The following assumptions are considered on the system dynamics

a) $\|f(x)\| \le b_f \|x\|$ for some constant b_f .

b) $g(x)$ is bounded by a constant b_g, i.e. $\|g(x)\| \le b_g$.

Note that Assumption 3.2(a) requires $f(x)$ be lipschitz and $f(0)=0$ (see Assumption 3.1) which is a standard assumption to make sure the solution $x(t)$ of the system (3.1) is unique for any finite initial condition. On the other hand, although Assumption 3.2(b) restricts the considered class of nonlinear systems, many physical systems, such as robotic systems [96] and aircraft systems fulfill such a property.

The goal of the optimal tracking problem is to find the optimal control policy $u^*(t)$ so as to make the system (3.1) track a desired (reference) trajectory $x_d(t) \in \mathbb{R}^{n\times 1}$ in an optimal manner by minimizing a predefined performance function. Moreover, the input must be constrained to remain within predefined limits $|u_i(t)| \le \lambda,\ i=1,...,m$.

Define the tracking error as

$$e_d(t) \triangleq x(t) - x_d(t) \tag{3.2}$$

A general performance function leading to the optimal tracking controller can be expressed as

$$V(e_d(t), x_d(t)) = \int_t^{\infty} e^{-\gamma(\tau - t)} [E(e_d(\tau)) + U(u(\tau))]\, d\tau \tag{3.3}$$

where $E(e_d)$ is a positive-definite function, $U(u)$ is a positive-definite integrand function, and γ is the discount factor.

Note that the performance function (3.3) contains both the tracking error cost and the whole control input energy cost. The following assumption is made in accordance to other work in the literature.

Assumption 3.3. The desired reference trajectory $x_d(t)$ is bounded and there exists a Lipschitz continuous command generator function $h_d(x_d(t)) \in \mathbb{R}^n$ such that

$$\dot{x}_d(t) = h_d(x_d(t)) \tag{3.4}$$

and $h_d(0) = 0$.

Note that the reference dynamics need only be stable in the sense of Lyapunov, not necessarily asymptotically stable.

3.2.2. Standard formulation and solution to the OTCP

In this section, the standard solution to the OTCP and its shortcomings are discussed. In the existing standard solution to the OTCP, the desired or the steady-state part of the control input $u_d(t)$ is obtained by assuming that the desired reference trajectory satisfies

$$\dot{x}_d(t) = f(x_d(t)) + g(x_d(t))\, u_d(t) \tag{3.5}$$

If the dynamics of the system is known and the inverse of the input dynamics $g^{-1}(x_d(t))$ exists, the steady-state control input which guarantees perfect tracking is given by

$$u_d(t) = g^{-1}(x_d(t))\,(\dot{x}_d(t) - f(x_d(t)) \tag{3.6}$$

On the other hand, the feedback part of the control is designed to stabilize the tracking error dynamics in an optimal manner by minimizing the following performance function

$$V(e_d(t)) = \int_t^{\infty} [e_d(\tau)^T Q\, e_d(\tau) + u_e^{\ T} R u_e]\, d\tau \tag{3.7}$$

where $u_e(t) = u(t) - u_d(t)$ is the feedback control input. The optimal feedback control solution $u_e^{\ *}(t)$ which minimizes (3.7) can be obtained by solving the HJB equation related to this performance function.

The standard optimal solution to the OTCP is then constituted by the optimal feedback control $u_e^{\ *}(t)$ obtained.

Remark 3.1. The optimal feedback part of the control input $u_e^{\ *}(t)$ can be learned using the integral reinforcement learning method to obviate knowledge of the system drift dynamics. However, the exact knowledge of the system dynamics is required to find the steady-state part of the control input given by (3.6), which cancels the usefulness of the IRL technique.

Remark 3.2. Because only the feedback part of the control input is obtained by minimizing the performance function (3.7), it is not possible to encode the input constraints into the optimization problem by using a nonquadratic performance function, as has been performed in the optimal regulation problem [3].

3.3. A new formulation for OTCP of CT constrained-input systems

In this section, a new formulation for the OTCP is presented. In this formulation, both the steady-state and feedback parts of the control input are obtained simultaneously by minimizing a new discounted performance function in the form of (3.3). The input constraints are also encoded into the optimization problem a priori. A tracking HJB equation for the constrained OTCP is derived and an iterative offline IRL algorithm is presented to find its solution. This algorithm provides a basis to develop an online IRL algorithm for learning the optimal solution to the OTCP for partially-unknown systems, which is discussed in the next section.

In the following, first an augmented system composed of the tracking error dynamics and the command generator dynamics is constructed. Then, based on this augmented system, a new discounted performance function for the OTCP is presented. It is shown that this performance function is identical to the performance function (3.3).

The tracking error dynamics can be obtained by using (3.1) and (3.2), and one has

$$\dot{e}_d(t) = f(x(t)) - h_d(x_d(t)) + g(x(t))\,u(t) \tag{3.8}$$

Define the augmented system state

$$X(t) = \left[e_d(t)^T \;\; x_d(t)\right]^T \in \mathbb{R}^{2n} \tag{3.9}$$

Then, putting (3.4) and (3.8) together yields the augmented system

$$\dot{X}(t) = F(X(t)) + G(X(t))\,u(t) \tag{3.10}$$

where $u(t) = u(X(t))$ and

$$F(X(t)) = \begin{bmatrix} f(e_d(t) + x_d(t)) - h_d(x_d(t)) \\ h_d(x_d(t)) \end{bmatrix} \tag{3.11}$$

$$G(X(t)) = \begin{bmatrix} g(e_d(t) + x_d(t)) \\ 0 \end{bmatrix} \tag{3.12}$$

Based on the augmented system (3.10), the following discounted performance function is introduced for the OTCP.

$$V(X(t)) = \int_t^\infty e^{-\gamma(\tau - t)}[X(\tau)^T Q_T\, X(\tau) + U(u(\tau))]\,d\tau \tag{3.13}$$

where $\gamma > 0$ is the discount factor,

$$Q_T = \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \; Q > 0 \tag{3.14}$$

and $U(u)$ is a positive-definite integrand function defined as

$$U(u) = 2\int_0^u (\lambda\,\beta^{-1}(v/\lambda))^T\, R\, dv \tag{3.15}$$

where $v \in \mathbb{R}^m$, $\beta(.) = \tanh(.)$, λ is the saturating bound for the actuators and $R = diag(r_1,...,r_m) > 0$ is assumed to be diagonal for simplicity of analysis. This nonquadratic performance function is used in the optimal regulation problem of constrained-input systems to deal with the input constraints [3], [79]. Denote $\omega(v) = (\lambda\beta^{-1}(v/\lambda))^T R = \left[\omega_1(v_1)\ldots\omega_m(v_m)\right]$. Then the integral in (3.15) is defined as

$$U(u) = 2\int_0^u \omega(v)\, dv = 2\sum_{i=1}^{m}\int_0^{u_i} \omega_i(v_i)\, dv_i \tag{3.16}$$

It is clear that $U(u)$ in (3.16) is a scalar for $u \in \mathbb{R}^m$. In fact, using this nonquadratic performance function, the following constraints are always satisfied.

$$\left|u_i(t)\right| \le \lambda \quad\; i = 1,...,m \tag{3.17}$$

Note that from (3.10)-(3.12) it is clear that, as expected, the command generator dynamics are not under our control. Since they are assumed be bounded, the admissibility of the control input implies the boundness of the states of the augmented system.

Remark 3.3. Note that for the first term under the integral we have $X^T Q_T\, X = e_d^{\;T} Q\, e_d$. Therefore, this performance function is identical to the performance function (3.3) with $E(e_d(\tau)) = e_d^{\;T} Q\, e_d$ and $U(u)$ given in (3.15).

Remark 3.4. The use of the discount factor in the performance function (3.13) is essential. This is because the control input contains a steady-state part which in general makes (3.13) unbounded without using a discount factor, and therefore the meaning of minimality is lost.

Remark 3.5. Note that both steady-state and feedback parts of the control input are obtained simultaneously by minimizing the discounted performance function (3.13) along the trajectories of the augmented system (3.10). As is shown in the subsequent sections, this formulation enables us to extend the IRL technique to find the solution to the OTCP without requiring the augmented system dynamics F. That is, both the system drift dynamics f and the command generator dynamics h_d are not required.

3.4. Tracking Bellman and Tacking HJB Equations

In this section, the optimal tracking Bellman equation and the optimal tracking HJB equation related to the defined performance function (3.13) are given.

Using Leibniz's rule to differentiate V along the augmented system trajectories (3.10), the following tracking Bellman equation is obtained

$$\dot{V}(X) = \int_t^{\infty} \frac{\partial}{\partial t} e^{-\gamma(\tau - t)} \big(X^T Q_T X + U(u) \big) d\tau - X^T Q_T X - U(u) \tag{3.18}$$

Using (3.15) in (3.18) and noting that the first term in the right hand side of (3.18) is equal to $\gamma V(X)$, gives

$$X^T Q_T X + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv - \gamma V(X) + \dot{V}(X) = 0 \tag{3.19}$$

or, by defining the Hamiltonian function

$$\begin{aligned} H(X, u, \nabla V) \equiv X^T Q_T X + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv - \gamma V(X) \\ + \nabla V^T(X)\,(F(X) + G(X)\,u(X)) = 0 \end{aligned} \tag{3.20}$$

where $\nabla V(X) = \partial V(X) / \partial X \in \mathbb{R}^{2n}$. Let $V^*(X)$ be the optimal cost function defined as

$$V^*(X(t)) = \min_{u \in \pi(\Omega)} \int_t^\infty e^{-\gamma(\tau - t)} [X^T Q_T X + U(u)] \, d\tau \tag{3.21}$$

Then, based on (3.20), $V^*(X)$ satisfies the tracking HJB equation

$$\begin{aligned} H(X, u^*, \nabla V^*) = X^T Q_T X + 2 \int_0^{u^*} (\lambda \tanh^{-1}(v / \lambda))^T R \, dv - \gamma V^*(X) \\ + \nabla V^{*T}(X) \, (F(X) + G(X) \, u^*(X)) = 0 \end{aligned} \tag{3.22}$$

The optimal control input for the given problem is obtained by employing the stationarity condition on the Hamiltonian (3.20). The result is

$$u^*(X) = \underset{u \in \pi(\Omega)}{\arg\min} \, [H(X, u, \nabla V^*)] = -\lambda \tanh((1 / 2\lambda) R^{-1} G^T(X) \, \nabla V^*(X)) \tag{3.23}$$

This control is within its permitted bounds $\pm\lambda$. The nonquadratic cost (3.15) for u^* is

$$U(u^*) = 2 \int_0^{u^*} (\lambda \tanh^{-1}(v / \lambda))^T R \, dv = 2 \lambda (\tanh^{-1}(u^* / \lambda))^T R \, u^* + \lambda^2 \bar{R} \ln(\underline{\mathbf{1}} \text{-} (u^* / \lambda)^2) \tag{3.24}$$

where $\underline{\mathbf{1}}$ is a column vector having all of its elements equal to one, and $\bar{R} = [r_1, ..., r_m] \in \mathbb{R}^{1 \times m}$. Putting (3.23) in (3.24) results in

$$U(u^*) = \lambda \nabla V^{*T}(X) G(x) \tanh(D^*) + \lambda^2 \bar{R} \ln(\underline{\mathbf{1}} - \tanh^2(D^*)) \tag{3.25}$$

where $D^* = (1 / 2\lambda) R^{-1} G(X)^T \nabla V^*(X)$. Substituting u^* (3.23) back into (3.22) and using (3.25), the tracking HJB equation (3.22) becomes

$$H(X, u^*, \nabla V^*) = X^T Q_T X - \gamma V^*(X) + \nabla V^{*T}(X) F(X) + \lambda^2 \bar{R} \ln(\underline{\mathbf{1}} - \tanh^2(D^*)) = 0 \tag{3.26}$$

To solve the OTCP, one solves the HJB equation (3.26) for the optimal value V^*, then the optimal control is given as a feedback $u(V^*)$ in terms of the HJB solution using (3.23).

Now a formal proof is given that the solution to the tracking HJB equation for constrained-input systems provides the optimal tracking control solution and when the discount factor is zero it locally asymptotically stabilizes the error dynamics (3.8). The following key fact is instrumental.

Lemma 3.1. For any admissible control policy $u(X)$, let $V(X) \geq 0$ be the corresponding solution to the Bellman equation (3.20). Define $u^*(X) = u(V(X))$ by (3.23) in terms of $V(X)$. Then

$$H(X,u,\nabla V) = H(X,u^*,\nabla V) + \nabla V^T(X)G(X)(u-u^*) + 2\int_{u^*}^{u} (\lambda \tanh^{-1}(v/\lambda))^T R\, dv \tag{3.27}$$

Proof. The Hamiltonian function is

$$\begin{aligned} H(X,u,\nabla V) = X^T Q_T X + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv - \gamma V(X) \\ + \nabla V^T(X)\,(F(X) + G(X)\,u(X)) \end{aligned} \tag{3.28}$$

Adding and subtracting the terms $2\int_0^{u^*} (\lambda \tanh^{-1}(v/\lambda))^T R\, dv$ and $\nabla V^T(X)\, G(X)\, u^*(X)$ to (3.28) yields

$$\begin{aligned} &H(X,u,\nabla V) = X^T Q_T X + 2\int_0^{u^*} (\lambda \tanh^{-1}(v/\lambda))^T R\, dv - \gamma V(X) \\ &+\nabla V^T(X)(F(X) + G(X)u^*(X)) + \nabla V^T(X)G(X)(u(X) - u^*(X)) + \\ &2\int_{u^*}^{u} (\lambda \tanh^{-1}(v/\lambda))^T R\, dv \end{aligned} \tag{3.29}$$

which gives (3.31) and completes the proof. □

Theorem 3.1. Consider the optimal tracking control problem for the augmented system (3.10) with performance function (3.13). Suppose that V^* is a smooth positive definite solution to the tracking HJB equation (3.26). Define control $u^* = u(V^*(X))$ as given by (3.23). Then, u^* minimizes the performance index (3.13) over all admissible controls constrained to $|u_i| \leq \lambda,\ i = 1,\ldots,m$, and the optimal value on $[0,\infty)$ is given by $V^*(X(0))$.

Moreover, when the discount factor is zero, the control input u^* makes the error dynamics (3.8) asymptotically stable.

Proof: The optimally of the HJB solution is first shown. Note that for any continuous value function $V(X)$, one can write the performance function (3.13) as

$$
\begin{aligned}
&V(X(0),u) = \int_0^\infty e^{-\gamma\tau}[X^T Q_T X + U(u)]\,d\tau + \int_0^\infty \frac{d}{dt}(e^{-\gamma\tau}V(X))\,d\tau + V(X(0)) = \\
&\int_0^\infty e^{-\gamma\tau}[X^T Q_T X + U(u)]\,d\tau + \int_0^\infty e^{-\gamma\tau}[\nabla V(X)^T(F+Gu) - \gamma V(X)]\,d\tau + \\
&V(X(0)) = \int_0^\infty e^{-\gamma\tau}\, H(X,u,\nabla V)\,d\tau + V(X(0))
\end{aligned}
\tag{3.30}
$$

Now, suppose $V(X)$ satisfies the HJB equation (3.26). Then $H(X^*, u^*, \nabla V^*) = 0$ and (3.31) yields

$$
\begin{aligned}
V(X(0),u) = &\int_0^\infty e^{-\gamma\tau}\Big(2\int_{u^*}^{u}(\lambda\tanh^{-1}(v/\lambda))^T R\,dv + \nabla V^{*T}(X)G(X)\,(u-u^*)\Big)d\tau \\
&+ V^*(X(0))
\end{aligned}
\tag{3.31}
$$

To prove that u^* is the optimal control solution and the optimal value is $V^*(X(0))$, it remains to show that the integral term in the right-hand side of the above equation is bigger than zero for all $u \neq u^*$ and attains it minimum value, i.e., zero, at $u = u^*$. That is, to show that

$$
H = 2\int_{u^*}^{u}(\lambda\tanh^{-1}(v/\lambda))^T R\,dv + \nabla V^{*T}(X)\,G(X)(u-u^*)
\tag{3.32}
$$

is bigger than or equal to zero. To show this, note that using (3.23) one has

$$
\nabla V^{*T}(X)G(X) = -2(\lambda\tanh^{-1}(v/\lambda))^T R
\tag{3.33}
$$

Substituting (3.33) in (3.32) and noting $\varphi^{-1}(.) = (\lambda\tanh^{-1}(./\lambda))^T$ yields

$$H = 2\varphi^{-1}(u^*)\,R(u^* - u) - 2\int_u^{u^*} \varphi^{-1}(v)\,R\,dv \tag{3.34}$$

As R is symmetric positive definite, one can rewrite it as $R = \Lambda\sum\Lambda$, where $\sum$ is a triangular matrix with its values being the singular values of R and Λ is an orthogonal symmetric matrix.

Substituting for R in (3.34) and applying the coordinate change $u = \Lambda^{-1}\overline{u}$, one has

$$\begin{aligned} H &= 2\varphi^{-1}(\Lambda^{-1}\overline{u}^*)\Lambda\sum(\overline{u}^* - \overline{u}) - 2\int_{\overline{u}}^{\overline{u}^*} \varphi^{-1}(\Lambda^{-1}\xi)\Lambda\sum d\xi = \\ &2\beta(\overline{u}^*)\sum(\overline{u}^* - \overline{u}) - 2\int_{\overline{u}}^{\overline{u}^*} \beta(\xi)\sum d\xi \end{aligned} \tag{3.35}$$

where $\beta(\overline{u}) = \varphi^{-1}(\Lambda^{-1}\overline{u})\Lambda$. Note that β is monotone odd because $\tanh^{-1}$ is monotonic odd. Since $\sum$ is a triangular matrix, one can decouple the transformed input vector as

$$H = 2\sum_{k=1}^{m}\sum\nolimits_{kk}\left[\beta(\overline{u}_k{}^*)(\overline{u}_k{}^* - \overline{u}_k) - \int_{\overline{u}_k}^{\overline{u}_k{}^*}\beta(\xi_k)\,d\xi_k\right] \tag{3.36}$$

where $\sum_{kk} > 0, k = 1,..,m$, since $R > 0$. To complete the proof it remains to show that the term

$$L_k = \beta(\overline{u}_k{}^*)(\overline{u}_k{}^* - \overline{u}_k) - \int_{\overline{u}_k}^{\overline{u}_k{}^*}\beta(\xi_k)\,d\xi_k \tag{3.37}$$

is bigger than zero for $u^* \neq u$ and is zero for $u^* = u$. To show this, first assume for simplicity that $\overline{u}_k$ is a scalar. The extension to the vector case is straightforward. Now assume that $\overline{u}_k{}^* > \overline{u}_k$. Then using mean value theorem for the integrals, there exists a $u_k \in (\overline{u}_k, \overline{u}_k^*)$ such that

$$\int_{\overline{u}_k}^{\overline{u}_k{}^*}\beta(\xi_k)\,d\xi_k = \beta(u_k)(\overline{u}_k{}^* - \overline{u}_k) < \beta(\overline{u}_k{}^*)(\overline{u}_k{}^* - \overline{u}_k) \tag{3.38}$$

where the inequality is obtained by the fact that β is monotone odd, and hence $\beta(u_k) < \beta(u_k^*)$. Therefore, $L_k > 0$ for $\bar{u}_k^* > \bar{u}_k$. Now suppose that $\bar{u}_k^* < \bar{u}_k$. Then, using mean value theorem for the integrals, there exists a $u_k \in (\bar{u}_k^*, \bar{u}_k)$ such that

$$\begin{aligned}\int_{\bar{u}_k}^{\bar{u}_k^*} \beta(\xi_k) d\xi_k = -\int_{\bar{u}_k^*}^{\bar{u}_k} \beta(\xi_k) d\xi_k = -\beta(u_k)(\bar{u}_k - \bar{u}_k^*) < -\beta(\bar{u}_k^*)(\bar{u}_k - \bar{u}_k^*) \\ = \beta(\bar{u}_k^*)(\bar{u}_k^* - \bar{u}_k)\end{aligned} \quad (3.39)$$

where the inequality is obtained by the fact that β is monotone odd, and hence $\beta(u_k) > \beta(u_k^*)$. Therefore $L_k > 0$ also for $\bar{u}_k^* < \bar{u}_k$. This completes the proof of the optimality.

Now the stability of the error dynamics is shown. Note that for any continuous value function $V(X)$, by differentiating $V(X)$ along the augmented system trajectories, one has

$$\frac{dV(X)}{dt} = \frac{\partial V(X)}{\partial t} + \frac{\partial V(X)^T}{\partial X}\dot{X} = \frac{\partial V(X)^T}{\partial X}(F(X) + G(X)u) \quad (3.40)$$

so that

$$H(X, u, \nabla V) = \frac{dV(X)}{dt} - \gamma V(X) + X^T Q_T X + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R \, dv \quad (3.41)$$

Suppose now that $V(X)$ satisfies the HJB equation $H(X^*, u^*, \nabla V^*) = 0$ and is positive definite. Then, substituting $u = u^*$ gives

$$\frac{dV(X)}{dt} - \gamma V(X) + X^T Q_T X + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R \, dv = 0 \quad (3.42)$$

or equivalently,

$$\frac{dV(X)}{dt} - \gamma V(X) = -X^T Q_T X - 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R \, dv \quad (3.43)$$

Multiplying $e^{-\gamma t}$ to both sides of (3.43) gives

$$\frac{d}{dt}(e^{-\gamma t}\,V(X)) = e^{-\gamma t}(-X^T Q_T X - 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\,dv) \leq 0 \tag{3.44}$$

Equation (3.36) shows that that the tracking error is bounded for the optimal solution, but its asymptotic stability cannot be concluded. However, if $\gamma = 0$ (which can be chosen only if the reference input goes to zero), LaSalle's extension can be used to show that the tracking error is locally asymptotically stable. In fact, based on LaSalle's extension, the augmented state $X = [e_d, x_d]$ goes to a region of $\mathbb{R}^{2n}$ wherein $\dot{V} = 0$. Considering that $X^T Q_T X = e_d^{\,T} Q e_d$ with $Q > 0$, $\dot{V} = 0$ only if $e_d = 0$ and $u = 0$. Since $u = 0$ also requires that $e_d = 0$, therefore, for $\gamma = 0$ the tracking error is locally asymptotically stable with Lyapunov function $V(X) > 0$. This confirms that in the limit as the discount factor goes to zero, the control input u^* makes the error dynamics (3.8) asymptotically stable. □

Note that although for $\gamma \neq 0$ (which is essential to be considered if the reference trajectory does not go to zero) only boundness of the tracking error is guaranteed for the optimal solution, one can make the tracking error as small as desired by choosing a small discount factor and/or large Q. To demonstrate this, assume that the tracking error is nonzero. Then, considering that $X^T Q_T X = e_d^{\,T} Q e_d$ with $Q > 0$, the derivative of the Lyapunov function in (3.36) becomes negative and therefore the tracking error decreases until the exponential term $e^{-\gamma t}$ becomes zero and makes the derivative of the Lyapunov function zero. After that, we can only conclude that the tracking error does not increase anymore. The larger the Q is the more the speed of decreasing the tracking error is and the smaller tracking error can be achieved. Moreover, the smaller the discount factor is the less the speed of decreasing the derivative of the Lyapunov function to zero is and the smaller tracking error can be achieved. Consequently, by choosing a smaller discount factor and/or larger Q one can make the tracking error as small as desired before the value of $e^{-\gamma t}$ becomes very small.

Remark 3.6. The use of discounted cost functions is common in optimal regulation control problems and the same conclusion can be drawn for asymptotic stability of the system state in the optimal regulator problem, as is drawn here for asymptotic stability of the tracking error in the OTCP. However, the discount factor is a design parameter and as is shown in optimal regulation control problems in the literature, it can be chosen small enough to make sure the system state goes to a very small region around zero. Simulation results in confirm this conclusion for the OTCP.

The tracking HJB equation (3.26) is a nonlinear partial differential equation which is extremely difficult to solve. In this section, two iterative offline policy iteration (PI) algorithms are presented for solving this equation. An IRL based offline PI algorithm is given which is a basis for our online IRL algorithm presented in the next section.

Note that the tracking HJB equation (3.26) is nonlinear in the value function derivative ∇V^*, while the tracking Bellman equation (2.19) is linear in the cost function derivative ∇V. Therefore, finding the value of a fixed control policy by solving (3.19) is easier than finding the optimal value function by solving (3.26). This is the motivation of introducing an iterative policy iteration (PI) algorithm for approximating the tracking HJB solution. The PI algorithm performs the following sequence of two-step iterations as follows to find the optimal control policy.

Algorithm 3. 1. Offline PI algorithm

Policy evaluation: Given a control input $u^i(X)$, find $V^i(X)$ using the following Bellman equation

$$X^T Q_T X + 2\int_0^{u^i} (\lambda \tanh^{-1}(v/\lambda))^T R\,dv - \gamma V^i(X) + \nabla V^{iT}(X)(F(X) + G(X)u^i) = 0 \tag{3.45}$$

Policy improvement: Update the control policy using

$$u^{i+1}(X) = -\lambda \tanh(\frac{1}{2\lambda} R^{-1} G^T(X)\, \nabla V^i(X)) \tag{3.46}$$

Algorithm 3.1 is an extension of the offline PI algorithm in [3] to the optimal tracking problem. The following theorem shows that this algorithm converges to the optimal solution of the HJB equation (3.26).

Theorem 3.2. If $u^0 \in \pi(\Omega)$, then $u^i \in \pi(\Omega),\ \forall i \geq 1$. Moreover, u^i converges to u^* and V^i converges to V^* uniformly on Ω.

Proof: See [3], [75] for the same proof. □

The tracking Bellman equation (3.45) requires complete knowledge of the systems dynamics. In order to find an equivalent formulation of the tracking Bellman equation that does not involve the dynamics, we use the IRL idea for optimal regulation problem. Note that for any integral reinforcement interval $T > 0$, the value function (3.13) satisfies

$$V(X(t-T)) = \int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \left[X(\tau)^T Q_T\, X(\tau) + U(u(\tau))\right] d\tau + e^{-\gamma T} V(X(t)) \tag{3.47}$$

This IRL form of the tracking Bellman equation does not involve the system dynamics.

Lemma 3.2. The IRL tracking Bellman equation (3.47) and the tracking Bellman equation (3.19) are equivalent and have the same positive semi-definite solution for the value function.

Proof. See [75] and [112] for the same proof. □

Using the IRL tracking Bellman equation (3.47), the following IRL-based PI algorithm can be used to solve the tracking HJB equation (3.26) using only partial knowledge about the system dynamics.

Algorithm 3.2. Offline IRL algorithm

Policy evaluation: Given a control input $u^i(X)$, find $V^i(X)$ using the tracking Bellman equation

$$V^i(X(t-T)) = \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\Big[X(\tau)^T Q_T\, X(\tau) + U(u(\tau))\Big]d\tau + e^{-\gamma T}V^i(X(t)) \tag{3.48}$$

Policy improvement: Update the control policy using

$$u^{i+1}(X) = -\lambda \tanh\big(\frac{1}{2\lambda} R^{-1}G^T(X)\,\nabla V^i(X)\big) \tag{3.49}$$

3.5. Online Actor-Critic for Solving the Tracking HJB Equation Using the IRL Technique

In this section, an online solution to the tracking HJB equation (3.26) is presented which only requires partial knowledge about the system dynamics. The learning structure uses the value function approximation [29] with two NNs, namely an actor and a critic. Instead of sequentially updating the critic and actor NNs, as in Algorithm 3.2, both are updated simultaneously in real time. This is called synchronous online PI.

3.5.1. Critic NN and Value Function Approximation

Assuming the value function is a smooth function, according to the Weierstrass high-order approximation theorem [29], there exists a single-layer neural network (NN) such that the solution $V(X)$ and its gradient $\nabla V(X)$ can be uniformly approximated as

$$V(X) = W_1^T \phi(X) + \varepsilon_v(X) \tag{3.50}$$

$$\nabla V(X) = \nabla \phi(X)^T W_1 + \nabla \varepsilon_v(X) \tag{3.51}$$

where $\phi(X) \in \mathbb{R}^l$ provides a suitable basis function vector, $\varepsilon_v(X)$ is the approximation error, $W_1 \in \mathbb{R}^l$ is a constant parameter vector and l is the number of neurons. Equation (3.50) defines a critic NN with weights W_1. It is know that the NN approximation error and its gradient are bounded over the compact set Ω, i.e. $\left\|\varepsilon_v(X)\right\| \le b_\varepsilon$ and $\left\|\nabla\varepsilon_v(X)\right\| \le b_{\varepsilon x}$.

Assumption 3.4 . The critic NN activation functions and their gradients are bounded, i.e. $\left\|\phi(X)\right\| \le b_\phi$ and $\left\|\nabla\phi(X)\right\| \le b_{\phi x}$.

The critic NN (3.50) is used to approximate the value function related to the IRL tracking Bellman equation (3.47). Using the value approximation (3.50) in the tracking IRL, the Bellman equation (3.47) yields

$$\varepsilon_B(t) \equiv \int_{t-T}^{t} e^{-\gamma(\tau-t+T)} \left[X(\tau)^T Q_T X(\tau) + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv \right] d\tau + W_1^T \Delta\phi(X(t)) \tag{3.52}$$

where

$$\Delta\phi(X(t)) = e^{-\gamma T}\phi(X(t)) - \phi(X(t-T)) \tag{3.53}$$

and ε_B is the tracking Bellman equation error due to the NN approximation error. Under Assumption 3.4, this approximation error is bounded on the compact set Ω. That is, there exists a constant bound $\varepsilon_{\max}$ for ε_B such that $\sup\limits_{\forall t}\left\|\varepsilon_B\right\| \le \varepsilon_{\max}$.

The tuning and convergence of the critic NN weights for a fixed control policy are now presented. As the ideal critic NN weights vector W_1 which provides the best approximate solution to the tracking Bellman (3.52) is unknown, it is approximated in real time as

$$\hat{V}(X) = \hat{W}_1^T \phi(X) \tag{3.54}$$

where $\hat{W}_1$ is the current estimation of W_1. Therefore, the approximate IRL tracking Bellman equation becomes

$$e_B(t) = \int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \left[X(\tau)^T Q_T\, X(\tau) + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv \right] d\tau + \hat{W}_1^T \Delta\phi(X(t)) \tag{3.55}$$

Equation (3.55) can be written as

$$e_B(t) = \hat{W}_1(t)^T\, \Delta\phi(X(t)) + p(t) \tag{3.56}$$

where

$$p(t) = \int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \left[X(\tau)^T Q_T\, X(\tau) + 2\int_0^u (\lambda \tanh^{-1}(v/\lambda))^T R\, dv \right] d\tau \tag{3.57}$$

is the integral reinforcement reward. The tracking Bellman error e_B in equations (3.55) and (3.56) is the continuous-time counterpart of the temporal difference (TD). The problem of finding the value function is now converted to adjusting the critic NN weights such that the TD error e_B is minimized. Consider the objective function

$$E_B = \frac{1}{2} e_B^{\,2} \tag{3.58}$$

From (3.55) and using the chain rule, the gradient descent algorithm for E_B is given by

$$\dot{\hat{W}}_1 = \frac{-\alpha_1}{(1+\Delta\phi^T\Delta\phi)^2}\frac{\partial E_B}{\partial \hat{W}_1} = -\frac{-\alpha_1\Delta\phi}{(1+\Delta\phi^T\Delta\phi)^2}e_B \tag{3.59}$$

where $\alpha_1 > 0$ is the learning rate and $(1+\Delta\phi^T\Delta\phi)^2$ is used for normalization. Note that the square of the denominator, i.e., $(1+\Delta\phi^T\Delta\phi)^2$, is used in (3.59) for normalization to assure the stability of the critic weights error $\tilde{W}_1$. Define

$$\Delta\bar{\phi} = \Delta\phi \big/ (1+\Delta\phi^T\Delta\phi) \tag{3.60}$$

The proof of the convergence of the critic NN weights is shown in the following theorem.

Theorem 3.3. Let u be any admissible bounded control policy and consider the adaptive law (3.59) for tuning the critic NN weights. If $\Delta\bar{\phi}$ in (3.60) is PE [45], i.e. if there exist $\gamma_1 > 0$ and $\gamma_2 > 0$ such that $\forall t > 0$

$$\gamma_1 I \le \int_t^{t+T_1} \Delta\bar{\phi}(\tau)\Delta\bar{\phi}^T(\tau)\,d\tau \le \gamma_2 I, \tag{3.61}$$

then,

(a): For $\varepsilon_B(t) = 0$ (no reconstruction error), the critic weight estimation error converges to zero exponentially fast.

(b): For bounded reconstruction error, i.e., $\left\|\varepsilon_B(t)\right\| < \varepsilon_{\max}$, the critic weight estimation error converges exponentially fast to a residual set.

Proof: Using the IRL tracking Bellman equation (3.52) one has

$$\int_{t-T}^{t} e^{-\gamma(\tau-t)}\left[2\int_0^u (\lambda\tanh^{-1}(v/\lambda))^T R\,dv + X(\tau)^T Q_T X(\tau)\right]d\tau = -W_1^T\Delta\phi(X(t)) + \varepsilon_B(t) \tag{3.62}$$

Substituting (3.62) in (3.55), the tracking Bellman equation error becomes

$$e_B(t) = \tilde{W}_1^T(t)\,\Delta\phi(t) + \varepsilon_B(t) \tag{3.63}$$

where $\tilde{W}_1 = W_1 - \hat{W}_1$ is the critic weights estimation error.

Using (3.63) in (3.59) and denoting $m = 1 + \Delta\phi^T \Delta\phi$, the critic weights estimation error dynamics becomes

$$\dot{\tilde{W}}_1(t) = -\alpha_1 \Delta\bar{\phi}(t)\Delta\bar{\phi}(t)^T \tilde{W}_1(t) + \alpha_1 \frac{\Delta\bar{\phi}(t)}{m(t)} \varepsilon_B(t) \tag{3.64}$$

This estimation error is the same as the critic weight estimation error obtained in [106] and the reminder of the proof is identical to the proof of Theorem 3.1 in [106]. □

Remark 3.7. The critic estimation error equation (3.64) implies that $\Delta\bar{\phi}^T \tilde{W}_1$ is bounded. However, in general the boundness of $\Delta\bar{\phi}^T \tilde{W}_1$ does not imply the boundness of $\tilde{W}_1$. Theorem 3.3 shows that if the PE condition (3.61) is satisfied, then the boundness of $\Delta\bar{\phi}^T \tilde{W}_1$ implies the boundness of the state $\tilde{W}_1$. We shall use this property in the proof of Theorem 3.4.

3.5.2. Synchronous Actor-Critic based IRL Algorithm to learn the solution to the OTCP

In this section, an online IRL algorithm is given which involves simultaneous or synchronous tuning of the actor and critic NNs to find the optimal value function and control policy related to the OTCP, adaptively.

Assume that the optimal value function solution to the tracking HJB equation is approximated by the critic NN in (3.50). Then, using (3.51) in (3.23), the optimal policy is obtained by

$$u = -\lambda \tanh\left(\frac{1}{2\lambda} R^{-1} G^T (\nabla\phi^T W_1 + \nabla\varepsilon_v{}^T) \right) \tag{3.65}$$

To see the effect of the error $\nabla\varepsilon_v$ on the tracking HJB equation, note that using integration by parts we have

$$\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\dot{\phi}\,d\tau = \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\nabla\phi(F+Gu)\,d\tau = \Delta\phi(X) + \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\phi(X)\,d\tau \tag{3.66}$$

or equivalently

$$\Delta\phi(X) = \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\nabla\phi(X)(F+Gu)\,d\tau - \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\phi(X)\,d\tau \tag{3.67}$$

Also, note that $U(u)$ in (3.24) for the optimal control input given by (3.65) becomes

$$\begin{aligned} &U(u) = 2\int_0^u (\lambda\tanh^{-1}(v/\lambda))^T R\,dv = W_1^T\nabla\phi F\,u + \\ &\lambda^2\bar{R}\ln(\underline{\mathbf{1}} - \tanh^2(D + 0.5\lambda R^{-1}G^T\nabla\varepsilon_v) \end{aligned} \tag{3.68}$$

Using (3.67) and (3.68) for the third and second terms of (3.52), respectively, the following tracking HJB equation is given

$$\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^TQ_TX - \gamma W_1^T\phi + W_1^T\,\nabla\phi\,F + \lambda^2\bar{R}\,\ln(\underline{\mathbf{1}} - \tanh^2(D)) + \varepsilon_{HJB})\,d\tau = 0 \tag{3.69}$$

where $D = (1/2\lambda)\,R^{-1}G^T\,\nabla\phi^T\,W_1$ and ε_{HJB}, i.e., the HJB approximation error due to the function approximation error, is

$$\begin{aligned} \varepsilon_{HJB} = \int_{t-T}^{t} & e^{-\gamma(\tau-t+T)}(\nabla\varepsilon_v{}^TF + \lambda^2R\ln(1-\tanh^2(D+0.5\lambda R^{-1}G^T\nabla\varepsilon_v) \\ & - \lambda^2\bar{R}\ln(\underline{\mathbf{1}} - \tanh^2(D)) - \gamma\varepsilon_v)d\tau \end{aligned} \tag{3.70}$$

Since the NN approximation error is bounded, there exists a constant error bound ε_h, so that $\sup\|\varepsilon_{HJB}\| \le \varepsilon_h$. We should note that the choice of the NN structure to make the error bound ε_h arbitrary small is commonly carried out by computer simulation in the literature. We assume here that the NN structure is specified by the designer, and the only unknowns are the NN weights.

To approximate the solution to the tracking HJB equation (3.69), the critic and actor NNs are employed. The critic NN given by (3.54) is used to approximate the unknown approximate optimal value function. Assuming that $\hat{W}_1$ is the current estimation for the optimal critic NN weights W_1, then using (3.55) the policy update law can be obtained by

$$u_1 = -\lambda \tanh((1/2\lambda) R^{-1} G^T \nabla\phi^T \hat{W}_1) \tag{3.71}$$

However, this policy update law does not guarantee the stability of the closed-loop system. It is necessary to use a second neural network $\hat{W}_2^T \nabla\phi$ for the actor because the control input must not only solve the stationarity condition (3.23), but also guarantee system stability while converging to the optimal solution. This is seen in the Lyapunov proof of Theorem 3.4. Hence, to assure stability in a Lyapunov sense, the following actor NN is employed.

$$\hat{u}_1 = -\lambda \tanh((1/2\lambda) R^{-1} G^T \nabla\phi^T \hat{W}_2) \tag{3.72}$$

where $\hat{W}_2$ is the actor NN weights vector and it is considered as the current estimated value of W_1. Define the actor NN estimation error as

$$\tilde{W}_2 = W_1 - \hat{W}_2 \tag{3.73}$$

Note that using the actor $\hat{u}_1$ in (3.72), the IRL Bellman equation error is now given by

$$\int_{t-T}^{t} e^{-\gamma(\tau-t+T)} \left[X(\tau)^T Q_T \, X(\tau) + \hat{U} \right] d\tau + \hat{W}_1^T \Delta\phi(X(t)) = \hat{e}_B(t) \tag{3.74}$$

where

$$\hat{U} = 2\int_0^{\hat{u}_1} (\lambda \tanh^{-1}(v/\lambda))^T R \, dv \tag{3.75}$$

Then, the critic update law (3.59) becomes

$$\dot{\hat{W}}_1 = -\frac{-\alpha_1 \Delta\phi}{(1+\Delta\phi^T \Delta\phi)^2} \hat{e}_B \tag{3.76}$$

Define the error e_u as the difference between the control input $\hat{u}_1$ (3.72) applied to the system and the control input $\hat{u}$ (3.71) as an approximation of the optimal control input given by (with V^* approximated by (3.54). That is,

$$e_u = \hat{u}_1 - u_1 = \lambda\left(\tanh\left(\frac{1}{2\lambda} R^{-1} G^T \nabla\phi^T \hat{W}_2\right) - \tanh\left(\frac{1}{2\lambda} R^{-1} G^T \nabla\phi^T \hat{W}_1\right)\right) \tag{3.77}$$

The objective function to be minimized by the action NN is now defined as

$$E_u = e_u^{\ T} R e_u \tag{3.78}$$

Then, the gradient-descent update law for the actor NN weights becomes

$$\dot{\hat{W}}_2 = -\alpha_2(\nabla\phi G e_u + \nabla\phi G \tanh^2(\hat{D}) e_u + Y\hat{W}_2) \tag{3.79}$$

where

$$\hat{D} = \frac{1}{2\lambda} R^{-1} G^T \nabla\phi^T \hat{W}_2, \tag{3.80}$$

$Y > 0$ is a design parameter and the last term of (3.79) is added to assure stability.

Before presenting our main theorem, note that based on Assumption 3.2 and the boundness of the command generator dynamics h_d, for the drift dynamics of the augment system F one has

$$\|F(X)\| \le b_{F1} \|e_d\| + b_{F2} \tag{3.81}$$

for some b_{F1} and b_{F1}.

Theorem 3.4. Given the dynamical system (3.1) and the command generator (3.4), let the tracking control law be given by the actor NN (3.72). Let the update laws for tuning the critic and actor NNs be provided by (3.76) and (3.79), respectively. Let Assumptions 3.1-3.4 hold and $\Delta\bar{\phi}$ in (3.60) be persistently exciting. Then there exists a T_0 defined by (A.25) such that for the integral reinforcement interval $T < T_0$ the tracking error e_d in (3.2), the critic NN error $\tilde{W}_1$, and the actor NN error $\tilde{W}_2$ in (3.73) are UUB [54].

Proof. See Appendix.

Remark 3.8. The stability analysis in the proof of Theorem 3.4 differs from the stability proof presented in [108] from at least two different perspectives. First, the actor update law in the mentioned papers is derived entirely by the stability analysis whereas our proposed actor update law is based on the minimization of the error between the actor neural network and the approximate optimal control input. Moreover, in this chapter the optimal tracking problem is considered, not the optimal regulation problem, and the tracking HJB equation has an additional term depending on the discount factor comparing to the regulation HJB equation considered in the mentioned papers.

Remark 3.9. The proof of Theorem 3.4 shows that the integral reinforcement learning time interval T cannot be too big. Moreover, based on the proof, one can conclude that the bigger the reinforcement interval T is, the bigger the parameter Y in learning rule (3.79) should be chosen to assure stability.

3.6. Simulation results

In this section, a simulation example is given to show the effectiveness of the proposed method. Fig 3.1 shows a spring, mass, damper system.

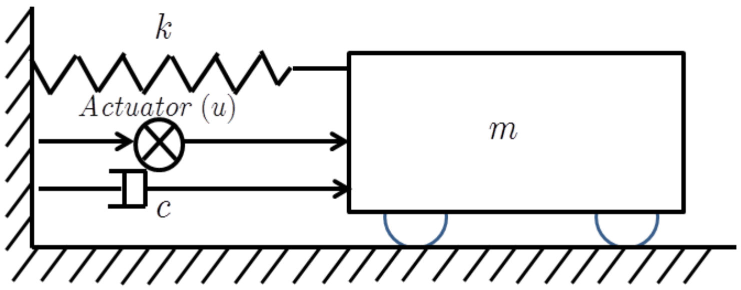

Fig. 3.1. Mass, spring and damper system.

The simulation results constitute of two parts. In the first part, the spring and damper are considered to be linear and the actuator bound is chosen large enough to make sure the input control does not exceed this bound, and it is shown that how the proposed algorithm converges to the optimal solution for a linear system in the absence of the input constraints. Note that there are no known solutions to optimal control problems for linear systems with input constraints to compare our results to. In the second part, the spring is considered to be nonlinear and the actuator saturation is also considered to show the effectiveness of the proposed method for control of nonlinear systems in the presence of the input constraints.

3.6.1. Linear system without actuator saturation

In this subsection, the results of the proposed method are compared to the results of the standard solution given in Section 3.2, and also it is shown that the proposed method converges to the optimal solution in the absence of the control bounds. To this end, the actuator bounds are chosen large enough to make sure the input control does not exceed these bounds.

Assuming that both spring and damper are linear, the spring-mass-damper system is described by the following dynamics

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}u(t)\end{aligned} \tag{3.82}$$

where $y = x_1$, x_1 and x_2 are the position and velocity, m is the mass of the object, k is the stiffness constant of the spring and c is the damping. The true parameters are set as $m = 1kg$, $c = 0.5\,N.s\big/m$ and $k = 5\,N\big/m$. Note that in our control design, only the input dynamics is needed to be known, which is given by m.

The desired trajectories for x_1 and x_2 are considered as

$$x_{d1}(t) = 0.5\sin(\sqrt{5}t) \tag{3.83}$$

and

$$x_{d2}(t) = 0.5\sqrt{5}\cos(\sqrt{5}t) \tag{3.84}$$

which are given by using the following command generator dynamics

$$\dot{x}_d = \begin{bmatrix} 0 & 1 \\ -5 & 0 \end{bmatrix} x_d \tag{3.85}$$

with initial condition $x_d(0) = [0.5, 0.5]$. Therefore, the augmented system (3.10) becomes

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5 & -0.5 & 3 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -5 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u \equiv TX + B_1 u \tag{3.86}$$

where $X = [X_1, X_2, X_3, X_4] = [e_{d1}, e_{d2}, x_{d1}, x_{d2}]$.

The input saturation limit is considered as 5*N*, i.e., $|u| \le 5$. The nonquadratic performance index is chosen as (3.13) with $R = 1$, $Q = 10\mathrm{I}$ and $\gamma = 0.1$.

As actuator saturation does not occur, the optimal value function should be close to the value function of the linear quadratic tracking (LQT) problem. By the LQT problem, we mean the optimal tracking problem for linear systems with quadratic performance functions. In fact, for the augmented system (3.86) with a quadratic performance function $U(u) = u^T R u$ in (3.13), the value function is in the quadratic form of $V(X) = X^T P X$ and therefore the HJB equation (3.22) converts to the following ARE

$$0 = T^T P + P T - \gamma P + P B_1 R^{-1} B_1^T P + Q_T \tag{3.87}$$

Efficient numerical methods exist to find the solution to this ARE which we can compare our results to.

We now simulate our proposed method as in Theorem 3.4. As we expect that optimal critic is quadratic in the system in the absence of the control bounds, the critic NN is chosen as

$$V(x) = W^T \phi(x) \tag{3.88}$$

where

$$W = [W_1, ..., W_{10}]^T,\ \phi(X) = [X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, X_2X_4, X_3^2, X_3X_4, X_4^2]^T \tag{3.89}$$

The reinforcement interval T is selected as 0.1. A small probing noise is added to the control input to excite the system states. Fig. 3.2 shows the convergence of the critic parameters which converges to

$$W = [17.94,\ 0.77,\ -2.01,\ -0.29,\ 2.86,\ 0.07,\ -0.59,\ 9.86, -0.08,\ 1.84]$$

The optimal control solution (3.23) then becomes

$$u = -5\tanh(0.155 e_{d1} + 0.577 e_{d2} + 0.014 x_{d1} - 0.112 x_{d2})$$

Note that the optimal critic weights obtained by solving the ARE is

$$W^* = [18.05,\ 0.77,\ -1.98,\ -0.34,\ 2.88,\ 0.08,\ -0.56,\ 9.77, -0.08,\ 1.87]$$

which are the components of the ARE solution matrix P in (3.87) and confirms the convergence of our algorithm close to the optimal control solution.

For the standard solution, the steady-state part of control input using (3.6) and the system and command generator dynamics becomes

$$u_d = [0.25, 0.25]\, x_d(t)$$

The optimal feedback part of the control input is

$$u_e = [-0.50,\ -0.25]\, e_d(t)$$

Thus, the optimal control is given by

$$u = -0.50 e_{d1} - 0.25 e_{d2} + 0.25 x_{d1} + 0.25 x_{d2}$$

Figs 3.3-3.8 show the system state and the control input for both the proposed and the standard methods, starting the system from a specific initial condition. From these figures, it can be concluded that although in contrast to the standard method, the proposed method does not require the system drift dynamics, its transient response is better than the standard method.

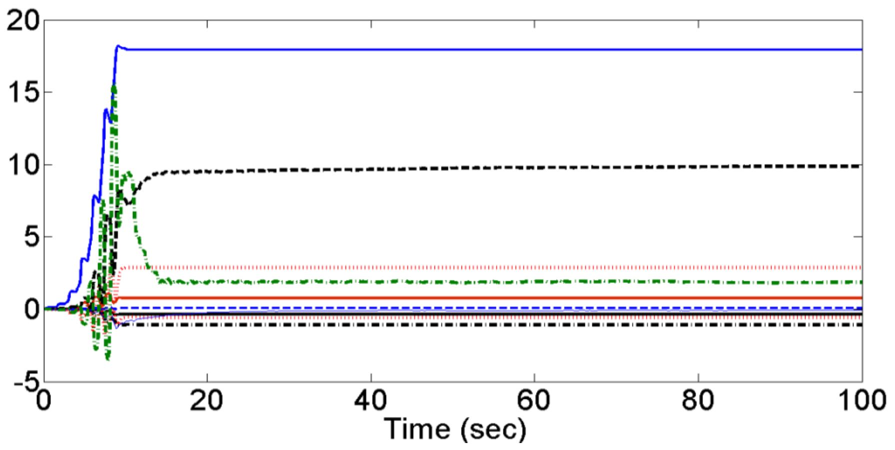

Fig. 3.2. Convergence of the critic NN weights.

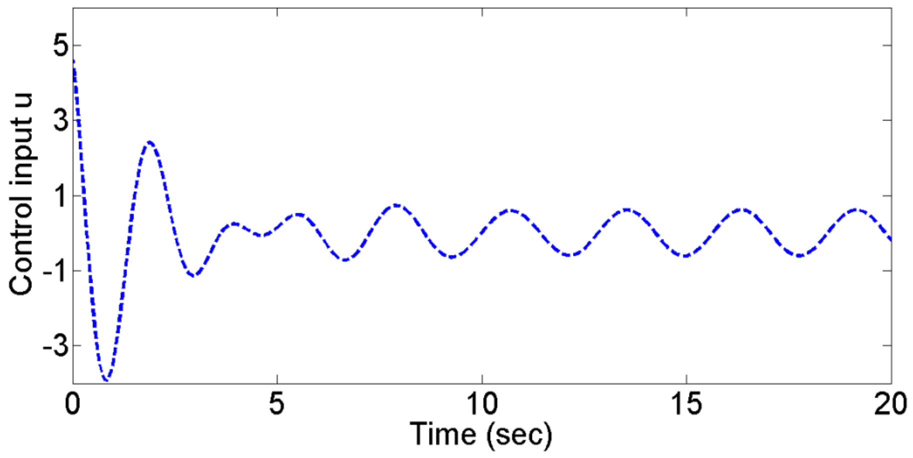

Fig. 3.3. Control input for the standard method.

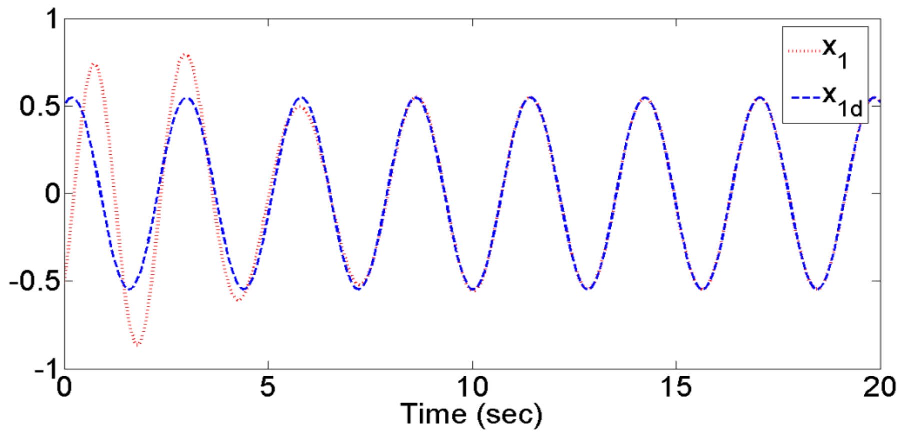

Fig. 3.4. First system state and desired trajectory for the standard method.

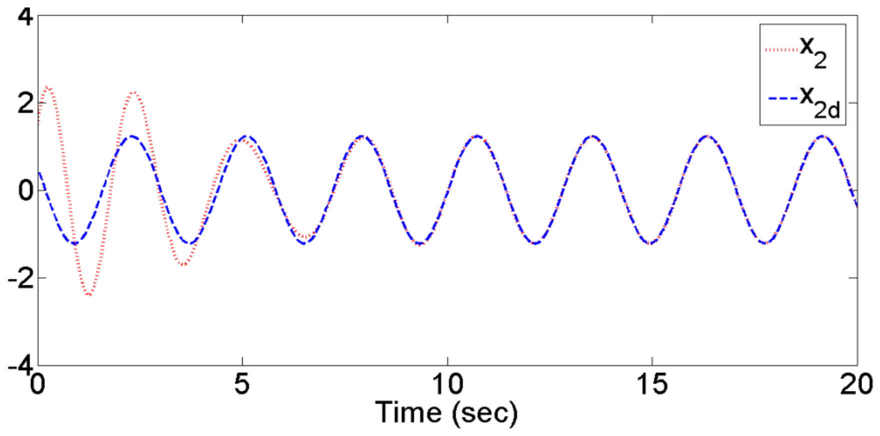

Fig. 3.5. Second system state and desired trajectory for the standard method.

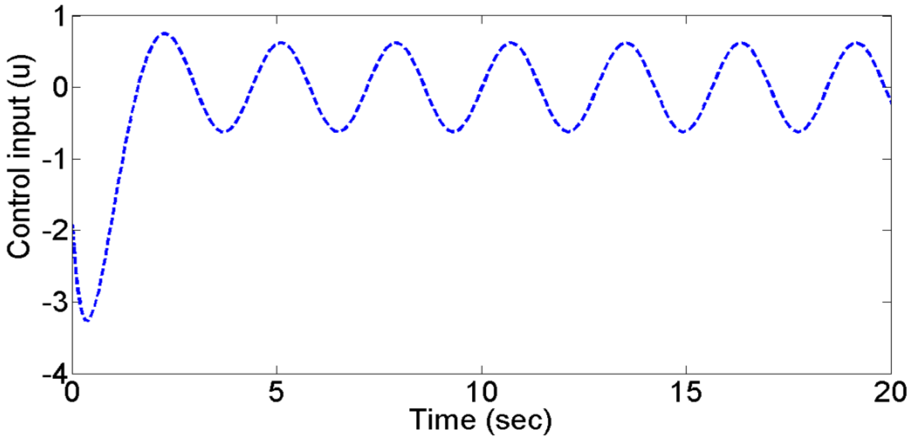

Fig. 3.6. Control input for the proposed method.

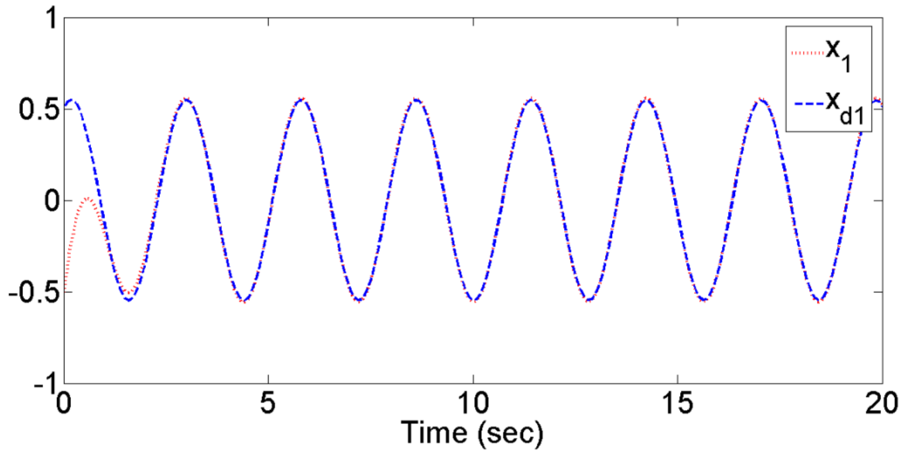

Fig. 3.7. First system state and desired trajectory for the proposed method.

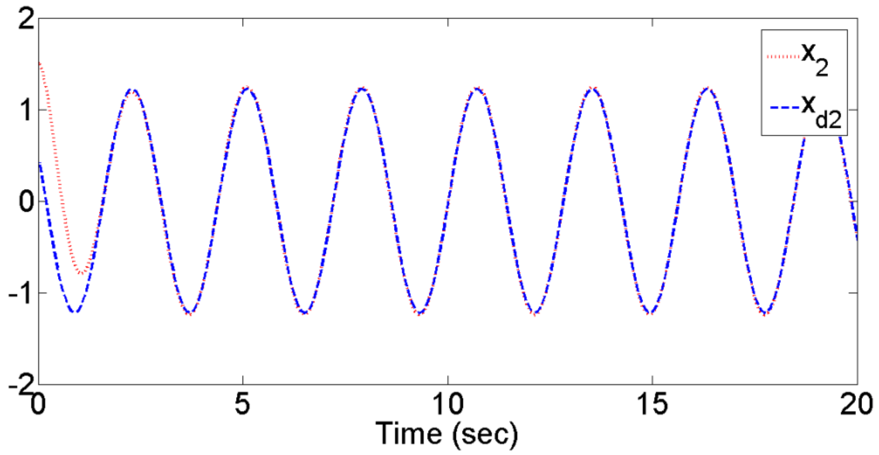

Fig. 3.8. Second system state and desired trajectory for the proposed method.

Remark 3.10. According to Theorem 3.4, the error bounds for optimal control solution depend on the NN approximation errors, the HJB residual, and the unknown critic NN weights. If the number of NN hidden layers is chosen appropriately, which is fulfilled for the linear system provided here, all of these go to zero except for the unknown critic NN weights. However, these bounds are in fact conservative and the simulation results show that the value function and the optimal control solution are closely identified.

3.6.2. Nonlinear system and considering the actuator bound

In this subsection, it is considered that the spring is nonlinear with the nonlinearity $k(x) = -x^3$ and therefore the system dynamics becomes

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -{x_1}^3 - 0.5x_2 + u(t) \end{aligned} \tag{3.90}$$

Now suppose that the control bound is $|\, u \,| \leq 0.25$.

To find the optimal solution using the proposed method, the critic NN is chosen as a power series neural network with 45 activation functions containing powers of the state variable of the augmented system up to order four. That is, the critic is chosen as (3.88) with weights and activation functions as

$$\begin{aligned} &W = [W_1, \ldots, W_{45}]^T,\ \ \phi(X) = [X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, X_2X_4, X_3^2, X_3X_4, \\ &X_4^2, X_1^4, X_1^3X_2, X_1^3X_3, X_1^3X_4, X_1^2X_2^2, X_1^2X_2X_3, X_1^2X_2X_4, X_1^2X_3^2, X_1^2X_3X_4, X_1^2X_4^2, X_1X_2^3, \\ &X_1X_2^2X_3, X_1X_2^2X_4, X_1X_2X_3^2, X_1X_2X_3X_4, X_1X_2X_4^2, X_1X_3^3, X_1X_3^2X_4, X_1X_3X_4^2, X_1X_4^3, \\ &X_2^4, X_2^3X_3, X_2^3X_4, X_2^2X_3^2, X_2^2X_3X_4, X_2^2X_4^2, X_2X_3^3, X_2X_3^2X_4, X_2X_3X_4^2, X_2X_4^3, X_3^4, X_3^3X_4, \\ &X_3^2X_4^2, X_3X_4^3, X_4^4]^T \end{aligned} \tag{3.91}$$

The reinforcement interval T is selected as 0.1. As no verifiable method exists to ensure PE in nonlinear systems, a small exploratory signal consisting of sinusoids of varying frequencies, i.e., $n(t) = 0.3\,sin(8t)^2 cos(2t) + 0.3\,sin(20t)^4 cos(7t)$, is added to the control input to excite the system states and ensure the PE qualitatively. The critic weights vector finally converges to

$W =$ [9.04, 3.95, -1.20, -1.64, 2.41, 0.71, -1.06, 14.28, 0.38, 2.93, -2.97, -0.75, 4.60, -2.40, -3.33, 1.79, 2.18, 3.11, 0.69, -2.45, -2.23, 1.70, 2.02, 0.94, 0.43, 1.21, -0.47, -0.75, 0.54, 1.31, 0.03, 1.70, 0.81, 0.88, -0.02, -0.76, 0.84, -0.15, -3.14, -0.83, 4.11, 0.29, 0.86, -0.88, 0.07]. Figs 3.9-3.11 show the performance of the proposed method.

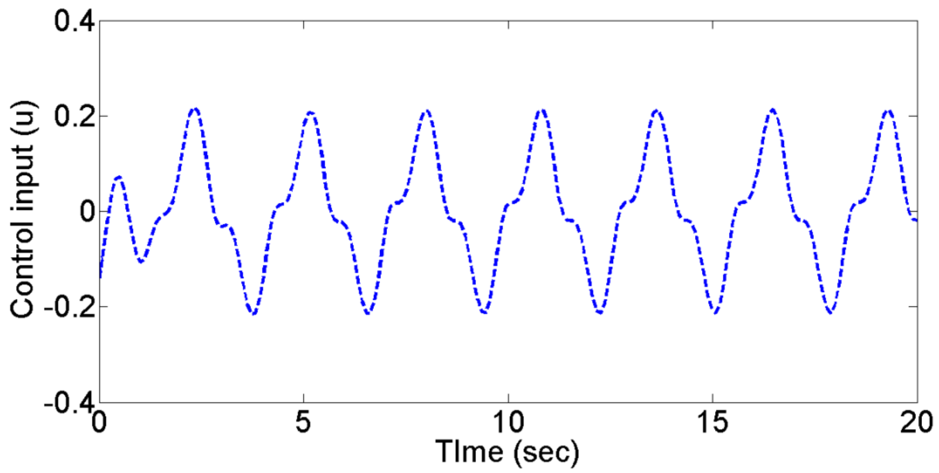

Fig. 3.9. Control input while considering the actuator saturation.

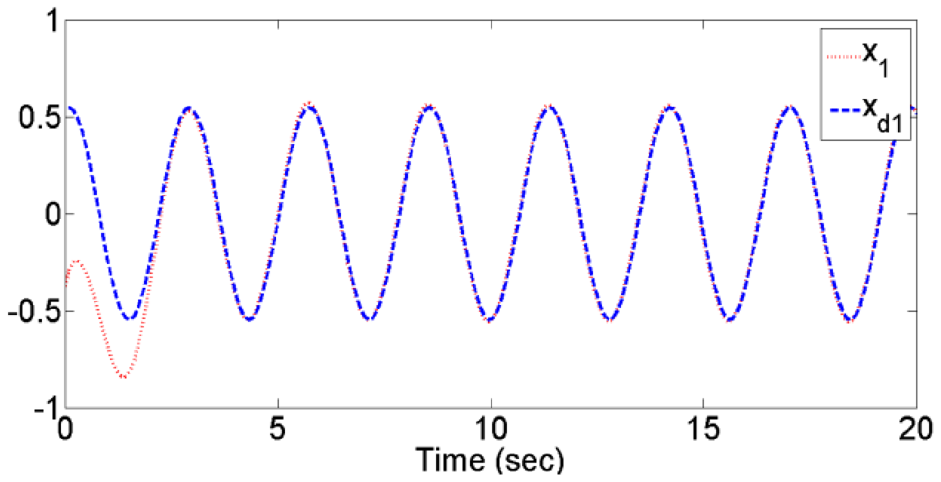

Fig. 3.10. The system state x_1 versus x_{1d} while considering the actuator saturation.

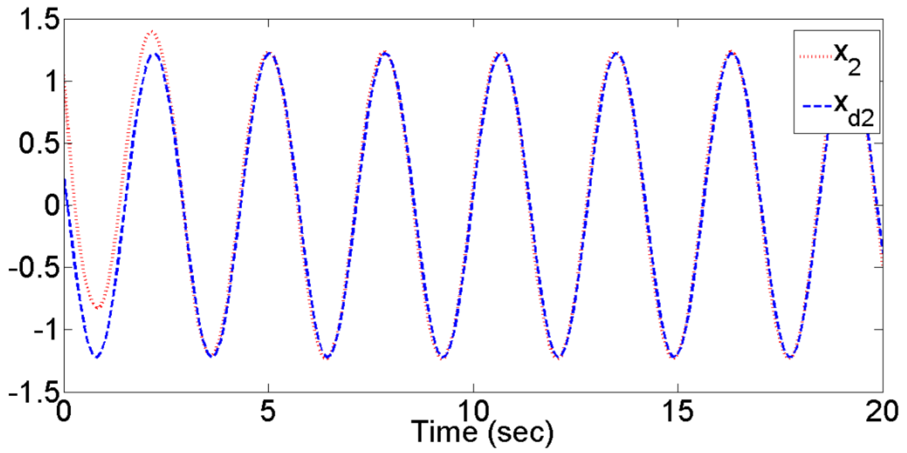

Fig. 3.11. The system state x_2 versus x_{2d} while considering the actuator saturation

3.7. Conclusion

A new formulation of the optimal tracking control problem was presented in this chapter. A tracking constrained HJB equation was derived where both feedback and feedforward parts of the bounded optimal control input were obtained simultaneously by solving this HJB equation. An online integral reinforcement learning algorithm was presented to find the solution to the tracking HJB equation for partially-unknown constrained-input systems. The proposed method did not require any preceding identification procedure. The stability of the whole system and convergence to a near-optimal control solution were shown.

Chapter 4

OPTIMAL H_∞ TRACKING CONTROL OF UNKNOWN SYSTEMS

4.1. Introduction

This chapter concerns with solving the problem of H_∞ tracking control of nonlinear continuous-time systems with completely unknown dynamics.

The H_∞ optimal control has been extensively used in the effort to design feedback controllers to reduce the effect of disturbances on the system performance. The study and design of H_∞ optimal controllers, [4], [13], [26], [46], [47], [111], [136] were considered after the H_∞ optimal control framework was initiated by Zames [1]. Significant insight into the design of H_∞ control problems has been provided, after it was formulated as a min-max two-player zero-sum game problem [14]. The optimal control in such a scenario is equivalent to finding the Nash equilibrium of the corresponding two-player zero-sum game [4], [21], which results in solving the so-called Hamilton-Jacobi-Isaacs (HJI) equation. For linear systems with quadratic performance function, the HJI equation reduces to the game ARE.

Existing work on H_∞ optimal control has mostly concentrated on designing regulator control systems. The objective in the regulator problem is to drive the states of the system to zero. In practice, however, it is often required to force the states or outputs of the system to track a reference (desired) trajectory. Despite its important, few results considered the H_∞ optimal tracking control problem. Existing solutions to the H_∞ optimal tracking are composed of two steps. In the first step, a feedforward control input is design by either dynamic inversion method [8], [103] or by solving Francis–Byrnes–Isidori (FBI) equations [46] to guarantee perfect tracking. In the second step, a feedback control input is designed by solving an HJI equation to stabilize the tracking error dynamics and satisfy a bounded L_2-gain condition. These methods are suboptimal as they ignore the cost of the feedforward control input in the performance function. This may result in a large control effort, especially if the initial tracking error is large. Moreover, in these methods, procedures for computing the feedback and feedforward terms are based on offline solution methods which must be done in a noncausal manner and require complete knowledge of the system dynamics.

During the last few years, strong connections between reinforcement learning (RL) and optimal control have prompted a major effort towards developing RL algorithms to learn the solution to the HJI equation arising in the H_∞ optimal regulation problem. Most of the available RL algorithms for learning the HJI solution are based on the policy iteration (PI) method. In this method, the HJI equation, which is a nonlinear partial differential equation (PDE), is solved successively by breaking it into a sequence of linear PDEs that are considerably easier to handle. Abu-Khalaf et al. [2] used an offline PI algorithm along with NN approximators to approximate solution to the HIJ equation. Online synchronous PI algorithms were proposed in [107], [108], [107][142] to find an approximate solution to the HJI equation. Computationally efficient simultaneous policy update algorithm for both linear and nonlinear systems were presented in [130], [131]. All of these mentioned methods require complete knowledge of the system dynamics. Moreover, in these methods, the disturbance needs to be adjusted which is not practical in most systems as the disturbance is not under our control. In [114], the authors used the integral reinforcement learning (IRL) [112], to learn the solution to the HJI equation using only partial knowledge about the system dynamics. However, this method still requires partial knowledge of the system dynamics and an adjustable disturbance input.

In [70], the authors proposed a PI algorithm for solving the game ARE equation without requiring knowledge of the system dynamics. However, their method is limited to linear systems and requires the disturbance be adjustable. In [78], the authors proposed an off-policy PI algorithm to learn the solution to the HJI equation. In the off-policy RL algorithm, the system data, which is used to learn the HJI solution, can be generated with arbitrary policies rather than the evaluating policy. Their method does not require an adjustable disturbance input. However, it requires partial knowledge of the system dynamics.

Existing above mentioned PI methods for solving the H_∞ optimal regulation of nonlinear systems either require at least partial knowledge of the system dynamics, or require the disturbance input be adjustable, or both. Moreover, while significant progress has been achieved in the use of PI algorithms for the design of the H_∞ optimal controllers, these algorithms are limited to the case of regulation problem. In practice, however, it is desired to make the system to follow a reference trajectory. Therefore, the H_∞ optimal tracking controllers are required. To our knowledge, only in [76] the authors proposed an RL solution to the H_∞ optimal tracking problem. However, their solution is suboptimal and requires complete knowledge of the system dynamics. This is because the dynamic inversion method is used to find the feedforward control input without considering any optimality criterion and it is done in a noncausal manner and require complete knowledge of the system dynamics.

In this chapter, an online off-policy RL algorithm is developed to find the solution to the H∞ optimal tracking problem of nonlinear completely unknown systems. An augmented system is constructed from the tracking error dynamics and the command generator dynamics and a new discounted performance function is introduced for the

H_∞ optimal tracking problem. This enables us to develop a more general version of the L_2-gain where the whole control input and the tracking error energies are weighted by an exponential discount factor in the performance function. This is in contrast to the existing methods that include only the cost of the feedback part of the control input in the performance function. The H_∞ tracing control problem is then transformed to a min-max optimization problem with a discounted performance function. A tracking HJI equation related to the formulated min-max problem is derived which gives both feedforward and feedback parts of the control input simultaneously. Stability and L_2-gain boundness of the solution to the tracking HJI equation is discussed. An off-policy RL algorithm is then developed to find the solution to the tracking HJB equation online using only measured data and without any knowledge about the system dynamics.

The remainder of this chapter is orgainized as follows. The H_∞ tracing control problem is formulated in Section 4.2. A tracking HJI equation is developed in Section 4.3 which gives the solution to the H_∞ tracing control problem. Section 4.4 presents an off-policy RL algorithm for solving the HJI equation online in real time. Sections 4.5 and 4.6 provide the simulation results and conclusion, respectively.

4.2. Problem formulation

In this section, a new formulation for the H_∞ optimal tracking of nonlinear continuous-time system is presented. A general L_2-gain or disturbance attenuation condition is defined. In this new L_2-gain condition, a discounted performance function is used which penalizes both the tracking error and the control effort. A solution to this problem is presented in the next section.

Consider the affine nonlinear continuous-time system defined as

$$\dot{x} = f(x) + g(x)\,u + k(x)\,d \tag{4.1}$$

where $x \in \mathbb{R}^n$ is the state, $u = [u_1, ..., u_m] \in \mathbb{R}^m$ is the control input, $d = [d_1, ..., d_q] \in \mathbb{R}^q$ denotes the external disturbance, $f(x) \in \mathbb{R}^n$ is the drift dynamics, $g(x) \in \mathbb{R}^{n\times m}$ is the input dynamics, and $k(x) \in \mathbb{R}^{n\times q}$ is the disturbance dynamics. It is assumed that the functions $f(x)$, $g(x)$ and $k(x)$ are Lipchitz with $f(0) = 0$, and that the system (4.1) is controllable in the sense that there exists a continuous control on a set $\Omega \subseteq \mathbb{R}^n$ which stabilizes the system in the absence of the disturbance. Moreover, it is assumed that the functions $f(x)$, $g(x)$ and $k(x)$ are unknown.

Let $r(t)$ be the bounded reference trajectory and assume that there exists a Lipschitz continuous command generator function $h_d(.) \in \mathbb{R}^n$ such that

$$\dot{r} = h_d(r) \tag{4.2}$$

and $h_d(0) = 0$. Define the tracking error

$$e_d(t) \triangleq x(t) - r(t) \tag{4.3}$$

Using (4.1)- (4.3), the tracking error dynamics is

$$\dot{e}_d(t) = f(x(t)) - h_d(x_d(t)) + g(x(t))\,u(t) + k(x(t))\,d(t) \tag{4.4}$$

The fictitious performance output to be controlled is defined such that it satisfies

$$\left\| z(t) \right\|^2 = e_d^{\ T} Q\, e_d + u^T R\, u \tag{4.5}$$

Fig. 4.1. shows the system dynamics (4.1) and its inputs and outputs. The goal of the H_∞ tracking is to attenuate the effect of the disturbance input d on the performance output z. Before defining the H_∞ tracking control problem, we define the following general L_2-gain or disturbance attenuation condition.

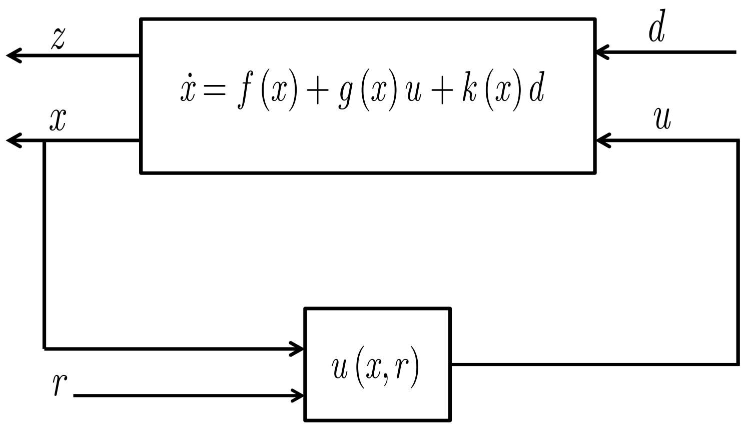

Fig. 4.1. State-feedback H_∞ tracking control problem configuration

Definition 4.1 (Bounded $L_2 -$ gain or disturbance attenuation). The nonlinear system (4.1) is said to have $L_2 -$ gain less than or equal to γ if the following disturbance attenuation condition is satisfied for all $d \in L_2[0,\infty)$.

$$\frac{\int_t^\infty e^{-\alpha(\tau-t)} \left\|z(\tau)\right\|^2 d\tau}{\int_t^\infty e^{-\alpha(\tau-t)} \left\|d(\tau)\right\|^2 d\tau} \le \gamma^2 \tag{4.6}$$

where α is the discount factor , and γ is the attenuation level.

Remark 4.1. The disturbance attenuation condition (4.6) implies that the effect of the disturbance input to the desired performance output is attenuated by a degree at least equal to γ. The minimum value of γ for which the disturbance attenuation condition (4.6) is satisfied gives the so-called optimal robust control solution. However, there exists no way to find the smallest amount of the disturbance attenuation for general nonlinear systems and a large enough value is usually predetermined for γ.

Definition 4.2 (H_∞ optimal tracking). The H_∞ optimal tracking control problem is to find a control policy $u = \beta(e, r)$ for some smooth function β depending on the tracking error e and the desired trajectory r , such that

i) the closed-loop system $\dot{x} = f(x) + g(x)\,\beta(x,r) + k(x)\,d$ satisfies the attenuation condition (4.6).

ii) the tracking error dynamics (4.4) with $d = 0$ is locally asymptotically stable.

Remark 4.2. Previous work on the H_∞ optimal tracking divided the control input into two parts. More specifically, the control input was considered as $u = u_e + u_d$, where u_e is the feedback part which depends only on the tracking error e , and u_d is the feedforward control input which depends only on the reference trajectory. In these methods, u_d was first obtained separately using the dynamic inversion method or the FBI equations without considering any optimality criterion. Then, the problem of optimal design of u_e was reduced to an H_∞ optimal regulation problem. However, ignoring the feedforward control input in the performance may result in a large control effort. Moreover, these methods lead to suboptimal solution as only part of the control input is penalized in the performance function.

Remark 4.3. Note that the performance function (4.6) represents a meaningful cost in the sense that it includes a positive penalty on the tracking error and a positive penalty on the control effort. The use of the discount factor is essential. This is because the feedforward part of the control input does not converge to zero in general and thus penalizing the control input in the performance function without a discount factor makes the performance function unbounded and therefore the meaning of the minimality is lost. Note that in contrast to existing methods, in the proposed method, both feedback and feedback parts of the control input are obtained simultaneously because of the general version of the L_2-gain defined in (4.6) where the whole control input and the tracking error energies are weighted by an exponential discount factor in the performance criterion. In fact, in this way the design of feedforward control input is not separated from the design of the feedback control input.

The control solution to the H_∞ tracking problem with the proposed attenuation condition (4.6) is provided in the subsequent sections. We shall see in the subsequent sections that this general disturbance attenuation condition enables us to find both feedback and feedforward parts of control input simultaneously and therefore extends the method of off-policy RL for solving the problem in hand without requiring any knowledge of the system dynamics.

4.3. Tracking HJI equation and the stability of its solution

In this section, a new formulation for solving the H_∞ tracking control problem is presented. The problem of solving the H_∞ tracking control problem is transformed into a min-max optimization problem subject to an augmented system composed of the tracking error dynamics and the command generator dynamics. A tracking HJI equation is developed which gives the solution to the min-max optimization problem. The stability and L_2-gain bound of the control solution obtained by solving the tracking HJI equation are discussed.

4.3.1. Tracking HJI equation

In the subsection, an augmented system composed of the tracking error system and the command dynamics is constructed. A discounted performance function in terms of the state of the augmented system is defined and it is shown that solving the H_∞ optimal tracking is equivalent to solving a min-max optimization problem with the defined discounted performance function. A tracking HJI equation is then developed to give the solution to the optimization problem in hand.

Define the augmented system state

$$X(t) = [e_d(t)^T \;\; r(t)^T]^T \in \mathbb{R}^{2n} \tag{4.7}$$

where $e_d(t)$ is the tracking error defined in (4.3) and $r(t)$ is the reference trajectory.

Putting (4.2) and (4.4) together yields the augmented system

$$\dot{X}(t) = F(X(t)) + G(X(t))\,u(t) + K(X(t))\,d(t) \tag{4.8}$$

where $u(t) = u(X(t))$ and

$$F(X) = \begin{bmatrix} f(e_d + r) - h_d(r) \\ h_d(r) \end{bmatrix}, \;\; G(X) = \begin{bmatrix} g(e_d + r) \\ 0 \end{bmatrix}, \\ K(X) = \begin{bmatrix} k(e_d + r) \\ 0 \end{bmatrix} \tag{4.9}$$

Using the augmented system (4.8), the disturbance attenuation condition (4.6) becomes

$$\int_t^\infty e^{-\alpha(\tau - t)} \left(X^T Q_T X + u^T R u \right) d\tau \le \gamma^2 \int_t^\infty e^{-\alpha(\tau - t)} (d^T d) d\tau \tag{4.10}$$

where

$$Q_T = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \tag{4.11}$$

Based on (4.10), define the performance function

$$J(u,d) = \int_t^\infty e^{-\alpha(\tau - t)} (X^T Q_T\, X + u^T R\, u - \gamma^2 d^T d)\, d\tau \tag{4.12}$$

Remark 4.4. Note that the problem of finding a control policy that satisfies bounded L_2-gain condition for the optimal tracking problem is equivalent to minimizing the discounted performance function (4.12) subject to the augmented system (4.8).

It is well-known that the H_∞ control problem is closely related to the two-player zero-sum differential game theory [28], [13]. In fact, solvability of the H_∞ control problem is equivalent to solvability of the following zero-sum game [13]

$$V^*(X(t)) = J(u^*, d^*) = \min_u \max_d J(u,d) \tag{4.13}$$

where J is defined in (4.12) and $V^*(X(t))$ is defined as the optimal value function. This two-player zero-sum game control problem has a unique solution if a game theoretic saddle point exist, i.e., if the following Nash condition holds

$$V^*(X(t)) = \min_u \max_d J(u,d) = \max_d \min_u \ J(u,d) \tag{4.14}$$

Note that differentiating (4.12) and noting that $V(X(t)) = J(u(t), d(t))$ gives the following Bellman equation

$$H(V,u,d) \overset{\Delta}{=} X^T Q_T X + u^T R\, u - \gamma^2 d^T d - \alpha V + {V_X}^T \big(F + G\, u + K\, d\big) = 0 \tag{4.15}$$

where $F(X) \triangleq F$, $G \triangleq G(X)$, $K \triangleq K(X)$, and $V_X = \partial V \big/ \partial X$. Applying stationarity conditions $\partial H(V^*,u,d) \big/ \partial u = 0, \partial H(V^*,u,d) \big/ \partial d = 0$ gives the optimal control and disturbance inputs as

$$u^* = -\frac{1}{2} R^{-1} G^T \ {V_X}^* \tag{4.16}$$

$$d^* = \frac{1}{2\gamma^2} K^T \ {V_X}^* \tag{4.17}$$

where V^* is the optimal value function defined in (4.13). Substituting the control input u (4.16) and the disturbance d (4.17) into (4.15), the following tracking HJI equation is obtained

$$H(V^*,u^*,d^*)\overset{\Delta}{=}X^TQ_TX+V_X{}^{*T}F-\alpha V_X-\frac{1}{4}V_X{}^{*T}G^TR^{-1}GV_X{}^*+\frac{1}{4\gamma^2}V_X{}^{*T}KK^TV_X{}^*=0 \qquad (4.18)$$

In the following, it is shown that the control solution (4.16), which is found by solving the HJI equation (4.18), solves the H_∞ tracking problem formulated in Definition 4.2.

4.3.2. Disturbance attenuation and stability of the solution to the HJI equation

In this subsection, it is first shown that the control solution (4.16) satisfies the disturbance attenuation condition (4.10) (part (i) of Definition 4.2). Then, the stability of the tracking error dynamics (4.4) without the disturbance is discussed (part (ii) of Definition 4.2). It is shown that there exists an upper bound α^* such that if the discount factor is less than α^* , the control solution (4.16) make the system locally asymptotically stable.

Theorem 4.1 (Saddle point solution). Consider the H_∞ tracking control problem as a two-player zero-sum game problem with the performance function (4.12). Then, the pair of strategies (u^*,d^*) defined in (4.16)-(4.17) provides a saddle point solution to the game.

Proof. See [2] for the same proof. □

Theorem 4.2 (L_2- gain of system for the solution to the HJI equation). Assume that there exists a continuous positive-semidefinite solution $V^*(X)$ to the tracking HJI equation (4.18). Then u^* in (4.16) makes the closed-loop system (4.18) to have L_2-gain less than or equal to γ .

Proof. The Hamiltonian (4.15) for the optimal value function V^* and any control policy u and disturbance policy w becomes

$$H(V^*,u,d) = X^T Q_T X + u^T R\,u - \gamma^2 d^T d - \alpha V^* + V_X^{*T}\big(F + G\,u + K\,d\big) \tag{4.19}$$

On the other hand, using (4.16)-(4.18) one has

$$H(V^*,u,d) = H(V^*,u^*,d^*) + (u-u^*)^T R\,(u-u^*) + \gamma^2 (d-d^*)^T (d-d^*) \tag{4.20}$$

Based on the HJI equation (4.18), we have $H(V^*,u^*,d^*) = 0$. Therefore, (4.19) and (4.20) give

$$\begin{aligned} &X^T Q_T X + u^T R u - \gamma^2 d^T d - \alpha V^* + V_X^{*T}\big(F + Gu + Kd\big) \\ &= -(u-u^*)^T R\,(u-u^*) - \gamma^2 (d-d^*)^T (d-d^*) \end{aligned} \tag{4.21}$$

Substituting the optimal control policy $u = u^*$ in the above equation yields

$$X^T Q_T X + u^{*T} R u^* - \gamma^2 d^T d - \alpha V^* + V_X^{*T}\big(F + Gu^* + Kd\big) = -\gamma^2 (d-d^*)^T (d-d^*) \le 0 \tag{4.22}$$

Multiplying both sides of this equation by $e^{-\alpha t}$ and defining $\dot{V}^* = V_X^{*T}\big(F + G\,u^* + K\,d\big)$ as the derivative of V^* along the trajectories of the closed-loop system, it gives

$$\frac{d}{dt}\big(e^{-\alpha t} V^*(X)\big) \le e^{-\alpha t}\big(-X^T Q_T\, X - u^{*T} R\, u^* + \gamma^2 d^T d\big) \tag{4.23}$$

Integrating from both sides of this equation yields

$$e^{-\alpha T} V^*(X(T)) - V^*(X(0)) \le \int_0^T e^{-\alpha\tau}\big(-X^T Q_T X - u^{*T} R u^* + \gamma^2 d^T d\big) d\tau \tag{4.24}$$

for every $T > 0$ and every $d \in L_2[0,\infty)$. Since $V^*(.) \ge 0$ the above equation yields

$$\int_0^T e^{-\alpha\tau}\big(X^T Q_T X + u^{*T} R u^*\big) d\tau \le \int_0^T e^{-\alpha\tau}\big(\gamma^2 d^T d\big) d\tau + V^*(X(0)) \tag{4.25}$$

This completes the proof. □

Theorem 4.2 solves part (i) of the state-feedback H_∞ tracking control problem given in Definition 4.2. In the following, we consider the problem of stability of the closed-loop system without disturbance, which is part (ii) of Definition 4.2.

Theorem 4.3 (Stability of the optimal solution for $\alpha \to 0$). Suppose that $V^*(X)$ is a smooth positive-semidefinite and locally quadratic solution to the tracking HJI equation . Then the control input given by (4.16) makes the error dynamics (4.4) with $d = 0$ asymptotically stable in the limit as the discount factor goes to zero.

Proof. Differentiating V^* along the trajectories of the closed-loop system with $d = 0$ and using the tracking HJI equation gives

$$V_X^{*T}(F + G\,u^*) = \alpha V^* - X^T Q_T\, X - u^{*T} R\, u^* + \gamma^2 d^T d \tag{4.26}$$

Or equivalently,

$$\frac{d}{dt}(e^{-\alpha t} V^*(X)) = e^{-\alpha t}(-X^T Q_T X - u^{*T} R u^* + \gamma^2 d^T d) \le 0 \tag{4.27}$$

If the discount factor goes to zero, then LaSalle's extension can be used to show that the tracking error is locally asymptotically stable. More specifically, if $\alpha \to 0$, based on LaSalle's extension, $X(t) = [e_d(t)^T \;\; r(t)^T]^T$ goes to a region wherein $\dot{V} = 0$. Since $X^T Q_T X = e_d(t)^T Q\, e_d(t)$ where Q is positive definite, $\dot{V} = 0$ only if $e_d(t) = 0$ and $u = 0$ when $d = 0$. On the other hand, $u = 0$ also requires that $e_d(t) = 0$, therefore, for $\gamma = 0$ the tracking error is locally asymptotically stable. □

Theorem 4.3 shows that if the discount factor goes to zero, then optimal control solution found by solving the tracking HJI equation makes the system locally asymptotically stable. However, if the discount factor is nonzero, local asymptotic stability of the optimal control solution cannot be guaranteed by Theorem 4.3. In the following Theorem 4.4, it is shown that local asymptotic stability of the optimal solution is guaranteed as long as the discount factor is smaller than an upper bound. Before presenting the proof of local asymptotic stability, the following example shows that if the discount factor is not small, the control solution obtained by solving the tracking HJI equation can make the system unstable.

Example 4.1. Consider the scalar dynamical system

$$\dot{X} = X + u + d \tag{4.28}$$

Assume that in the HJI equation (4.18) we have $Q_T = R = 1$ and the attenuation level is $\gamma = 1$. For this linear system with quadratic performance, the value function is quadratic. That is, $V(X) = p\,X^2$ and therefore the HJI equation reduces to

$$(2-\alpha)\,p - \frac{3}{4}p^2 + 1 = 0 \tag{4.29}$$

and the optimal control solution becomes

$$u = -p\,X \tag{4.30}$$

Solving this equation gives the optimal solution as

$$u = -\Big(\frac{4}{3}(1-0.5\,\alpha) + \frac{2}{\sqrt{3}}\sqrt{\frac{4}{3}(1-0.5\,\alpha)^2 + 1}\,\Big)X \tag{4.31}$$

However, this optimal solution does not make the system stable for all values of the discount factor α. If fact, if $\alpha > \alpha^* = 27/12$, then the system is unstable. The next theorem shows how to find an upper bound α^* for the discount factor to assure the stability of the system without disturbance.

Before presenting the stability theorem, note that the augmented system dynamics (4.8) can be written as

$$\dot{X} = F(X) + G(X)u + K(X)d = AX + Bu + Dd + \bar{F}(X) \tag{4.32}$$

where $AX + Bu + Kd$ is the linearized model with

$$A = \begin{bmatrix} A_{l1} & A_{l1} - A_{l2} \\ 0 & A_{l2} \end{bmatrix} B = [B_l^T \quad 0^T]^T, D = [D_l^T \quad 0^T]^T \tag{4.33}$$

where A_{l1} and A_{l2} are the linearized models of the drift system dynamics f and the command generator dynamics h_d, respectively, and $\bar{F}(X)$ is the remaining nonlinear terms.

Theorem 4.4 (Stability of the optimal solution and upper bound for α**).** Consider the system (4.8). Define

$$L_l = B_l R^{-1} B_l^T + \frac{1}{\gamma^2} D_l D_l^T \tag{4.34}$$

where B_l and D_l are defined in (4.33). Then, the control solution (4.16) makes the error system (4.4) with $d = 0$ locally asymptotically stable if

$$\alpha \le \alpha^* = 2\left\| (L_l Q)^{1/2} \right\| \tag{4.35}$$

Proof. Given the augmented dynamics (4.8) and the performance function (4.12), the Hamiltonian function in terms of the optimal control and disturbance is defined as

$$H(\rho, u^*, d^*) = e^{-\alpha t}(X^T Q_T X + u^{*T} R u^* - \gamma^2 d^{*T} d^*) + \rho^T\left(F + G u^* + K d^*\right) \tag{4.36}$$

where ρ is known as the costate variable. Using Pontryagin's maximum principle, the optimal solutions u^* and d^* satisfy the following state and costate equations.

$$\dot{X} = H_\rho(X, \rho) \tag{4.37}$$

$$\dot{\rho} = -H_X(X, \rho) \tag{4.38}$$

Define the new variable

$$\mu = e^{\alpha t} \rho \tag{4.39}$$

Based on (4.39), define the modified Hamiltonian function as

$$H^m = e^{-\alpha t} H = (X^T Q_T X + u^{*T} R u^* - \gamma^2 d^{*T} d^*) + \mu^T\left(F + G u^* + K d^*\right) \tag{4.40}$$

Then, conditions (4.37) and (4.38) become

$$\dot{X} = {H^m}_{\mu}(X, \mu) \tag{4.41}$$

$$\dot{\mu} = \alpha\,\mu - {H^m}_X(X, \mu) \tag{4.42}$$

Equation (4.41) gives the augmented system dynamics (4.8) and equation (4.42) is equivalent to the HJI equation (4.18) with $\mu = {V^*}_X$. In order to prove the local stability of the closed-loop system, the stability of the closed-loop linearized system is investigated. Using (4.32) for the system dynamics, equation (4.40) becomes

$$H^m = (X^T Q_T X + u^{*T} R\, u^* - \gamma^2 d^{*T} d^*) + \mu^T\big(AX + Bu^* + Dd^* + \bar{F}(X)\big) \tag{4.43}$$

Then, the costate can be written as sum of a linear and a nonlinear term as

$$\mu = 2PX + \varphi_0(X) \equiv \mu_1 + \varphi_0(X) \tag{4.44}$$

Using $\partial H^m \big/ \partial u = 0,\ \partial H^m \big/ \partial d = 0$ and (4.44) one has

$$u^* = -R^{-1}B^T PX + \varphi_1(X) \tag{4.45}$$

$$d^* = \frac{1}{\gamma^2} D^T PX + \varphi_2(X) \tag{4.46}$$

for some $\varphi_1(X)$ and $\varphi_2(X)$ depending on $\varphi_0(X)$, $\bar{F}(X)$ and P. Using (4.36)- (4.46), conditions (4.41) and (4.42) becomes

$$\begin{bmatrix} \dot{X} \\ \dot{\mu}_1 \end{bmatrix} = \begin{bmatrix} A & -(B\,R^{-1}B^T - \frac{1}{\gamma^2} D\,D^T) \\ -Q_T & -A^T + \alpha I \end{bmatrix} \begin{bmatrix} X \\ \mu_1 \end{bmatrix} + \begin{bmatrix} F_1(X) \\ F_2(X) \end{bmatrix} \triangleq W \begin{bmatrix} X \\ \mu_1 \end{bmatrix} + \begin{bmatrix} F_1(X) \\ F_2(X) \end{bmatrix} \tag{4.47}$$

for some nonlinear functions $F_1(X)$ and $F_1(X)$. The linear part of costate is a stable manifold of W and thus based on the linear part of (4.47), it satisfies the following GARE

$$Q_T + A^T P + PA - \alpha P - PBR^{-1}B^T P + \frac{1}{\gamma^2} PDD^T P = 0 \tag{4.48}$$

Define

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix}$$

Then, based on (4.11) and (4.33), the upper left-hand side of the LQT GARE (4.48) becomes

$$Q + A_{l1}^{\ T} P_{11} + P_{11} A_{l1} - \alpha P_{11} - P_{11} B_l R^{-1} B_l^T P_{11} + \frac{1}{\gamma^2} P_{11} D_l D_l^T P_{11} = 0 \qquad (4.49)$$

The closed-loop system dynamics for the control input (4.45) and without the disturbance is

$$\dot{X} = (A - BR^{-1} B^T P) X + F_f(X) \qquad (4.50)$$

for some nonlinear function $F_f(X)$ with $F_f = [F_{f1}^{\ T}, F_{f2}^{\ T}]^T$, which gives the following tracking error dynamics

$$\dot{e}_d = (A_{l1} - B_l R^{-1} B_l^{\ T} P_{11}) e_d + F_{f1} = A_c e_d + F_{f1} \qquad (4.51)$$

Based on the closed-loop error dynamics A_c, the GARE becomes

$$Q + A_c^{\ T} P_{11} + P_{11} A_c - \alpha P_{11} + P_{11} B_l R^{-1} B_l^T P_{11} + \frac{1}{\gamma^2} P_{11} D_l D_l^T P_{11} = 0 \qquad (4.52)$$

To find a condition on the discount factor to assure stability of the linearized error dynamics, assume that λ is an eigenvalue of the closed-loop error dynamics A_c. That is $A_c x = \lambda x$ with x the eigenvector corresponding to λ. Then, multiplying the left- and right- hand sides of the GARE (4.52) by x^T and x, respectively, one has

$$2\left(\mathrm{Re}(\lambda) - 0.5\,\alpha\right) x^T P_{11} x = -x^T Q\, x - x^T P_{11} (B_l R^{-1} B_l^T + DD^T) P_{11} x \qquad (4.53)$$

Using the inequality $a^2 + b^2 \geq 2ab$ and since $P_{11} > 0$, (4.53) becomes

$$\left(\mathrm{Re}(\lambda) - 0.5\,\alpha\right) \leq -\left\|(Q P_{11}^{\ -1})^{1/2}\right\| \left\|(L_l P_{11})^{1/2}\right\| \qquad (4.54)$$

or equivalently,

$$\mathrm{Re}(\lambda) \leq -\left\|(Q P_{11}^{\ -1})^{1/2}\right\| \left\|(L_l P_{11})^{1/2}\right\| + 0.5\,\alpha \qquad (4.55)$$

where L_l is defined in (4.34). Using the fact that $\|A\|\|B\| \geq \|AB\|$ gives

$$\mathrm{Re}(\lambda) \leq -\left\|(L_l Q)^{1/2}\right\| + 0.5\,\alpha \tag{4.56}$$

Therefore, the linear error dynamics in (4.51) is stable if condition (4.35) is satisfied and this completes the proof. □

Remark 4.5. Note that the GARE (4.49) can be written as

$$Q_T + (A - 0.5\alpha I)^T P + P(A - 0.5\alpha I) - PBR^{-1}B^T P + \frac{1}{\gamma^2} PDD^T P = 0$$

This amounts to a GARE without discount factor and with the system dynamics given by $A - 0.5\alpha I$, B and D. Therefore, existence of a unique solution to the GARE requires $(A - 0.5\alpha I, B)$ be stabilizable. Based on definition of A and B in (4.33), this requires that $(A_{l1} - 0.5\alpha I, B_l)$ be stabilizable and $(A_{l2} - 0.5\alpha I)$ be stable. However, since (A_{l1}, B_l) is stabilizable, as the system dynamics in (4.1) is assumed robustly stabilizing, then $(A_{l1} - 0.5\alpha I, B_l)$ is also stabilizable for any $\alpha > 0$. Moreover, since the reference trajectory is assumed bounded, the linearized model of the command generator dynamics, i.e. A_{l2}, is marginally stable and thus $(A_{l2} - 0.5\alpha I)$ is stable. Therefore, the discount factor does not affect the existence of the solution to the GARE.

Remark 4.6. Theorem 4.4 shows that the asymptotic stability of only the first n variables of X is guaranteed, which are the error dynamic states. This is reasonable as the last n variables of X are the reference command generator variables which are not under our control.

Remark 4.7. For Example 4.1, condition (4.34) gives the bound $\alpha < \sqrt{80}/12$ to assure the stability. This bound is very close to the actual bound obtained in Example 4.1. However, it is obvious that condition (4.34) gives a conservative bound for the discount factor to assure the stability.

Remark 4.8. Theorem 4.4 confirms the existence of an upper bound for the discount factor to assure stability of the solution to the HJI tracking equation and relates this bound to the input and disturbance dynamics, and the weighting matrices in the performance function. Condition (4.35) is not a restrictive condition even if the system dynamics are unknown. In fact, one can always pick a very small discount factor, and/or large weighting matrix Q (which is a design matrix) to assure that condition (4.35) is satisfied.

4.4. Off-policy RL for solving the tracking HJI equation

In this section, an offline RL algorithm is first given to solve the problem of H_∞ optimal tracking by learning the solution to the tracking HJI equation. An off-policy IRL algorithm is then developed to learn the solution to the HJI equation online and without requiring any knowledge of the system dynamics. Three neural networks on an actor-critic-disturbance structure are used to implement the proposed off-policy IRL algorithm.

4.4.1. Off-policy RL algorithm

The Bellman equation (4.15) is linear in the cost function V, while the HJI equation (4.18) is nonlinear in the value function V^*. Therefore, solving the Bellman equation for V is easier than solving the HJI for V^*. Instead of directly solving for V^*, policy iteration (PI) algorithm iterates on both control and disturbance players to break the HJI equation into a sequence of differential equations linear in the cost. An offline PI algorithm for solving the H_∞ optimal tracking problem is given as follows:

Algorithm 4.1. Offline RL algorithm

Initialization: Start with an admissible stabilizing control policy u_0

1. For a control input u_i and disturbance policy d_i, find V_i using the following Bellman equation

$$H(V_i,\, u_i, d_i) = X^T\, Q_T\, X + {V_{Xi}}^T (F + G\,u_i + K\,d_i) - \alpha V_i + {u_i}^T R\, u_i - \gamma^2 {d_i}^T d_i = 0 \tag{4.57}$$

2. Update the disturbance using

$$d_{i+1} = \arg\max_d \left[H(V_i, u_i, d)\right] = \frac{1}{2\gamma^2} K^T\, V_{Xi} \tag{4.58}$$

and the control policy using

$$u_{i+1} = \arg\min_u \left[H(V_i, u, d)\right] = -\frac{1}{2} R^{-1} G^T V_{Xi} \tag{4.59}$$

3. Go to 1.

Algorithm 4.1 extends the results of the simultaneous RL algorithm in [130] to the tracking problem. The convergence of this algorithm to the minimal nonnegative solution of the HJI equation was shown in [130]. In fact, similar to [130], the convergence of Algorithm 4.1 can be established by proving that iteration on (4.58) is essentially a Newtons iterative sequence which converges to the unique solution of the HJI equation (4.18).

Algorithm 4.1 requires complete knowledge of the system dynamics. In the following, an off-policy IRL algorithm is developed solve the H_∞ optimal tracking for systems with completely unknown dynamics. To this end, the system dynamics (4.8) is first written as

$$\dot{X} = F + G\,u_i + K\,d_i + G(u - u_i) + K(d - d_i) \tag{4.60}$$

where $u_i = [u_{i,1}, ..., u_{i,m}] \in \mathbb{R}^m$ and $d_i = [d_{i,1}, ..., d_{i,q}] \in \mathbb{R}^q$ are policies to be updated. Differentiating $V_i(X)$ along with the system dynamics (4.60) and using (4.57)- (4.59) gives

$$\begin{aligned}\dot{V}_i &= V_{Xi}{}^T(F + G\,u_i + K\,d_i) + V_{Xi}{}^T G(u - u_i) + V_{Xi}{}^T K(d - d_i) = \alpha V_i - X^T Q_T\, X \\ &\quad - u_i{}^T R\,u_i + \gamma^2 d_i{}^T d_i - 2\,u_{i+1}{}^T R(u - u_i) + 2\gamma^2 d_{i+1}{}^T (d - d_i)\end{aligned} \tag{4.61}$$

Multiplying both sides of (4.61) by $e^{-\alpha(\tau - t)}$ and integrating from both sides yields the following off-policy IRL Bellman equation

$$\begin{aligned}&e^{-\alpha T} V_i(X(t+T)) - V_i(X(t)) = \int_t^{t+T} e^{-\alpha(\tau - t)}(-X^T Q_T\, X - u_i{}^T R\,u_i + \gamma^2 d_i{}^T d_i)\, d\tau + \\ &\int_t^{t+T} e^{-\alpha(\tau - t)}(-2\,u_{i+1}{}^T R(u - u_i) + 2\gamma^2 d_{i+1}{}^T (d - d_i))\, d\tau\end{aligned} \tag{4.62}$$

Note that for a fixed control policy u (the policy which is applied to the system), and a given disturbance d (the actual disturbance which is applied to the system), equation (4.62) can be solved for both value function V_i and updated policies u_{i+1} and d_{i+1}, simultaneously.

Lemma 4.1. The off-policy IRL equation (4.62) gives the same solution for the value function as the Bellman equation (4.57) and the same updated control and disturbance policies as (4.58) and (4.59).

Proof. Dividing both sides of the off-policy IRL Bellman equation (4.62) by T and taking limit results in

$$\lim_{T\to 0}\frac{e^{-\alpha T}V_i(X(t+T))-V_i(X(t))}{T}+\lim_{T\to 0}\frac{\int_t^{t+T} e^{-\alpha(\tau-t)}(X^TQ_TX+u_i^TRu_i-\gamma^2 d_i^Td_i)d\tau}{T}+ \\ \lim_{T\to 0}\frac{\int_t^{t+T} e^{-\alpha(\tau-t)}(2u_{i+1}^TR(u-u_i)-2\gamma^2 d_{i+1}^T(d-d_i))d\tau}{T}=0 \tag{4.63}$$

By L'Hopital's rule, the first term in (4.42) becomes

$$\lim_{T\to 0}\frac{e^{-\alpha T}V_i(X(t+T))-V_i(X(t))}{T}=\lim_{T\to 0}[-\alpha e^{-\alpha T}V_i(X(t+T))+e^{-\alpha T}\dot V_i(X(t+T))]= \\ -\alpha V_i+V_{Xi}\big(F+Gu_i+Kd_i+G(u-u_i)+K(d-d_i)\big) \tag{4.64}$$

where the last term in the right-hand side is obtained by using $\dot V=V_X\,\dot X$. Similarly, for the second and third terms of (4.42) one has

$$\lim_{T\to 0}\frac{\int_t^{t+T} e^{-\alpha(\tau-t)}(X^TQ_TX+u_i^TRu_i-\gamma^2 d_i^Td_i)d\tau}{T}=X^TQ_TX+u_i^TRu_i-\gamma^2 d_i^Td_i$$
(4.65)

$$\lim_{T\to 0}\frac{\int_t^{t+T} e^{-\alpha(\tau-t)}(2u_{i+1}^TR(u-u_i)-2\gamma^2 d_{i+1}^T(d-d_i))d\tau}{T}= \\ 2u_{i+1}^TR(u-u_i)-2\gamma^2 d_{i+1}^T(d-d_i) \tag{4.66}$$

Substituting (4.64)- (4.66) in (4.42) yields

$$-\alpha V_i+V_{Xi}\big(F+G\,u_i+K\,d_i+G\,(u-u_i)+K\,(d-d_i)\big)+X^TQ_TX+u_i^TRu_i \\ -\gamma^2 d_i^Td_i+2\,u_{i+1}^TR\,(u-u_i)-2\gamma^2 d_{i+1}^T(d-d_i)=0 \tag{4.67}$$

Substituting the updated policies u_{i+1} and d_{i+1} from (4.58) and (4.59) into (4.67), gives the Bellman equation (4.57). This completes the proof. □

Remark 4.9. In the off-policy IRL Bellman equation (4.62), the control input u which is applied to the system can be different from the control policy u_i which is evaluated and updated. The fixed control policy u should be a stable and exploring control policy. Moreover, in this off-policy IRL Bellman equation, the disturbance input d is the actual external disturbance that comes from a disturbance source and is not under our control. However, the disturbance d_i is the disturbance which is evaluated and updated. One advantage of this off-policy IRL Bellman equation is that, in contrast to on-policy RL-based methods, the disturbance input which is applied to the system does not require to be adjustable.

The following algorithm uses the off-policy tracking Bellman equation (4.62) to iteratively solve the HJI equation (4.18) without requiring any knowledge of the system dynamics. The implementation of this algorithm is discussed in the next subsection. It is shown how the data collected from a fixed control policy u is reused to evaluate many updated control policies u_i sequentially until convergence to the optimal solution is achieved.

Algorithm 4.2. Online Off-policy RL algorithm for solving tracking HJI equation

Phase 1 (data collection using a fixed control policy): Apply a fixed control policy u to the system and collect required system information about the state, control input and disturbance at *N* different sampling interval T .

Phase 2 (reuse of collected data sequentially to find an optimal policy iteratively): Given u_i and d_i, use collected information in phase 1 to Solve the following Bellman equation for V_i , u_{i+1} and d_{i+1} simultaneously

$$e^{-\alpha T}V_i(X(t+T)) - V_i(X(t)) = \int_t^{t+T} e^{-\alpha(\tau-t)}(-X^T Q_T\ X - u_i^T R u_i + \gamma^2 d_i^T d_i)\, d\tau$$
$$+\int_t^{t+T} e^{-\alpha(\tau-t)}(-2u_{i+1}^T R(u-u_i) + 2\gamma^2 d_{i+1}^T (d-d_i))d\tau \tag{4.68}$$

Stop if a stopping criterion is met, otherwise set $i = i+1$ and got to 2.

Remark 4.10. Algorithm 4.2 has two separate phases. First, a fixed initial exploratory control policy u is applied and the system information is recorded over the time interval T. Second, without requiring any knowledge of the system dynamics, the information collected in phase 1 are repeatedly used to find a sequence of updated policies u_i and d_i converging to u^* and d^*. Note that equation (4.68) is a scalar equation and can be solved in a least square sense after collecting enough number of data samples from the system. It is shown in the following section how to collect required information in phase 1 and reuse them in phase 2 in a least-square sense to solve (4.68) for V_i , u_{i+1} and d_{i+1} simultaneously. After the learning is done and the optimal control policy u^* is found, it can then be applied to the system.

Theorem 4.5 (Convergence of Algorithm 4.2). The off-policy Algorithm 4.2 converges to the optimal control and disturbance solutions given by (4.16) and (4.17) where the value function satisfies the tracking HJI equation (4.18).

Proof. It was shown in Lemma 1 that the off-policy tracking Bellman equation (4.68) gives the same value function as the Bellman equation (4.57) and the same updated policies as (4.58) and (4.59). Therefore both Algorithms 4.1 and 4.2 have the same convergence properties. Convergence of Algorithm 4.1 is proven in [130]. This confirms that Algorithm 4.2 converges to the optimal solution. □

Remark 4.11. Although both Algorithms 4.1 and 4.2 have the same convergence properties, Algorithm 4.2 is a model-free algorithm which finds an optimal control policy without requiring any knowledge of the system dynamics. This is in contrast to Algorithm 4.1 which requires full knowledge of the system dynamics. Moreover, Algorithm 4.1 is an on-policy RL algorithm which requires the disturbance input be specified and adjustable. On the other hand, Algorithm 4.2 is an off-policy RL algorithm which obviates this requirement.

4.4.2. Implementing the proposed off-policy RL algorithm

In order to implement the off-policy RL Algorithm 4.2, it is required to reuse the collected information found by applying a fixed control policy u to the system to solve equation (4.68) for V_i, u_{i+1} and d_{i+1} iteratively. Three neural networks (NNs), i.e. the actor NN, the critic NN, and the disturber NN are used here to approximate the value function and the updated control and disturbance policies in the Bellman equation (4.68). That is, the solution V_i, u_{i+1} and d_{i+1} of the Bellman equation (4.68) is approximated by three NNs as

$$\hat{V}_i(X) = \hat{W}_1^T \sigma(X) \tag{4.69}$$

$$\hat{u}_{i+1}(X) = \hat{W}_2^T \phi(X) \tag{4.70}$$

$$\hat{d}_{i+1}(X) = \hat{W}_3^T \varphi(X) \tag{4.71}$$

where $\sigma = [\sigma_1, ..., \sigma_{l_1}] \in \mathbb{R}^{l_1}$, $\phi = [\phi_1, ..., \phi_{l_2}] \in \mathbb{R}^{l_2}$ and $\varphi = [\varphi_1, ..., \varphi_{l_3}] \in \mathbb{R}^{l_3}$ provide suitable basis function vectors, $\hat{W}_1 \in \mathbb{R}^{l_1}$, $\hat{W}_2 \in \mathbb{R}^{m \times l_2}$, and $\hat{W}_3 \in \mathbb{R}^{q \times l_3}$ are constant weight vectors, and l_1, l_2 and l_2 are the number of neurons. Define $v^1 = [v_1^1, ..., v_1^m]^T = u - u_i$, $v^2 = [v_1^2, ..., v_q^2]^T = d - d_i$ and assume $R = diag(r, ..., r_m)$. Then, substituting (4.69)-(4.71) in (4.68) yields

$$e(t) = \hat{W}_1^T\big(e^{-\alpha T}\sigma(X(t+T)) - \sigma(X(t))\big) - \int_t^{t+T} e^{-\alpha(\tau-t)}\big(-X^T Q_T X - u_i^{\ T} R u_i + \gamma^2 d_i^{\ T} d_i\big)d\tau$$
$$+2\sum_{l=1}^{m} r_l \int_t^{t+T} e^{-\alpha(\tau-t)}\ \hat{W}_{2,l}^{\ \ T}\phi(X(t))\, v_l^1\, d\tau - 2\gamma^2 \sum_{k=1}^{q} \int_t^{t+T} e^{-\alpha(\tau-t)}\ \hat{W}_{3,k}^{\ \ T}\varphi(X(t))\, v_k^2\, d\tau \tag{4.72}$$

where $e(t)$ is the Bellman approximation error, $\hat{W}_{2,l}$ is the l-th column of $\hat{W}_2$, and $\hat{W}_{3,k}$ is the k-th column of $\hat{W}_3$. The Bellman approximation error is the continuous-time counterpart of the temporal difference (TD). In order to bring the TD error to its minimum value, least squares method is used. To this end, rewrite equation (4.72) as

$$y(t) + e(t) = \hat{W}^T h(t) \tag{4.73}$$

where

$$\hat{W} = [\hat{W}_1^T, \hat{W}_{2,l}^{\ \ T}, ..., \hat{W}_{2,m}^{\ \ T}, \hat{W}_{3,1}^{\ \ T}, ..., \hat{W}_{3,q}^{\ \ T}]^T \in \mathbb{R}^{l_1 + m\times l_2 + q\times l_3} \tag{4.74}$$

$$h(t) = \begin{bmatrix} e^{-\alpha T}\sigma(X(t+T)) - \sigma(X(t))\big) \\ 2r_1 \int_t^{t+T} e^{-\alpha(\tau-t)}\ \phi(X(t))\, v_1^1\, d\tau \\ \vdots \\ 2r_m \int_t^{t+T} e^{-\alpha(\tau-t)}\ \phi(X(t))\, v_m^1\, d\tau \\ -2\gamma^2 \int_t^{t+T} e^{-\alpha(\tau-t)}\ \varphi(X(t))\, v_1^2\, d\tau \\ \vdots \\ -2\gamma^2 \int_t^{t+T} e^{-\alpha(\tau-t)}\ \varphi(X(t))\, v_q^2\, d\tau \end{bmatrix} \tag{4.75}$$

$$y(t) = \int_t^{t+T} e^{-\alpha(\tau-t)}(-X^T Q_T X - u_i^{\ T} R u_i + \gamma^2 d_i^{\ T} d_i)d\tau \tag{4.76}$$

The parameter vector $\hat{W}$, which gives the approximated value function, actor and disturbance (4.76)-(4.71), is found by minimizing, in the least-squares sense, the Bellman error (4.74). Assume that the systems state, input and disturbance information are collected at $N \geq l_1 + m \times l_2 + q \times l_3$ (the number of independent elements in $\hat{W}$) points t_1

to t_N in the state space, over the same time interval T in phase 1. Then, for a given u_i and d_i, one can use this information to evaluate (4.75) and (4.76) at N points to form

$$H = [h(t_1), ..., h(t_N)] \tag{4.77}$$

$$Y = [y(t_1), ..., y(t_N)]^T \tag{4.78}$$

The least-squares solution to (4.73) is then equal to

$$\hat{W} = (HH^T)^{-1} HY \tag{4.79}$$

which gives V_i, u_{i+1} and d_{i+1}.

Remark 4.12. Note that although $X(t+T)$ appears in equation (4.72), this equation is solved in a least square sense after observing N samples $X(t)$, $X(t+T)$, …, $X(t+NT)$. Therefore, the knowledge of the system is not required to predict the future state $X(t+T)$ at time t to solve (4.72).

4.5. Simulation Results

In this section, the proposed off-policy IRL method is first applied to a linear system to show that it converges to the optimal solution. Then, it is tested on a nonlinear system.

4.5.1. Linear system

Consider the F-16 aircraft system described by $\dot{x} = Ax + Bu + Dd$ with the following dynamics

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.80}$$

The system state vector is $x = [x_1 \ \ x_2 \ \ x_3] = [\alpha \ \ q \ \ \delta_e]$, where α denotes the angle of attack, q is the pitch rate, and δ_e is the elevator deflection angle. The control input is the elevator actuator voltage, and the disturbance is wind gusts on angle of attack. It is assumed that the output is $y=\alpha$ and the desired value is constant. Thus the command generator dynamics become $\dot{r}=0$. Therefore, the augmented dynamics (4.8) becomes equal to equation (4.81). Since only $e_1 = x_1 - r_1$ is concerned as the tracking error, the first element of the matrix Q_T in (4.11) is consider to be 20 and all other elements are zero. It is also assumed here that $R=1$, and $\gamma=10$. The offline solution to the game ARE (4.48) and consequently the optimal control policy are given as

$$\dot{X}=\begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 & 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} d \tag{4.81}$$

$$P^* = \begin{bmatrix} 12.677 & 5.420 & -0.432 & -7.474 & 5.420 & -0.432 \\ 5.420 & 3.405 & -0.332 & -4.980 & 3.405 & -0.332 \\ -0.432 & -0.332 & 0.040 & 0.544 & -0.332 & 0.040 \\ -7.474 & -4.980 & 0.544 & 201.451 & -4.980 & 0.544 \\ 5.420 & 3.405 & -0.332 & -4.980 & 3.405 & -0.332 \\ -0.432 & -0.332 & -0.205 & 0.040 & -0.332 & 0.040 \end{bmatrix}, \tag{4.82}$$

$$u^* = -[-2.1620,-1.6623,0.2005,2.7198,-1.6623,0.2005]X$$

We now implement the off-policy IRL Algorithm 4.2. The reinforcement interval is chosen as $T=0.05$. The initial control gain is chosen as zero. Figs. 4.2 and 4.3 show convergence of the kernel matrix P and the control gain to their optimal values. In fact, P converges to

$$P = \begin{bmatrix} 12.675 & 5.418 & -0.432 & -7.481 & 5.424 & -0.439 \\ 5.420 & 3.412 & -0.330 & -4.985 & 3.404 & -0.329 \\ -0.427 & -0.323 & 0.042 & 0.546 & -0.333 & 0.046 \\ -7.495 & -4.973 & 0.545 & 201.408 & -4.985 & 0.527 \\ 5.419 & 3.406 & -0.328 & -4.968 & 3.405 & -0.339 \\ -0.421 & -0.347 & -0.201 & 0.036 & -0.333 & 0.046 \end{bmatrix}$$

which is very close to its optimal value. These results and Figs. 4.2 and 4.3 confirm that the proposed method converses to the optimal tracking solution without requiring the knowledge of the system dynamics. The optimal control solution found by the proposed method is now applied to the system to test its performance. To this end, it is assumed that the desired value for the output is $r_1 = 2$ for time zero to 30sec and changes to $r_1 = 3$ at time 30sec. The disturbance is assumed to be $d = 0.1e^{-0.1t}\sin(0.1t)$. Fig. 4.4. shows how the output converges to its desired values after the optimal control solution is applied to the system and confirms that the proposed optimal control solution achieves suitable results.

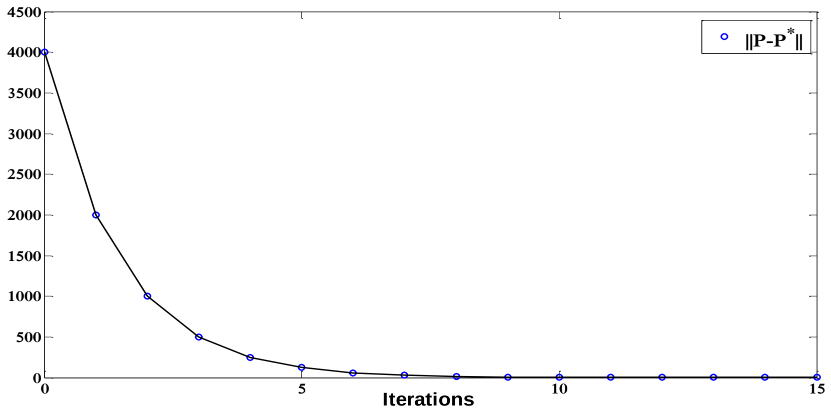

Fig. 4.2. Convergence of the kernel matrix P to its optimal value for F-16 example

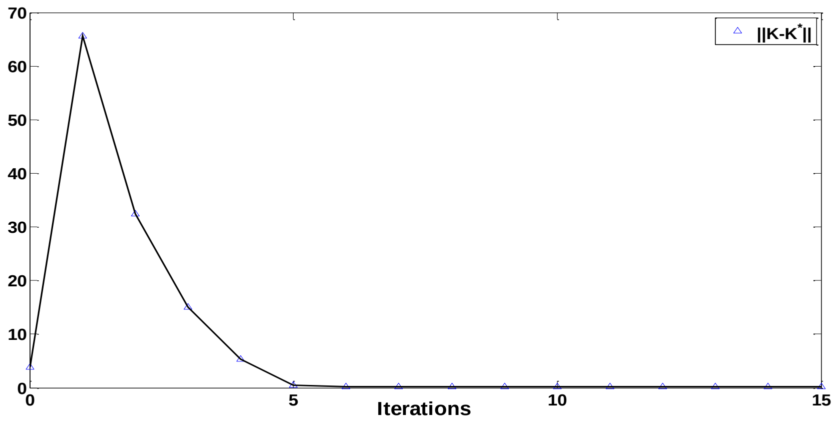

Fig. 4.3. Convergence of the control gain to its optimal value for F-16 example

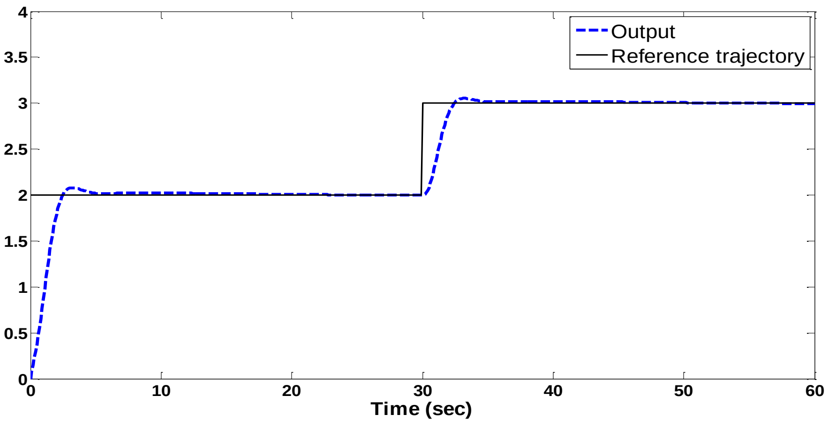

Fig. 4.4. Reference trajectory versus output for F-16 systems using the proposed control method

4.5.2. Nonlinear system

In this subsection, the proposed off-policy IRL algorithm is applied to a two-link manipulator [64], which is modeled using

$$M\ddot{q} + V_m\dot{q} + F_d\dot{q} + G(q) = u + d \tag{4.83}$$

where $q = [q_1 \;\; q_2]^T$ is the vector of joint angles and $\dot{q} = [\dot{q}_1 \;\; \dot{q}_2]^T$ is the vector of joint angular velocities, and

$$M = \begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix}, \; V_m = \begin{bmatrix} -p_3s_2\dot{q}_2 & -p_3s_2(\dot{q}_1 + \dot{q}_2) \\ p_3s_2\dot{q}_1 & 0 \end{bmatrix}$$

are the inertia and Coriolis-centripetal matrices, respectively, with $c_2 = \cos(q_2)$, $s_2 = \sin(q_2)$, $p_1 = 3.473\ kg\,m^2$, $p_2 = 0.196\ kg\,m^2$ and $p_3 = 0.242\ kg\,m^2$. Moreover, $F_d = diag\,[5.3,\,1.1]$, $G(q) = [8.45\tanh(\dot{q}_1),\,2.35\tanh(\dot{q}_2)]^T$, u and τ_d are the static friction, the dynamic friction, the control torque, and the disturbance, respectively.

Defining the state vector as $x = [q_1\ q_2\ \dot{q}_1\ \dot{q}_2]^T$, the state-space equations for (4.83) becomes (4.1) with

$$f(x) = \begin{bmatrix} x_3 & x_4 & \left(M^{-1}(-V_m - F_d)\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} - G(q)\right)^T \end{bmatrix}^T$$

$$g(x) = k(x) = \begin{bmatrix} [0\ 0]^T & [0\ 0]^T & [0\ 0]^T & (M^{-1})^T \end{bmatrix}^T$$

The objective is to find the control input u to make the state follow the desired trajectory given as

$$r = \begin{bmatrix} 0.5\cos(2t) & 0.33\cos(3t) & -\sin(2t) & -\sin(3t) \end{bmatrix}^T$$

which is generated by the command generator (4.2) with

$$h_d(r) = \begin{bmatrix} r_3 & r_4 & -4r_1 & -9r_2 \end{bmatrix}^T$$

It is assumed in the disturbance attenuation condition (4.7) that $Q = 10\,I \quad R = 1$, and $\gamma = 20$. The augmented state becomes $X = [e_1\ \ e_2\ \ e_3\ \ e_4\ \ r_1\ \ r_2\ \ r_3\ \ r_4]^T$ with $e_i = x_i - r_i,\ i = 1,2,3,4$. A power series neural network containing even powers of the state variables of the system up to order four is used for the critic. The activation functions for the control and disturbance policies are chosen as polynomials of all powers of the states up to order three. We now implement Algorithm 4.2 to find the optimal control solution online. The reinforcement interval is chosen as $T = 0.05$. The proposed algorithm starts the learning process from the beginning of the simulation and finishes it after 25 second, when the control policy is updated. The plots of state trajectories of the closed-loop system and the reference trajectory are shown in Figs. 4.5-4.8. The disturbance is assumed to be $d = 0.1e^{-0.1t}\sin(0.1t)$ after the learning is done. From these figs, it is obvious that the system tracks the reference trajectory after the learning is finished and the optimal controller is found.

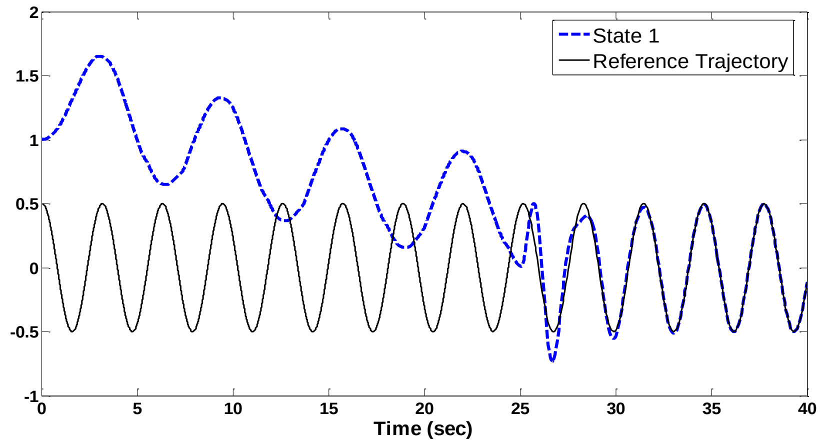

Fig. 4.5. Reference trajectory versus the first state of the robot manipulator systems

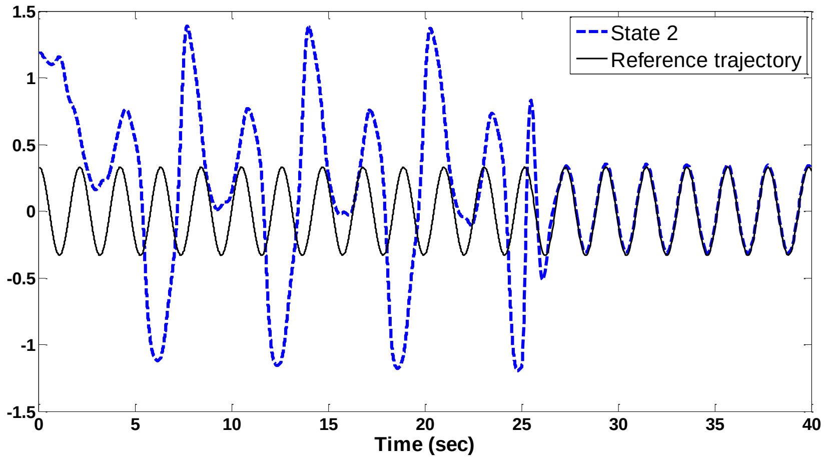

Fig. 4.6. Reference trajectory versus the second state of the robot manipulator systems

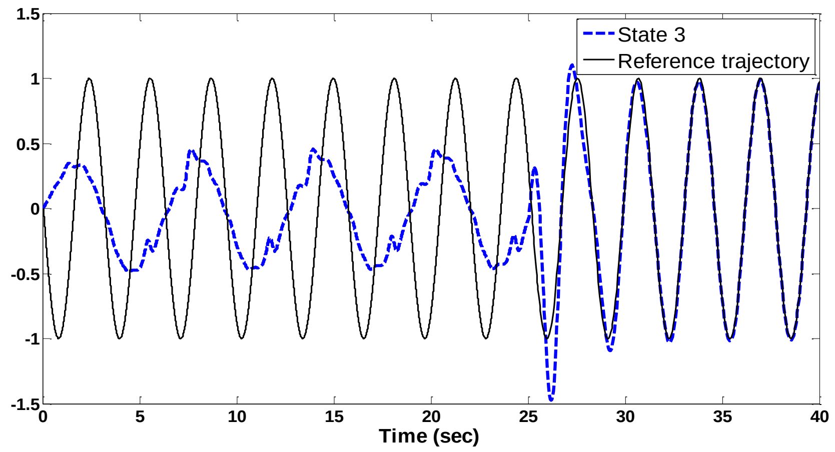

Fig. 4.7. Reference trajectory versus the third state of the robot manipulator systems

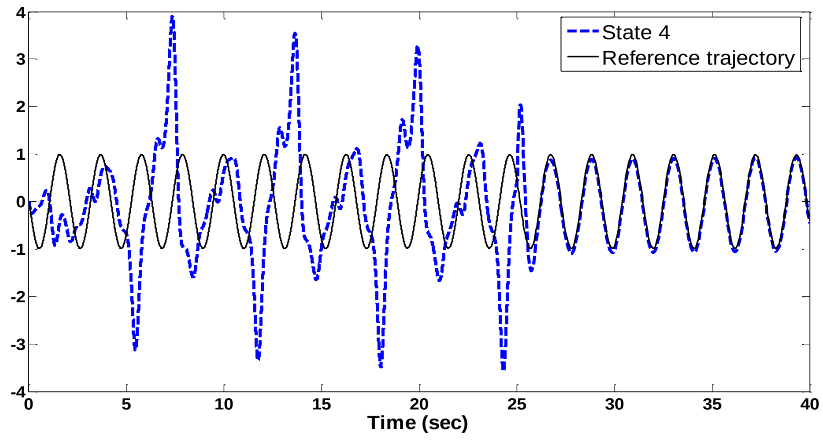

Fig. 4.8. Reference trajectory versus the fourth state of the robot manipulator systems

4.6. Conclusion

A model-free H_∞ tracker was developed for nonlinear continuous-time systems in the presence of disturbance. A generalized discounted L_2-gain condition was proposed for obtaining the solution to this problem in which the norm of the performance output includes both feedback and feedforward control inputs. This enables us to extend RL algorithms for solving the H_∞ optimal tracking problem without requiring complete knowledge of the system dynamics. A tracking HJI equation is developed to find the solution to the problem in hand. The stability and optimality of the resulting solution was analyzed and an upper bound for the discount factor was found to assure the stability of the control solution found by solving the tracking HJI equation. An online off-policy RL algorithm was proposed to learn the solution to the tracking HJI equation without requiring any knowledge of the system dynamics. It is shown that using off-policy RL, the disturbance input does not required being specified and adjusted. Simulation results confirmed the suitability of the proposed method.

Chapter 5

OPTIMAL DYNAMIC OUTPUT-FEEDBACK CONTROL DESIGN FOR UNKNON LINEAR SYSTEMS

5.1. Introduction

While significant progress has been achieved in the use of RL algorithms for the design of optimal controllers, these algorithms are limited to the case when full state of the controlled plant is available for measurement. In practice, however, all the states of the system are not always available for measurement. Therefore, the design of output-feedback (OPFB) controllers is required. Static OPFB has been studied in considerable details in the literature. However, guaranteed closed-loop stability cannot be achieved by using static OPFB. Nevertheless, information about the system is included in a long-enough set of input/output data and it would be desirable to design a state estimator by using input/output data without any system knowledge.

In this chapter, an online RL algorithm is developed to learn the optimal OPFB controller for linear CT systems. A discounted performance function is considered to make the proposed method applicable for solving both LQR and LQT problems. It is first shown that one can construct the system states form a limited number of measured system outputs over the past history of the trajectory. Then, a new Bellman equation is developed which gives both the value function and the updated policy corresponding to a control policy simultaneously using only measured system outputs over a period of the history of the system. An online RL algorithm is then developed using this Bellman equation which does not require the knowledge of neither the system dynamics nor the system state. Finally, convergence to the optimal control solution is shown.

This chapter is organized as follows. The next section provides background on optimal control problem of linear continuous time systems and the RL algorithm for solving this problem. Section 5.3 shows how to reconstruct the state using measured output. The proposed RL-based optimal OFPB control design method is presented in Section 5.4. It is shown that how the value function can be constructed based on measured output systems and how this value function can be used to develop an OFPB based RL algorithm. Sections 5.5 and 5.6 present the simulation results and conclusion, respectively.

5.2. Background

In this section, the optimal control of CT linear systems is formulated. A discounted performance function is used to make the proposed method applicable for solving both LQR and LQT problems. An offline PI algorithm and an online off-policy RL algorithm are presented to solve the problem.

Consider the linear CT system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{5.1}$$

where $x \in \mathbb{R}^{n\times n}$ is the system state vector, $y \in \mathbb{R}^{p\times 1}$ is the system output, $u \in \mathbb{R}^{m\times 1}$ is the control input, $A \in \mathbb{R}^{n\times n}$ gives the drift dynamics of the system, and $B \in \mathbb{R}^{n\times m}$ is the input matrix. It is assumed that the pair (A,B) is controllable and the pair (A,C) is observable.

The goal of the LQR problem is to find a control policy that makes the system (5.1) stable and minimizes a predefined performance function. Define the discounted performance function as

$$V(x(t)) = \int_t^\infty e^{-\gamma(\tau-t)}(y^T Q y + u^T R u)\; d\tau \tag{5.2}$$

where the state weight matrix Q is symmetric positive semi-definite and control input weight matrix R is symmetric positive definite. It is assumed that $(A, C\sqrt{Q})$ is observable.

Remark 5.1. The reason for using the general quadratic performance function with discount factor defined in (5.2) is that as is shown in Chapter 2, the LQT problem can be formalized as minimizing a discounted performance function subject to an augmented system in form of (5.1). Therefore, the results of this chapter can be used to solve both LQR and LQT problems online and without requiring any knowledge of the system dynamics or the system state.

Consider a fixed admissible state-feedback control policy as

$$u = -K\,x \tag{5.3}$$

It was shown in Chapter 2 that the value function for a control policy in form of (5.3) can be written as the quadratic form

$$V(x(t)) = \int_t^{\infty} e^{-\gamma(\tau - t)} x^T (C^T Q C + K^T R K)\, x\, d\tau = x(t)^T P\, x(t) \tag{5.4}$$

and the optimal control input is given by $u = -K^* x$ with

$$K^* = R^{-1} B^T P \tag{5.5}$$

where P is the solution to the ARE

$$A^T P + PA - \gamma P + C^T Q C - P\,B\,R^{-1} B^T P = 0 \tag{5.6}$$

In order to find the optimal state-feedback control solution, the ARE (5.6) is first needed to be solved for P, and then the optimal control gain is obtained by substituting the ARE solution to (5.5).

Using the same procedure as in Chapter 2, one has the following off-policy Bellman equation.

$$\begin{aligned}&e^{-\gamma T}x(t+T)^T P^i\, x(t+T)-x(t)^T P^i\, x(t)=\\&-\int_t^{t+T} e^{-\gamma(\tau-t)}x^T Q_i\, x\, d\tau+2\int_t^{t+T} e^{-\gamma(\tau-t)}(u+K^i x)^T R\, K^{i+1}x\, d\tau\end{aligned} \tag{5.7}$$

where $Q_i = C^T QC+(K^i)^T R(K^i)$. For a fixed control gain K^i, the above Bellman equation can be solved for both the value function kernel matrix P^i and the updated improved gain K^{i+1}, simultaneously. The following Algorithm 5.1 uses the above Bellman equation to iteratively solve the ARE equation (5.6).

Algorithm 5.1. Online Off-policy RL State-feedback algorithm

Initialization: Start with a control policy $u^0 = -K^0\, x+e$, where K^0 is stabilizing and e is the probing noise.

Policy evaluation and improvement: Solve the following Bellman equation for P^i and K^{i+1} simultaneously

$$\begin{aligned}&e^{-\gamma T}x(t+T)^T P^i\, x(t+T)-x(t)^T P^i\, x(t)=-\int_t^{t+T} e^{-\gamma(\tau-t)}x^T Q_i\, x\, d\tau+\\&2\int_t^{t+T} e^{-\gamma(\tau-t)}(u+K^i x)^T R\, K^{i+1}\, x\, d\tau\end{aligned} \tag{5.8}$$

Stop if a stopping criterion is met, otherwise set $i=i+1$ and go to 2.

5.3. State reconstruction using measured data

In this section, it is first shown that the system states can be observed using only a limited number of measured system outputs over the past history of the system trajectory for a fixed control policy. Then, using these observed system outputs, an online OPFB controller based on measured data is presented.

In this subsection, it is shown that for an observable linear CT system, the system states and consequently the value function can be expressed in terms of a limited number of measured system outputs over the past history of the system.

Suppose that at time *t*, we have a set of *N* output values from the history of the system and consider that they are stored in a history stack y_N. That is,

$$\mathrm{y}_N = \{y(t-h_i), i=0,...,N-1\} \tag{5.9}$$

where $h_i,\ \ i=0,...,N-1$ are the delayed values of the output and are assumed fixed. Consider that these *N* output samples are sampled from the system (5.1) at *N* time instances stored at vector τ_N as

$$\tau_N = \{t-h_i \geq 0, i=0,1,...,N-1\} \tag{5.10}$$

Definition 5.1 [118]. System (5.1) is said to be τ_N observable if $x(0)$ can be uniquely determined from an observations y_N on τ_N.

Definition 5.2 [118]. For a given time interval $[0,\bar{T}]$ and an integer $N_{\bar{T}}$, the system is said to be $N_{\bar{T}}$-sample observable if the system is $\tau_{N_{\bar{T}}}$ observable for any $\tau_{N_{\bar{T}}}$ with $0 \leq t-h_i \leq \bar{T}, i=1,...,N_{\bar{T}}$.

The following theorem shows that for the system dynamic (5.1), if (A,C) is observable, one can always find an integer N such that if $N > N_{\bar{T}}$ the system is $N_{\bar{T}}$-sample observable.

Theorem 5.1. Suppose the matrix A in (5.1) has eigenvalues $\lambda_j, j=1,...,n$. Denote $\delta \equiv \max_{1\leq i,j\leq n}\{\mathrm{im}(\lambda_i - \lambda_j)\}$, where $\mathrm{im}(\mathrm{Z})$ is the imaginary part of $\mathrm{Z} \in \mathbb{C}$. For a given interval $[0,\bar{T}]$, define

$$\mu_{\bar{T}} \equiv 2(n-1) + \frac{\bar{T}}{2\pi}\delta \tag{5.11}$$

Given (A, C) is observable, if $N_{\bar{T}} > \mu_{\bar{T}}$, then the system is $N_{\bar{T}}$ -sample observable.

Proof. See [118].

If the condition of Theorem 5.1 is satisfied, then the system state at each time can be calculated from the knowledge of the system output at *N* points in its history. The next Lemma shows that if the interval $\bar{T}$ is small enough, one can construct the system state using *n* previous values of the output, for an *n*-dimensional system.

Lemma 5.1 [118]. For any given *n*-dimensional observable system, there exists a sufficiently small time interval $[0, \bar{T}]$ such that if *n* sampling times $0 \leq t - h_i \leq \bar{T}, i = 1, ..., n$, then the system is *n*-sample observable.

Note that in the state-feedback off-policy RL Algorithm 5.1, during the evaluation of a control policy, the control policy is considered to be fixed and the knowledge of the system state is used to evaluate the policy. In the following, using Theorem 5.1, a formula is given by which the knowledge of the state needed to evaluate a fixed control policy in Algorithm 5.1 is obtained by the knowledge of the system output at *N* points in its history of using the control policy under evaluation. These *N* points are collected and stored in the history stack at reinforcement interval times $t - iT,\ i = 1, ..., N$. That is, in (5.10) we have $h_i = iT$ and hence

$$\tau_N = \{t - iT \geq 0, i = 0, 1, ..., N-1\} \tag{5.12}$$

Now, assume that the control policy which is applied to the system and is under evaluation is given by (5.3). Then, using (5.3) in (5.1), the closed-loop system dynamic becomes

$$\dot{x}(t) = (A - BK)\,x(t) \tag{5.13}$$

It is now shown that the state needed for solving the off-policy Bellman equation (5.8) can be expressed in terms of *N* measurements of the output in the history of using the control gain K. To this end, first the system state for every time instance stored in the vector τ_N is expressed in terms of the system state at current time *t*. In fact, since the control policy is considered to be fixed (which occurs during policy evaluation step of the off-policy RL algorithm), using the solution of (5.13), the state for an arbitrary time $t - iT$ with respect to the state for the current time t can be written as

$$x(t - iT) = e^{-iT(A-BK)}x(t) \tag{5.14}$$

The output $y(t - iT)$ can then be expressed as

$$y(t - iT) = C\,e^{-iT(A-BK)}x(t) \tag{5.15}$$

Suppose that at the current time t, a set of N output values $\Upsilon_N = \{y(t - h_i), i = 0,...,N-1\}$ are sampled at N time instances $\tau_N = \{t - h_i \geq 0, i = 0,1,...,N-1\}$ and stored in a history stack while the fixed control gain K is being evaluated. Then, using the output dynamics, one has

$$\begin{bmatrix} y(t) \\ y(t-T) \\ \vdots \\ y(t-(N-1)T) \end{bmatrix} = \begin{bmatrix} C \\ Ce^{-T(A-BK)} \\ \vdots \\ Ce^{-(N-1)T(A-BK)} \end{bmatrix} x(t) \tag{5.16}$$

Define

$$\overline{y}_t = \left[y(t)^T \quad y(t-T)^T \quad \ldots \quad y(t-(N-1)T)^T \right]^T \tag{5.17}$$

$$G = \left[C^T \quad e^{-T(A-BK)^T} C^T \quad \ldots \quad e^{-NT(A-BK)^T} C^T \right]^T \tag{5.18}$$

where $\overline{y}_t \in \mathbb{R}^{pN\times 1}$ and $G \in \mathbb{R}^{pN\times n}$ with p the dimension of the output. Then, (5.16) becomes

$$\overline{y}_t = G\,x(t) \tag{5.19}$$

If the system (5.1) is observable and the number of samples *N* is larger than μ_T defined in (5.11), then based on Theorem 5.1 the system is N-sample observable. Therefore, G is full rank. Thus, from (5.16), the system state vector is given by

$$x(t) = G_N\,\overline{y}_t = [L_1,...,L_N]\,\overline{y}_t = \sum_{i=1}^{N} L_i\, y(t-iT) \tag{5.20}$$

where

$$G_N = (G^T G)^{-1} G^T \in \mathbb{R}^{n\times pN}$$

and

$$L_i = G_N(1:n,(i-1)p+1:ip).$$

Equation (5.20) shows that if the system is observable, one can construct the system state needed to evaluate the Bellman equation uniquely using a limited number of the measured system outputs in the history of using the given control policy.

Note that the system dynamics information A, B, and C must be known to construct the system state form the measurement system outputs. In fact, G depends on A, B, and C. In the next step, it is shown how to use the structural dependence in (5.20) yet avoid knowledge of the system dynamics. We first show that the value function

(5.17) can be expressed as a quadratic form in terms of a limited number of measured system outputs in the history of the system. Using (5.20) in (5.4) gives

$$V(t) = x(t)^T P\, x(t) = (G_N\, \bar{y}_t)^T P\, (G_N\, \bar{y}_t) = \bar{y}_t^{\,T} G_N^{\;T} P\, G_N \bar{y}_t \tag{5.21}$$

where the last equality is obtained using $G_N = (G^T G)^{-1} G^T$. This equation is equation can be written as

$$V(t) = \bar{y}_t^{\,T} \bar{P}\, \bar{y}_t \tag{5.22}$$

where

$$\bar{P} = G_N^T P\, G_N \in \mathbb{R}^{pN \times pN}$$

and is constant. Using (5.22), (5.17) becomes

$$\bar{y}_t^{\,T} \bar{P}\, \bar{y}_t = \int_t^\infty e^{-\gamma(\tau - t)} (y^T Q\, y + u^T R\, u)\; d\tau \tag{5.23}$$

Note that the matrix $\bar{P}$ depends on the system dynamics *A, B*, and *C*. In the next section, it is shown how to use RL methods to learn this matrix without knowing the system drift dynamics *A*.

5.4. Model-free RL algorithm using measured data

In this subsection, a model-free OPFB IRL algorithm is developed. First, an IRL Bellman equation is developed using measured system outputs that is equivalent to Algorithm 5.1, which requires full state measurement.

Algorithm 5.1 is a model-free IRL algorithm in which the policy u which is applied to the system can be different that the policy $u^i = K^i x$ which is updated and evaluated. In this chapter, we assume that u is the updated policy plus a probing noise. That is

$$u = K^i x + w \tag{5.24}$$

where w is the probing noise. Based on (5.20), the relation between the state-feedback and output-feedback control gains is obtained by

$$u^i = K^i x = K^i G^i{}_N \, \bar{y}_t = \bar{K}^i \, \bar{y}_t \tag{5.25}$$

with $G^i{}_N = ((G^i)^T G^i)^{-1} (G^i)^T \in \mathbb{R}^{n \times pN}$ and

$$G^i = \begin{bmatrix} C^T & e^{-\delta t (A+BK^i)^T} C^T & \dots & e^{-N\delta t (A+BK^i)^T} C^T \end{bmatrix}^T \tag{5.26}$$

Note that $\bar{K}^i = K^i \, G^i{}_N$ is a nonlinear function of the state-feedback gain K^i and the system dynamics. The key equation (5.7) in Algorithm 5.1, which uses the state information to evaluate both value function and control policy, can be written in terms of the measured outputs as

$$\begin{aligned} & e^{-\gamma \, \delta t} \bar{y}_{t+\delta t}{}^T \bar{P}^i \, \bar{y}_{t+\delta t} - \bar{y}_t{}^T \bar{P}^i \, \bar{y}_t = -\int_t^{t+\delta t} e^{-\gamma(\tau - t)} \bar{y}_t{}^T \bar{Q}_i \, \bar{y}_t \, d\tau \\ & +2\int_t^{t+\delta t} e^{-\gamma(\tau-t)} \, w^T R \, \bar{K}^{i+1} \bar{y}_t d\tau \end{aligned} \tag{5.27}$$

where $\bar{Q} = [\underline{1} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]^T \, Q \, [\underline{1} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]$.

We now use this OPFB Bellman equation to present an optimal model-free OPFB control design method as follows.

Algorithm 5.2. Model-free RL based OPFB Control design algorithm

1. *Initialization*: Start with a stabilizing control policy $u^0 = \bar{K}^0 \, \bar{y}_t$,

2. Solve the following Bellman equation for $\bar{P}^i$ and $\bar{K}^{i+1}$ simultaneously

$$\begin{aligned} & e^{-\gamma \, \delta t} \bar{y}_{t+\delta t}{}^T \bar{P}^i \, \bar{y}_{t+\delta t} - \bar{y}_t{}^T \bar{P}^i \, \bar{y}_t = -\int_t^{t+\delta t} e^{-\gamma(\tau - t)} \bar{y}_t{}^T \bar{Q}_i \, \bar{y}_t \, d\tau \\ & +2\int_t^{t+\delta t} e^{-\gamma(\tau-t)} \, w^T R \, \bar{K}^{i+1} \bar{y}_t \, d\tau \end{aligned} \tag{5.28}$$

3. Stop if a stopping criterion is met, otherwise set $i = i + 1$ and go to 2.

Algorithm 5.2 does not require any knowledge of the system dynamics or the system states. In fact the requirement of the system dynamics and system states are replaced by the input and output information measured online. The solution $\bar{P}^i$ and $\bar{K}^{i+1}$ to (5.28) can be found using the least square methods.

Theorem 5.2. Algorithm 5.2 converges to the optimal OPFB control gain $\bar{K}^*$ and value function kernel matrix $\bar{P}^*$. Moreover, if condition of Theorem 5.1 is satisfied, one has $u^* = \bar{K}^*\bar{y}_t = K^* x$ where K^* is given in (5.5) and satisfies the state-feedback ARE (5.6). That is, the optimal OPFB solution gives the optimal state-feedback solution.

Proof. Using (5.25), one has

$$\bar{K}^{i+1} = K^{i+1}G^i{}_N = -R^{-1}B^T P^i G^i{}_N \tag{5.29}$$

Dividing both sides of (5.28) by δt and taking limit yields

$$\begin{aligned}&\lim_{\delta t\to 0}\frac{e^{-\gamma\,\delta t}\,\bar{y}_{t+\delta t}{}^T\bar{P}^i\,\bar{y}_{t+\delta t} - \bar{y}_t{}^T\bar{P}^i\,\bar{y}_t}{\delta t} + \lim_{\delta t\to 0}\frac{\displaystyle\int_t^{t+\delta t} e^{-\gamma(\tau-t)}\bar{y}_t{}^T\bar{Q}_i\,\bar{y}_t\,d\tau}{\delta t} + \\ &\lim_{\delta t\to 0}\frac{2\displaystyle\int_t^{t+\delta t} e^{-\gamma(\tau-t)}(u-\bar{K}^i\bar{y}_t)^T R\,\bar{K}^{i+1}\bar{y}_t\,d\tau}{\delta t} = 0\end{aligned} \tag{5.30}$$

By L'Hopital's rule, this equation becomes

$$-\gamma\,\bar{y}_t{}^T\bar{P}^i\,\bar{y}_t + \dot{\bar{y}}_t{}^T\bar{P}^i\,\bar{y}_t + \bar{y}_t{}^T\bar{P}^i\,\dot{\bar{y}}_t + \bar{y}_t{}^T\bar{Q}_i\,\bar{y}_t - 2(u-\bar{K}^i\bar{y}_t)^T R\,\bar{K}^{i+1}\bar{y}_t = 0 \tag{5.31}$$

On the other hand, by differentiating (5.19), one has

$$\dot{\bar{y}}_t = G\,\dot{x}(t) = G(Ax(t)+Bu(t)) = GAG_N\,\bar{y}_t + GB\,u(t) \tag{5.32}$$

Using the system dynamics(5.32) and the updated law (5.25) in (5.32) gives

$$G_N{}^T[(A+BK^i)^T P^i + P^i(A+BK^i) - \gamma P^i + C^T QC + (K^i)^T R(K^i)]G_N = 0 \tag{5.33}$$

Since G_N is full rank, the state-feedback Lyapunov equation is satisfied. That is, evaluating a fixed OPFB control policy $u = \bar{K}^i \bar{y}_t$ using the Bellman equation (5.28) gives the same value function as evaluating the fixed state-feedback control policy $u = K^i x(t)$, with $\bar{K}^i = K^i \, G^i_{\ N}$, using the state-feedback Lyapunov equation. Moreover, the policy improvement $\bar{K}^{i+1}$ in terms of K^{i+1} becomes $K^{i+1} = -R^{-1} B^T P^i$. Therefore, Algorithm 5.2 give the same results for policy evaluation and improvement steps as the state-feedback RL-based algorithmpresented in Chapter 2, and thus have the same convergence properties. This confirms that the proposed OPFB design method converges to an optimal solution and gives a state-feedback control. □

Remark 5.2. The proposed control input is more powerful than the static OPFB in form of $u = K\,y(t)$. In fact, as shown in proof of Theorem 5.2, the proposed control input is equivalent to a state-feedback control input as a result of using the delayed outputs. Therefore, in contrast to the static OPFB, using the proposed controller one can stabilize a system which is state-feedback stabilizable but are not static OPFB stabilizable. Simulation results confirm this statement.

Remark 5.3. It is interesting to compare the form of the proposed control input with the control obtained using the fast output sampling technique [50], [127]. In this technique, similar to the proposed control law, the control input is a linear combination of observation of the last *N* output samples. The problem is to find a fast output sampling feedback gain that realizes this state feedback gain matrix. Unlike static output feedback, fast output sampling technique guarantees the stability of the closed-loop system, as long as the system is controllable.

5.5. Simulation results

In this section, two simulation examples are provided to show the suitability of the proposed method. The first example is a LQR problem for a system which is not static OPFB stabilizable, yet is stabled by our proposed method. The second system is a LQT problem for F-16 aircraft system.

5.5.1. OPFB regulator for a system which is not OPFB stabilizable

Consider the dynamical systems as

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= [1 \quad 0] x\end{aligned} \tag{5.34}$$

where $x = [x_1, x_2]^T$. The system (5.34) is both controllable and observable. However, it is not static OPFB stabilizable. That is, there is no gain k such that the control input $u = k\,y$ make the system asymptotically stable. In the following, we show that although this system is not static OPFB stabilizable, we can stabilize it using the proposed OPFB design method. In order to show the suitability of the proposed OPFB controller, its results are compared to the results of the optimal state-feedback controller. The discount factor is consider ed as $\gamma = 0$. The weighting matrices in the performance function are chosen as $Q = R = 1$. The optimal state-feedback control is given by

$$u = -0.414\,x_1(t) - 0.910\,x_2(t) \tag{5.35}$$

We now use Algorithm 5.2 to find the optimal OPFB gain. The reinforcement interval time is considered as $\delta t = 0.2$ and the number of stored data in the history stack is 4. That is, the control input $u(t)$ is constructed form the current output $y(t)$ and the past outputs $y(t - 0.2\,\delta t)$, $y(t - 0.4\,\delta t)$ and $y(t - 0.6\,\delta t)$. A probing noise is added to the control input to persistently excite the system output. Define

$\bar{y}_t = [y(t) \quad y(t-0.2) \quad y(t-0.4) \quad y(t-0.6)]$ and assume $u(t) = K\bar{y}_t$. Fig. 5.1 shows convergence of the control gain K. In fact the OFFB gain converges to $K = [k_1, \ldots, k_4] = [4.8950, -11.3513, 8.4717, -1.7177]$. The optimal OPFB policy is then given by

$$u = 4.8950\,y(t) - 11.3513\,y(t-0.2) + 8.4717\,y(t-0.4) - 1.7177\,y(t-0.6) \quad (5.36)$$

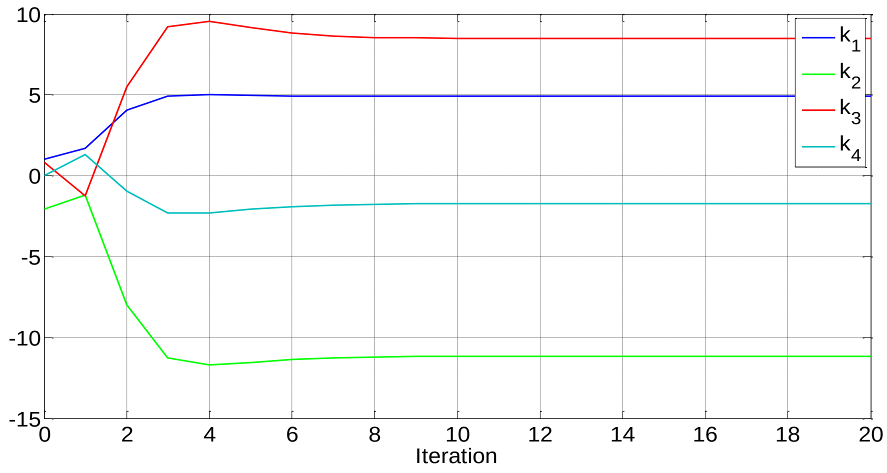

Fig. 5.1. Convergence of OPFB control gains for the LQR problem

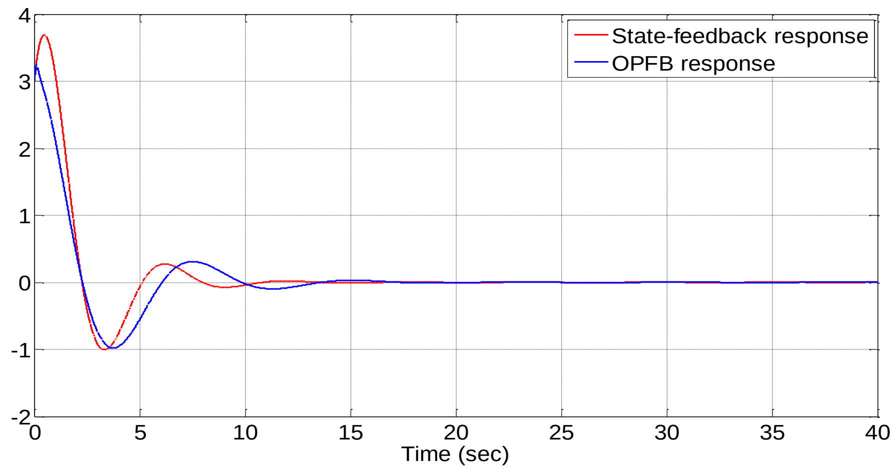

Fig. 5.2. Comparing the performance of the state-feedback and OPFB controllers for the LQR problem

Fig. 5.2 shows the system output for the OPFB control policy and the optimal state-feedback control, starting from the system the initial condition $x=[3,\ 3]^T$. From this Fig, it is obvious that the performance of the OPFB controller and the optimal state-feedback controller are close to each other.

5.5.2. OPFB for F-16 aircraft system

Consider the F-16 aircraft system described by

$$\dot{x}_a = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} x_a + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \qquad (5.37)$$

The system state vector is $x_a = [x_1 \quad x_2 \quad x_3] = [\alpha \quad q \quad \delta_e]$, where α denotes the angle of attack, q is the pitch rate, and δ_e is the elevator deflection angle. It is assumed that the output is $y=\alpha$ and the desired value is constant. Thus, the command generator dynamics become $\dot{r}=0$ and thus the augmented system becomes

$$\dot{x} = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & 0 \\ 0.82225 & -1.07741 & -0.17555 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \qquad (5.38)$$

where the augmented state is $x=[x_a, r]$. The performance function is considered as

$$V(t) = \int_t^{\infty} e^{-\gamma(\tau - t)}((y-r)^T Q (y-r) + u^T R u)\ d\tau \qquad (5.39)$$

with $Q=1$, $R=0.1$ and $\gamma=0.01$. This performance function in terms of the augmented state becomes

$$V(t) = \int_t^{\infty} e^{-\gamma(\tau - t)}(x^T Q_T\, x + u^T R u)\ d\tau \qquad (5.40)$$

with $Q_T = [1 \quad 0 \quad 0 \quad -1]^T Q [1 \quad 0 \quad 0 \quad -1]$.

The optimal state-feedback control input is given by

$$u = 14.7060\, x_1(t) + 10.8294\, x_2(t) - 1.2055 x_3(t) - 31.5253 r(t) \tag{5.41}$$

We now use Algorithm 5.2 to find the optimal OPFB gain. The reinforcement interval time is considered as $\delta t = 0.2$ and the number of stored data in the history stack for constructing the state is 3. That is, the control input $u(t)$ is constructed form $\bar{y}_t = [y(t) \quad y(t-0.1) \quad y(t-0.2)]$, and the reference signal $r(t)$. The performance function is then quadratic in terms of $\bar{x} = [\bar{y}_t, r]$ and it is assumed that $u(t) = -K\,\bar{x}$. Fig. 5.3 shows the convergence of K. In fact, this gain converges to $K = [k_1, ..., k_4] = [-9.075, 1.637, 0.185, 10.165]$. The optimal OPFB policy is then given by

$$u = 9.0754 y(t) - 1.6371 y(t-0.2) - 0.1849 y(t-0.2) - 9.9965\, r(t) \tag{5.42}$$

Fig 5.4 shows the system output for the OPFB control policy and the optimal state-feedback control, assuming that the desired value is $r(t) = 1$. These results confirm that the proposed model-free optimal OPFB controller has a performance close to the optimal state-feedback controller.

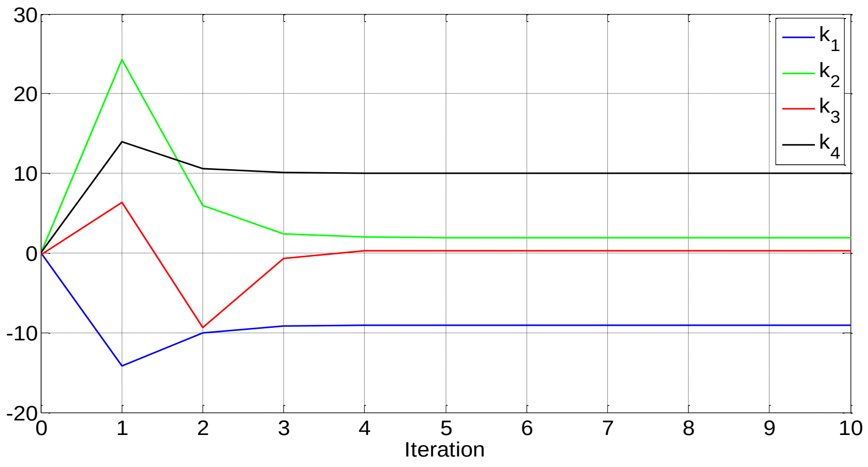

Fig. 5.3. Convergence of OPFB control gains for F-16 system

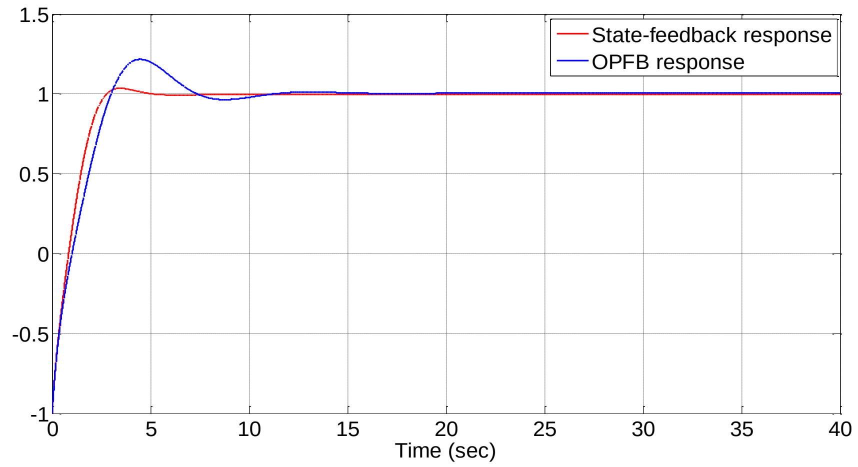

Fig. 5.4. Comparing the performance of the state-feedback and OPFB controllers for F-16 system

5.6. Conclusion

An off-policy RL based method was presented to learning the optimal control law for linear continuous-time systems using only measured outputs. The proposed method did not require the knowledge of the system dynamics or the system state. An off-policy Bellman equation was developed to evaluate a control policy and find an improved policy simultaneously using only measured outputs. An off-policy RL algorithm was then developed which converged to the optimal control solution using only measured output data. The proposed method was tested on a simulation example.

Chapter 6

OPTIMAL MODEL-FREE SOLUTION TO OUTPUT SYNCHRONIZATION OF HETEROGENEOUS MULTI-AGENT SYSTEMS

6.1. Introduction

Cooperative control of multi-agent systems has undergone a paradigm shift from centralized to distributed, due to reliability, flexibility, scalability and computational efficiency of distributed control systems. In distributed control, unlike centralized control, there is no central authority with the ability to control the network of agents as a whole. Instead, each agent designs a controller based on limited information about itself and its neighbors to assure all agents reach agreement on certain quantities of interests. If the common value that agents agree on is not prescribed, the problem is called leaderless consensus, and if all agents follow trajectories of a leader node, the problem is known as cooperative tracking (leader-follower) control. A rich body of literature has been developed on distributed control of multi-agent systems. See for example [48], [69], [85][92], [93] to name a few.

Most of the existing work on distributed control focuses on state synchronization of homogeneous multiagent systems, where individual agents have identical dynamics. In many real-world applications of multi-agent systems, however, individual systems do not have identical dynamics. This has led to the emergence of new challenges in the design of distributed controllers for heterogeneous systems, in which the dynamics and dimension of agents can be different. Since state synchronization is not practical for general heterogeneous systems (as individual systems may have different states and state dimensions), distributed output synchronization of heterogeneous systems has attracted compelling attention in the literature [39][42][39][41]. Existing mentioned methods, however, require complete knowledge of the agents and the leader dynamics which is not available in many real-world applications. In practical applications, it is often desirable to design model-free distributed controllers conducive to real time implementation and able to handle modeling uncertainties in dynamics of agents. Moreover, solutions found by these methods are generally far from optimal.

Adaptive and robust distributed controllers have been developed in the literature to adapt online to modeling uncertainties in the dynamics of the agents. However, classical adaptive and robust distributed controllers do not converge to an optimal distributed solution. Optimal distributed control refers to a class of methods that can be used to synthesize a distributed control policy which results in best possible team behavior with respect to prescribed criteria (i.e. local control policies which leads to minimization of local performances for each agent). A suboptimal distributed controller is designed in [140] for linear homogenous systems using linear quadratic regulator. The distributed games on graphs are presented in [110] in which each agent only minimizes its own performance index. The distributed inverse optimal control is also considered in [139]. All mentioned optimal distributed controllers are limited to state synchronization of homogeneous systems and they require complete knowledge of the agents and the leader. To our knowledge distributed adaptive optimal output synchronization is not considered in the literature.

In this chapter, a novel RL algorithm is developed to solve the output synchronization problem of heterogeneous multi-agent systems. It is shown that the explicit solution to the output regulator equation is not necessary, hence the agents do not need to know the leader’s dynamics. The key components of the given method are

- A distributed adaptive observer is designed to estimate the leader's state. This observer does not require the knowledge of the leader dynamics.
- A novel off-policy RL algorithm is developed to solve the output synchronization problem without requiring any knowledge of the agent's dynamics or the leader's dynamics.
- It is shown that this distributed RL approach implicitly satisfies the output regulation equations without actually solving them.

The proposed approach is detaled as follows. The estimated leader state obtained from the presented distributed observer is used along with the local state of each agent to design a model-free optimal output synchronization controller for each agent so as to track the output of an exo-system i.e., the leader in an optimal manner. To this end, the optimal output synchronization problem is cast into a set of optimal output tracking problems for each agent. A local discounted performance function is defined for each agent in which its minimization gives both feedback and feedforward gains. Online solution to the tracking problem is then found by using an off-policy RL algorithm. This algorithm does not require any knowledge of the dynamics of the agents and uses only the measured data along the system and the reference trajectories to find the optimal distributed solution to the output synchronization problem.

The rest of this chapter is organized as follows. In Section 6.2, the essential theoretical background is provided. A distributed adaptive observer is designed in Section 6.3. The off-policy RL algorithm is employed in Section 6.4 to solve the output synchronization problem. It is shown in Section 6.5 that the well-known separation principle is satisfied and thus the observer and the controller design problem can be treated separately. In Section 6.6, the simulation results for output tracking of multi-agent heterogeneous systems are given. Section 6.7 concludes this chapter.

6.2. Thechnical background

In this section, the essential theoretical background on graph theory is provided. The problem of output synchronization for heterogeneous multi-agent systems is also defined. The standard solution to this problem is presented and its shortcoming is emphasized.

Consider a weighted directed graph or digraph $\mathcal{G}$ = ($\mathcal{V}$,$\mathcal{E}$,$\mathcal{A}$) consisting of a nonempty finite set of N nodes $\mathcal{V}$ = (v_1, v_2, ..., v_N), a set of edges or arcs $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ and the associated adjacency matrix $\mathcal{A}$ = [a_{ij}] $\in \mathbb{R}^{N\times N}$. Here the diagraph is assumed to be time-invariant or alternatively we assume   to be constant. An edge from a node v_j to v_i is indicated by an arrow with head at node i and tail at node j, this implies that the information flow is from node j to node i. The neigh bor set of node i is depicted by N_i = {j | (v_j, v_i) $\in \mathcal{E}$}. For each node the entry a_{ij} of the adjacency matrix $\mathcal{A}$ is nonzero (i.e., a_{ij} > 0) if and only if there is an edge (v_j, v_i) $\in \mathcal{E}$ else a_{ij} = 0, also a_{ij} indicates the weight associated with the graph edge. We consider simple graphs without self-loop, this means a_{ii} = 0. The in-degree of a node i is defined as d_i = $\sum_{j=1}^{N} a_{ij}$ and in-degree matrix as $D = \operatorname{diag}\left[d_i\right] \in \mathbb{R}^{N\times N}$, then the graph Laplacian matrix is defined as $L = D - \mathcal{A}$. For $\mathbf{1}_N = \left[1,1,\cdots,1\right] \in \mathbb{R}^N$, then $L\mathbf{1}_N = 0$. The out-degree of a node i is defined as $d_i^o = \sum_{j=1}^{N} a_{ji}$, a graph is said to be balanced if its in-degree is same as the out-degree, this implies $L^T\mathbf{1}_N = 0$. For a given digraph $\mathcal{G}$ a sequence of successive edges in the form $(v_i, v_k), (v_k, v_l), \cdots (v_m, v_j)$ is a directed path from node i to node j. A diagraph is said to have a spanning tree if there exist a root node i_r, such that there is a directed path from i_r to every other node in the graph.

Assumption 6.1: The digraph $\mathcal{G}$ has a spanning tree and the leader is pinned to the root node i_r , with a pinning gain $g_i > 0$.

Note that the leader can be pinned to multiple nodes in graph or the leader can itself be a root node. This results in a diagonal pinning matrix $G = \mathrm{diag}\left[g_i\right] \in \mathbb{R}^{N\times N}$ with the pinning gain $g_i > 0$ if the node has access to the leader else otherwise zero. Under the above assumption, the eigenvalues of $L+G$ have positive real parts.

The output synchronization problem is now reviewed. Consider the dynamics of leader or trajectory generator to be followed is

$$\dot{\zeta}_0 = S\zeta_0 \tag{6.1}$$

where $\zeta_0 \in \mathbb{R}^p$ is the reference trajectory, and $S \in \mathbb{R}^{p\times p}$ is the leader dynamic matrix. The leader output can be defined as

$$y_0 = R\zeta_0 \tag{6.2}$$

where $y_0 \in \mathbb{R}^q$.

Assumption 6.2: The leader dynamic matrix is marginally stable.

The dynamics of N linear heterogeneous agents is given by

$$\begin{aligned} \dot{x}_i &= A_i x_i + B_i u_i \\ y_i &= C_i x_i \end{aligned} \tag{6.3}$$

where $x_i \in \mathbb{R}^{n_i}$ is the system state, $u_i \in \mathbb{R}^{m_i}$ is the input and $y_i \in \mathbb{R}^q$ is the output for $i = 1,\cdots,N$. The multi-agent system is called heterogeneous because agents dynamics (A_i, B_i, C_i) and the dimension of their states are generally not the same.

Assumption 6.3. (A_i, B_i) is stabilizable and (A_i, C_i) is observable.

Problem 6.1 (Output Synchronization): Design local control protocols u_i such that the outputs of all heterogeneous agents synchronize to the output of the leader node. That is, $y_i(t) - y_0(t) \to 0, \forall i$.

To solve this problem, standard methods in the literature requires solving the output regulation equations given by

$$\begin{aligned} A_i\Pi_i + B_i\Gamma_i &= \Pi_i S \\ C_i\Pi_i &= R \end{aligned} \tag{6.4}$$

where $\Pi_i \in \mathbb{R}^{n_i \times p}$ and $\Gamma_i \in \mathbb{R}^{m_i \times p}$ for $i = 1, \cdots, N$ are the solution of the output regulator equation (6.4). Based on these solutions, the following standard controller guarantees output tracking among heterogeneous agents [7], [10].

$$u_i = K_{1i}(x_i - \Pi_i\zeta_0) + \Gamma_i\zeta_0 \tag{6.5}$$

where $K_{1i} \in \mathbb{R}^{m_i \times n_i}$ is the state-feedback gain which stabilizes $A_i + B_i K_{1i}$. The tracking control law (6.5) depends on the agent state and the leader state. However, the leader state ζ_0 is not available to all agents in a distributed multi-agent network. This issue is circumvented in the literature by designing the following local observer called synchronizer in order to obtain an estimate of the leader trajectory in all the agents

$$\dot{\zeta}_i = S\zeta_i + c\left[\sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i)\right] \tag{6.6}$$

resulting in the modified tracking law

$$u_i = K_{1i}(x_i - \Pi_i\zeta_i) + \Gamma_i\zeta_i \tag{6.7}$$

where ζ_i is the estimation of ζ_0 for agent i and constant c is the coupling gain. The output synchronization is guaranteed when the control protocol (6.7) is applied to the multi-agent system.

Remark 6.1. Note that the solution to the output regulator equation (6.4) for each agent requires the complete knowledge of the leader dynamics, i.e., S, which is overly conservative. Moreover, all agents need to be aware of their own dynamics, i.e. (A_i, B_i, C_i), to solve the output regulator equation (6.4) and to obtain feedforward component K_{1i}. This knowledge, however, is not available in many applications.

<u>6.3. Distributed adaptive observer</u>

In the previous section, a standard solution to output regulation for heterogeneous multi-agent systems was given. The standard approach requires the solution of the output regulator equations (6.4). This needs full knowledge of the leaders dynamics (S,R), and the agent's dynamics (A_i, B_i, C_i).

In this section, a novel distributed adaptive observer is designed to estimate the leader state for all the agents. In contrast to standard observer (6.6), the proposed method does not require the knowledge of the leader dynamic matrix S. In the next section, it is shown how to use this adaptive observer along with reinforcement learning to solve problem 6.1 without solving the regulator equations (6.4) and without knowing the agent's dynamics.

To estimate the leader state, the following distributed observer is used.

$$\dot{\zeta}_i = \hat{S}_i \zeta_i + c\left[\sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i)\right] \tag{6.8}$$

where $\hat{S}_i \in \mathbb{R}^{p \times p}$ is the estimation of the leader dynamic matrix S for node i.

Tthe local neighborhood observation error for node i is defined as

$$e_i = \sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i) \tag{6.9}$$

Based on the observer (6.8) and the local estimation error (6.9), the following theorem provides a tuning law for $\hat{S}_i$ and shows the convergence of ζ_i to ζ_0. This proves the convergence of global estimation error $\delta_i(t)$, defined as follows, to zero.

$$\delta_i(t) = \zeta_i(t) - \zeta_0(t) \tag{6.10}$$

Assumption 6.4. The communication graph for the multi-agent heterogeneous systems is balanced.

Theorem 6.1. Let Assumption 6.1, 6.2 and 6.4 be satisfied. Consider the distributed observer given in (6.8). Design the tuning law for $\hat{S}_i$ as follows.

$$\dot{\hat{S}}_{\mathrm{vec}i} = -\Gamma_{Si}(I_q \otimes \zeta_i)\left[\sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i)\right] \tag{6.11}$$

where $\hat{S}_{\mathrm{vec}i}$ is the vector representation of $\hat{S}_i$, and Γ_{Si} is diagonal positive update rate matrix. Then, the observer estimation error (6.10) converges to zero asymptotically fast, provided the constant c in (6.8) is chosen large enough.

Proof. Differentiating (6.8) gives the error dynamics

$$\dot{\delta}_i = \hat{S}_i\zeta_i + c\left[\sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i)\right] - S\zeta_0 \tag{6.12}$$

which can be rewritten as

$$\dot{\delta}_i = S\zeta_i + c\left[\sum_{j=1}^{N} a_{ij}(\zeta_j - \zeta_i) + g_i(\zeta_0 - \zeta_i)\right] - S\zeta_0 + (\hat{S}_i - S)\zeta_i \tag{6.13}$$

The global error dynamics then becomes

$$\dot{\delta} = \left[I_N \otimes S - c(L+G) \otimes I_p\right]\delta + \mathrm{diag}(\tilde{S}_i)\zeta \tag{6.14}$$

or equivalently,

$$\dot{\delta} = \left[I_N \otimes S - c(L+G) \otimes I_p\right]\delta + \mathrm{diag}(I_q \otimes \zeta_i)^T \tilde{S}_{\mathrm{vec}} \tag{6.15}$$

where $\tilde{S}_{\text{vec}}$ is the vector representation of the error in leader dynamics estimation. Equation (6.15) in compact form is

$$\dot{\delta} = A_S \delta + \zeta_M \tilde{S}_{\text{vec}} \tag{6.16}$$

where

$$A_s = I_N \otimes S - c(L+G) \otimes I_p \tag{6.17}$$

$$\zeta_M = \operatorname{diag}(I_q \otimes \zeta_i)^T \tag{6.18}$$

The error dynamics matrix A_s defined in (6.17) can be made Hurwitz for an appropriate choice of the constant *c*, because $L+G$ is nonsingular and has eigenvalues with positive real part.

Now consider the Lyapunov function

$$V = \delta^T[(L+G)\otimes I_q + (L+G)^T \otimes I_q]\delta + \tilde{S}_{\text{vec}}^T \Gamma_S^{-1} \tilde{S}_{\text{vec}} = \delta^T P \delta + \tilde{S}_{\text{vec}}^T \Gamma_S^{-1} \tilde{S}_{\text{vec}} \tag{6.19}$$

where P is positive definite, and Γ_S is a diagonal positive definite scaling matrix. The derivative of the Lyapunov function is

$$\begin{aligned}\dot{V} &= \dot{\delta}^T P \delta + \delta^T P \dot{\delta} + \dot{\tilde{S}}_{\text{vec}}^T \Gamma_S^{-1} \tilde{S}_{\text{vec}} + \tilde{S}_{\text{vec}}^T \Gamma_S^{-1} \dot{\tilde{S}}_{\text{vec}} = \delta^T (PA_S + A_S^T P)\delta + \tilde{S}_{\text{vec}}^T \zeta_M^T P \delta \\ &+ \delta^T P \zeta_M \tilde{S}_{\text{vec}} + \dot{\tilde{S}}_{\text{vec}}^T \Gamma_S^{-1} \tilde{S}_{\text{vec}} + \tilde{S}_{\text{vec}}^T \Gamma_S^{-1} \dot{\tilde{S}}_{\text{vec}}\end{aligned} \tag{6.20}$$

By choosing

$$\dot{\tilde{S}}_{\text{vec}} = \dot{\hat{S}}_{\text{vec}} = -\Gamma_S \zeta_M^T P \delta \tag{6.21}$$

the Lyapunov derivative becomes

$$\dot{V} = \delta^T (PA_S + A_S^T P)\delta \tag{6.22}$$

In order to demonstrate the negative semi-definiteness of $\dot{V}$, we need to show $PA_S + A_S^T P < 0$. Let us redefine $I_N \otimes S = M$ and $(L+G) \otimes I_q = N$, where *M* has no eigenvalues with positive real parts, and *N* is non-singular. The Lyapunov equation in compact form is

$$\begin{aligned} & PA_S + A_S^T P = (N+N^T)(M-cN) + (M^T - cN^T)(N+N^T) \\ & = (NM + M^T N^T) - c(NN + N^T N^T) + (N^T M + M^T N) - c(N^T N + N^T N) \end{aligned} \tag{6.23}$$

It is well known that for any given Hermitian matrices E, F, and some constant *c*, the eigenvalue of the matrix sum is

$$\lambda_{i+j-1}(E-cF) \le \lambda_i(E) - c\lambda_j(F) \tag{6.24}$$

for $i+j \le N+1, i \le N, j \le N$. Thus, if *c* is greater than a certain bound, one can ensure that the eigenvalues of matrix sum $E-cF$ have negative real part.

Note that the two terms in (6.23) are of form (6.24), hence the eigenvalues of the terms

$$(NM + M^T N^T) - c(NN + N^T N^T)$$

and

$$(N^T M + M^T N) - c(N^T N + N^T N)$$

have negative real parts provided that *c* is large enough. Additionally, the overall matrix $PA_S + A_S^T P$ is symmetric, as it is obtained by addition/subtraction of the symmetric matrices. This confirms that the eigenvalues of $PA_S + A_S^T P$ are negative real and thus proves negative definiteness of $PA_S + A_S^T P$ and hence negative semi-definiteness of the Lyapunov derivative (6.23). That is,

$$\dot{V} = -\delta^T Q \delta \le 0 \tag{6.25}$$

for some $Q \geq 0$. This shows the convergence of the local synchronizer, i.e., $\delta_i \to 0$.

On the other hand, since the scaling matrix was chosen to be block diagonal, also using the well-known equality $L 1_N = L^T 1_N = 0$ for the balanced graph, the update rule (6.22) gives (6.12) which completes the proof.

Remark 6.2. Theorem 6.1 provides the proof of convergence for the local synchronizer i.e., $(\delta_i, \hat{S}_i) \to (0, S^*)$. Note that the convergence of the parameter $\hat{S}_i$ to true leader dynamics S cannot be guaranteed. In fact, the convergence of the $\hat{S}_i$ to true dynamics is not required. As the reinforcement learning based optimal tracking control law presented in Section 6.4 doesn't need the knowledge of S.

6.4. Optimal model-free output regulation

In this section an off-policy RL algorithm is proposed to make the agents track the leader's output. Based on the adaptive observer of Section 6.2, it is assumed that every agent has a local estimate of the leader's trajectory. In Section 6.5, this RL-based optimal control is combined with the distributed adaptive synchronizer of Section 6.2. Due to this combination the design of the output synchronizing controller doesn't require either the leader's dynamics S or the agent's dynamics (A_i, B_i, C_i), because solution to (6.4) is not explicitly needed.

Consider a linear continuous-time system with the following dynamics

$$\begin{aligned} \dot{x}_i &= A_i x_i + B_i u_i \\ y_i &= C_i x_i \end{aligned} \tag{6.26}$$

where $x_i \in \mathbb{R}^{n_i}$ is the system state, $y_i \in \mathbb{R}^q$ is the system output, $u_i \in \mathbb{R}^{m_i}$ is the control input, $A_i \in \mathbb{R}^{n_i \times n_i}$ gives the drift dynamics of the system, and $B_i \in \mathbb{R}^{n_i \times m_i}$ is the input matrix. It is assumed that the pair (A_i, B_i) is stabilizable and the pair (A_i, C_i) is observable.

Assume that the reference trajectory $\zeta_0 \in \mathbb{R}^q$ is bounded and it is generated by the command generator system given by (6.1) and (6.2).

In optimal output regulation problem, the goal is to find a control policy to make the system output y_i in (6.26) follow the reference trajectory output y_o generated by (6.1), (6.2), while minimizing a predefined performance function. Define the discounted performance function for the system (6.27) as

$$V(x_i(t), u_i(t)) = \int_t^{\infty} e^{-\gamma_i(\tau - t)}((y_i - y_0)^T Q_i (y_i - y_0) + u_i^T W_i u_i) d\tau \tag{6.27}$$

where the state weight matrix Q_i and the control input weight matrix W_i are symmetric positive definite, and $\gamma_i > 0$ is the discount factor.

Remark 6.3. As stated in Chapter 2, the discount factor $\gamma_i > 0$ in (6.27) is used to ensure that the performance function is bounded for a given control policy which assures the output regulation. This is because the steady state part of the control input does not go to zero unless the command generator dynamics is stable.

Consider a fixed state-feedback control policy linear in the system state and the command generator state as

$$u_i = K_{1i} x_i + K_{2i} \zeta_0 \tag{6.28}$$

and define an augmented state as

$$X(t) = \left[x_i(t)^T \ \zeta_0^{\ T} \right]^T \in \mathbb{R}^{n_i + p} \tag{6.29}$$

where ζ_0 is the leader state. The control input (6.28) in terms of the augmented state (6.29) becomes

$$u_i = K_{1i} x_i + K_{2i}\zeta_0 = K_i X_i \tag{6.30}$$

where $K_i = [K_{1i} \;\; K_{2i}]$. Moreover, the augmented dynamics becomes (see Chapter 2 for more details)

$$\dot{X}_i = T_i X_i + B_{1i} u_i \tag{6.31}$$

with

$$T_i = \begin{bmatrix} A_i & 0 \\ 0 & S \end{bmatrix}, B_{1i} = \begin{bmatrix} B_i \\ 0 \end{bmatrix} \tag{6.32}$$

Finally, the value function for a control policy in form of (6.30) can be written as the quadratic form

$$\begin{aligned} V(X_i(t)) &= \int_t^\infty e^{-\gamma_i(\tau-t)} X_i^T (C_{1i}{}^T Q_i C_{1i} + K_i{}^T W_i K_i) X_i \, d\tau \\ &= X_i(t)^T P_i X_i(t) \end{aligned} \tag{6.33}$$

where

$$C_{1i} = [C_i \quad -R] \tag{6.34}$$

with R as in (6.2). The optimal control input is then given by $u_i = K_i X_i$ [36] with

$$K_i = [K_{1i}, K_{2i}] = -W_i^{-1} B_{1i}{}^T P_i \tag{6.35}$$

where P_i is the solution to the discounted algebraic Riccati equation (ARE)

$$T_i^T P_i + T_i P_i - \gamma_i P_i + C_{1i}{}^T Q_i C_{1i} - P_i B_{1i} W_i^{-1} B_{1i}{}^T P_i = 0 \tag{6.36}$$

The ARE (6.36) is first solved for P_i, and then the optimal gain is obtained by substituting the ARE solution to (6.35).

An upper bound is now found for the discount factor in the performance function (6.27) to assure that the tracking error $e_{ri} = y_i - \zeta_0$ goes to zero asymptotically, when the optimal control gain (6.35) found by solving the ARE (6.36) is applied to the system. In Chapter 4, we showed that the control gain given in (6.35) makes $e^{-\gamma_i t} e_{ri}$ converge to zero asymptotically. However, the tracking error may diverge if the discount factor is not chosen appropriately. The following theorem shows that perfect output regulation achieves if (6.35) is applied to the system and the discount factor is chosen small enough.

Theorem 6.2. Let Assumptions 2 and 3 be satisfied. Let the control input (6.30) with gain given by (6.35), (6.36) be applied to the system. Then, $A_i + B_i K_{1i}$ is Hurwitz and the tracking error $e_{ri} = y_i - y_0$ goes to zero asymptotically fast if the discount factor satisfies the following condition

$$\gamma_i \le \gamma_i^* = 2\left\|(B_i W_i^{-1} B_i^T Q_i)^{1/2}\right\|. \tag{6.37}$$

Proof. We first show that $A_i + B_i K_{1i}$ is Hurwitz. To this end, define

$$P_i = \begin{bmatrix} P^i{}_{11} & P^i{}_{12} \\ P^i{}_{21} & P^i{}_{22} \end{bmatrix} \tag{6.38}$$

Then, using (6.32), for the upper left-hand side of the discounted ARE (6.36) one has

$$A_i^T P^i{}_{11} + P^i{}_{11} A_i - \gamma_i P^i{}_{11} + C_{1i}{}^T Q_i C_{1i} - P^i{}_{11} B_{1i} W_i^{-1} B_{1i}{}^T P^i{}_{11} = 0 \tag{6.39}$$

and the control gain K_{1i} becomes

$$K_{1i} = -W_i^{-1} B^T P^i{}_{11} \tag{6.40}$$

Since $Q_i > 0$ and (A_i, C_i) is observable, then $(A_i, Q_i^{1/2} C_i)$ is observable and thus there exists and unique positive definite solution $P^i{}_{11}$ to (6.39). It is shown in Chapter 4 that if condition (6.37) is satisfied, then the eigenvalues of the closed-loop system $A_i - B_i W_i^{-1} B_i^T P^i{}_{11} = A_i + B_i K_{1i}$ have negative definite parts and thus $A_i + B_i K_{1i}$ is

Hurwitz. On the other hand, it is shown in Chater 2 that there exists a positive semi-definite solution to ARE (6.36) if (A_i, B_i) is stabilizable and $S - 0.5\gamma I$ is stable. Since (A_i, B_i) is assumed stabilizable and S is assumed marginally stable, existence of a positive semi-definite solution to the ARE (6.36) is guaranteed. Multiplying the left- and right- hand sides of the ARE (6.36) by X_i^T and X_i, respectively, one has

$$2X_i^T T_i^T P_i X_i - \gamma_i X_i^T P_i X_i + X_i^T C_{1i}^T Q_i C_{1i} X_i - (P_i X_i)^T B_{1i} W_i^{-1} B_{1i}^T (P_i X_i) = 0 \qquad (6.41)$$

From this equation one can see that if $P_i X_i = 0$ then $X_i^T C_{1i}^T Q_i C_{1i} X_i = 0$. That is, the null space of P_i is a subspace of the null space of $C_{1i}^T Q_i C_{1i}$. This indicates that if $X_i^T P_i X_i = 0$ then $X_i^T C_{1i}^T Q_i C_{1i} X_i = 0$ and thus $(y_i - y_0)^T Q_i (y_i - y_0) = 0$ which yields $e_{ri} = y_i - y_0 = 0$. Therefore, the null space of P_i is in fact a subspace of the space in which the tracking error is zero. Now, consider the following Lyapunov function

$$V_i(X_i) = X_i^T P_i X_i \geq 0 \qquad (6.42)$$

To complete the proof, it remains to show that $\dot{V}_i(X_i) < 0$ if $X_i^T P_i X_i \neq 0$ and (6.37) is satisfied. This is because since $P_i \geq 0$, if $\dot{V}_i(X_i) = 0$, then $\dot{V}_i(X_i) = \dot{X}_i^T P_i X_i = 0$ and consequently $P_i X_i = 0$ which conclude the tracking error is zero. On the other hand, if $\dot{V}_i(X_i) < 0$, then, starting from any initial trajectory, it converges to the null space of P_i which is a subspace of the space of the solutions in which the tracking error is zero. To show that $\dot{V}_i(X_i) < 0$ if (6.37) is satisfied for all X_i such that $P_i X_i \neq 0$, taking the derivative of $V_i(X_i)$ gives

$$\dot{V}_i(X_i) = X_i^T (P_i A_{ci} + A_{ci}^T P_i) X_i \qquad (6.43)$$

where

$$A_{ci} = \begin{bmatrix} A_i + B_i K_{1i} & B_i K_{2i} \\ 0 & S \end{bmatrix} \tag{6.44}$$

is the closed-loop dynamics. Assume now that λ_k is an eigenvalue of A_{ci} and X_k is its corresponding eigenvector. That is,

$$A_{ci} X_k = \lambda_k X_k, \qquad k = 1, \ldots, n_i + p \tag{6.45}$$

Assuming for simplicity that A_{ci} is diagonalizable, then for any arbitrary vector X one has

$$X_i = \sum_{k=1}^{n_i+p} \alpha_k X_k \tag{6.46}$$

for some α_k. Using (6.44) and (6.45) in (6.43) yields

$$\dot{V}_i(X_i) = 2 \sum_{k=1}^{n_i+p} \alpha_k^{\,2} \operatorname{Re}(\lambda_k) X_k^{\,T} P_i X_k \tag{6.47}$$

If condition (6.37) is satisfied, then $A_i + B_i K_{1i}$ is Hurwitz and since S is assumed marginally stable, one has $\operatorname{Re}(\lambda_k) < 0 \ \forall k = 1 : n_i$ and $\operatorname{Re}(\lambda_k) = 0 \ \forall k = n_i + 1 : n_i + p$ for the eigenvalues of A_{ci} in (6.44) . Therefore, if $P_i X_i \neq 0$ and (6.37) is satisfied, then $\dot{V}_i(X_i) < 0$ and this completes the proof.

A state-feedback off-policy IRL algorithm is nowgiven to learn the solution to the discounted optimal output regulation problem. This algorithm does not require any knowledge of the system dynamics or the leader's dynamics S.

In order to obviate the requirement of the knowledge of the system dynamics, an off-policy IRL algorithm was proposed in Chapter 2 for solving the optimal regulation problem with discounted performance functions. The system dynamics (6.31) is first written as

$$\dot{X}_i = T_i X_i + B_{1i} (-K_i^{\kappa} X_i + u_i) \tag{6.48}$$

With the abuse of notation $T_i = T_i + B_{1i} K_i^{\kappa}$. Then, the Bellman equation becomes

$$
\begin{aligned}
&e^{-\gamma_i \delta t} X_i(t+\delta t)^T P_i^\kappa \, X_i(t+\delta t) - X_i(t)^T P_i^\kappa \, X_i(t) = \int_t^{t+\delta t} \frac{d}{d\tau}(e^{-\gamma_i(\tau-t)} \, X_i^{\,T} P_i^\kappa X_i) \, d\tau = \\
&\int_t^{t+\delta t} e^{-\gamma_i(\tau-t)} [X_i^{\,T} (T_i^T P_i^\kappa + P_i^\kappa T_i - \gamma_i P_i^\kappa) x + 2(u_i - K_i^{\,\kappa} X_i)^T B_{1i}^{\,T} P_i^\kappa X_i] d\tau = \\
&= -\int_t^{t+\delta t} e^{-\gamma_i(\tau-t)} X_i^{\,T} Q_i \, X_i \, d\tau + 2\int_t^{t+\delta t} e^{-\gamma_i(\tau-t)} (u_i - K_i^{\,\kappa} X_i)^T W_i \, K_i^{\,\kappa+1} X_i \, d\tau
\end{aligned}
\tag{6.49}
$$

where $Q_i = C_{1i}^{\,T} Q_i \, C_{1i} + (K_i^\kappa)^T W_i \, (K_i^\kappa)$. For a fixed control gain K_i^κ, (6.49) can be solved for both the kernel matrix P_i^κ and the improved gain $K_i^{\kappa+1}$, simultaneously. The following Algorithm 1 uses the above Bellman equation to iteratively solve the ARE equation (6.36).

Algorithm 6.1. Online Off-policy IRL State-feedback algorithm

1. *Initialization*: Start with a control policy $u_i^\kappa = K_i^0 \, X_i + e$, where K_i^κ is stabilizing and e is the probing noise.

2. Solve the following Bellman equation for P_i^κ and $K_i^{\kappa+1}$ simultaneously.

$$
\begin{aligned}
&e^{-\gamma_i \delta t} X_i(t+\delta t)^T P_i^\kappa \, X_i(t+\delta t) - X_i(t)^T P_i^\kappa \, X_i(t) = -\int_t^{t+\delta t} e^{-\gamma_i(\tau-t)} X_i^{\,T} Q_i \, X_i \, d\tau + \\
&2\int_t^{t+\delta t} e^{-\gamma_i(\tau-t)} (u_i - K_i^\kappa X_i)^T W_i \, K_i^{\,\kappa+1} \, X_i \, d\tau
\end{aligned}
\tag{6.50}
$$

3. Stop if convergence is achieved, otherwise set $\kappa = \kappa + 1$ and got to 2.

4. On convergence set $K_i = K_i^\kappa$.

In Algorithm 6.1, the control policy which is applied to the systems, i.e. u_i, can be a fixed stablizing policy. The data which is gather by applying this fixed policy to the system is then used in (6.50) to find the value function kernel matrix P_i^κ and the improved policy $u_i^{\kappa+1} = K_i^{\kappa+1} X_i$ corresponds to an updated policy $u_i = K_i X_i$.

6.5. Optimal model-free output regulation for a multi-aget heterogeneous systems

In this section the distributed observer and the optimal tracking control from previous two sections are combined, to solve Problem 6.1. The developed approach, unlike the standard method does not require the explicit solution of the output regulator equation (6.4). However, it is shown that this distributed reinforcement learning approach implicitly solves the output regulation equations without actually doing so. The optimal control law (6.30) for a single-agent system (6.26) depends on the leader's state ζ_0. But, in a distributed multi-agent network, only few agents will be aware of the leader's trajectory. Hence, the control law (6.30) cannot be used for all the agents. However, as explained in Section 6.3, by using the local adaptive synchronizer (6.8) and the corresponding update law (6.11), every agent can get a local estimation of the leader's state ζ_0 denoted by ζ_i. By using the local estimate ζ_i in (6.30), the modified optimal tracking controller for each agent is

$$u_i = K_{1i}x_i + K_{2i}\zeta_i \equiv K_iX_i \tag{6.51}$$

where K_i is obtained using the online Algorithm 6.1. Note that the tracking control (6.51) is optimal and does not depend on either the agent's system matrices (A_i, B_i, C_i) or the leader dynamics S.

The proof of the asymptotic convergence of the distributed observer and the optimal tracker are given in Sections 6.3 and 6.4, respectively. In the following theorem this results are combined to achieve output-synchronization of multi-agent heterogeneous systems.

Theorem 6.3. Consider the distributed adaptive synchronizer (6.8) and the optimal tracking controller (6.51) obtained using Algorithm 6.1 for each agent i. Then the output synchronization problem is solved for $i = 1, \cdots, N$ as $t \to \infty$, i.e., $y_i(t)$ -$y_0(t) \to 0\, \forall i$, provided that *c* in (6.8) is sufficiently large and the discount factor γ is less than the bound (6.37).

Proof: Using the control law (6.51), consider the augmented dynamics for a single agent

$$\begin{bmatrix} \dot{x}_i \\ \dot{\zeta}_i \end{bmatrix} = \begin{bmatrix} A_i + B_i K_{1i} & B_i K_{2i} \\ 0 & \hat{S}_i \end{bmatrix} \begin{bmatrix} x_i \\ \zeta_i \end{bmatrix} + \begin{bmatrix} 0 \\ e_i \end{bmatrix} \tag{6.52}$$

along with the adaptive law given by (6.11)

$$\dot{\hat{S}}_{\text{vec}i} = -\Gamma_{Si}(I_q \otimes \zeta_i) e_i \tag{6.53}$$

where e_i is defined in (6.9). Due to the block-triangular structure, the observer dynamics is independent of the agent state x_i, thus based on the separation principle the observer and the tracking control can be designed independent of each other. In Theorem 6.1 it is shown, $\zeta_i(t) - \zeta_0(t) \to 0, t \to \infty$, $\forall i = 1, \cdots, N$. For any full rank matrix *R*, $R\zeta_i(t) - R\zeta_0(t) \to 0, t \to \infty$, i.e., $R\zeta_i(t) - y_0(t) \to 0, t \to \infty$. Now based on Theorem 6.2, $y_i(t) - R\zeta_i(t) \to 0, t \to \infty$, this proves $y_i(t) - y_0(t) \to 0, t \to \infty, \forall i$.

Remark 6.4. This theorem illustrates the separation principle for output regulation of heterogeneous multi-agent systems. It also shows that the explicit solution for the output regulator equation (6.4) is not necessary since tracking is achieved by controller (6.51), which is learned online using Algorithm 6.1 for each agent.

In the previous theorem, it was shown that the explicit solution of the output regulator equation (6.4) is not required to achieve output synchronization. However, the following lemma demonstrates that the output regulator equations (6.4) are solved intrinsically for the optimal tracking control (6.51).

Lemma 6.1. Consider a network of heterogeneous multi-agent systems (6.3), and the leader (6.1). The control law (6.51) obtained using Algorithm 1 implicitly solves the output regulation equations

$$\begin{aligned} A_i\Pi_i + B_i\Gamma_i &= \Pi_i S \\ C_i\Pi_i &= R \end{aligned} \tag{6.54}$$

where $\Pi_i \in \mathbb{R}^{n_i \times p}$ and $\Gamma_i \in \mathbb{R}^{m_i \times p}$ are unique nontrivial matrices. Moreover, if the gain K_{1i} in (6.51) and (6.5) are same then $K_{2i} = \Gamma_i - K_{1i}\Pi_i$.

Proof: From Theorem 6.2, the control law (6.51) obtained using Algorithm 6.1,

$$u_i = K_{1i}x + K_{2i}\zeta_i \tag{6.55}$$

stabilizes the system (6.3), i.e., $(A_i + B_iK_{1i})$ is made Hurwitz, and guarantees output synchronization i.e., $\lim_{t\to\infty} y_i(t) - y_0(t) = 0$.

Now based on Assumption 2, there exists unique nontrivial matrix $\Pi_i \in \mathbb{R}^{n_i \times p}$ that satisfies

$$(A_i + B_iK_{1i})\Pi_i + B_iK_{2i} = \Pi_i S \tag{6.56}$$

This is a Sylvester equation and the existence of the solution Π_i is guaranteed since $\sigma(S) \bigcap \sigma(A_i + B_iK_{1i}) \in \varnothing$.

Also, based on Theorem 6.2, the output regulation for control law (6.51) achieves output synchronization, i.e.,

$$\lim_{t\to\infty} y_i(t) - y_0(t) = \lim_{t\to\infty} C_i x_i(t) - R\zeta_0(t) = 0 \tag{6.57}$$

This is based on Theorem 6.1 where $\zeta_i \to \zeta_0$ is shown. Consider the state transformation $\bar{x}_i = x_i - \Pi_i \zeta_0$. The dynamics of the new state $\bar{x}_i$ under (6.55) and (6.56) is

$$\dot{\bar{x}}_i = \dot{x}_i - \Pi_i \dot{\zeta}_0 = A_i x_i + B_i K_{1i} x_i + B_i K_{2i} \zeta_0 - \Pi_i S \zeta_0 = (A_i + B_i K_{1i}) \bar{x}_i \tag{6.58}$$

Thus $\lim_{t\to\infty} \bar{x}_i = 0$. From Theorem 6.2,

$$\lim_{t\to\infty} C_i x_i(t) - R\zeta_0(t) = \lim_{t\to\infty} C_i \bar{x}_i(t) + \lim_{t\to\infty} (C_i \Pi_i - R)\zeta_0(t) = 0 \tag{6.59}$$

since $\zeta_0(t)$ is obtained from a marginally stable system (Assumption 6.2) this implies $C_i \Pi_i - R = 0$. Using the transformation $\Gamma_i = K_{2i} + K_{1i}\Pi_i$ in (6.56) along with (6.59) gives (6.54), this completes the proof.

6.6. Simulation results

In this section, we provide a detailed simulation analysis of the proposed adaptive optimal output synchronization approach. We choose the leader to be sinusoidal trajectory generator and its dynamics is given by

$$\begin{aligned} \dot{\zeta}_0 &= \begin{pmatrix} 0 & 2 \\ -2 & 0 \end{pmatrix} \zeta_0 \\ y_0 &= \begin{bmatrix} 1 & 0 \end{bmatrix} \zeta_0 \end{aligned} \tag{6.60}$$

The heterogeneous followers are given by (6.4) for $i = 1 \cdots 4$ and their dynamics is

$$\begin{aligned} A_1 &= 0, B_1 = 10, C_1 = 1 \\ A_2 &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ 5 \end{pmatrix}, C_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \\ A_3 &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, B_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, C_3 = \begin{bmatrix} 1 & 0 \end{bmatrix} \\ A_4 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, B_4 = \begin{pmatrix} 5 \\ 0 \\ 10 \end{pmatrix}, C_4 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \end{aligned} \tag{6.61}$$

The underlying communication network of heterogeneous systems is given in Fig. 6.1

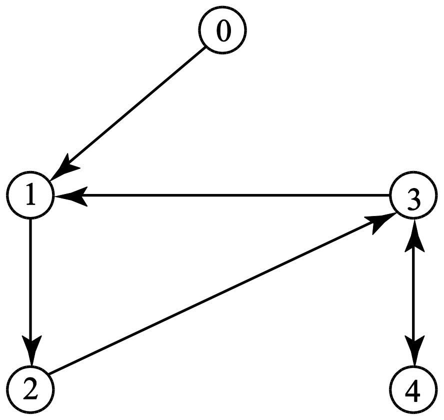

Fig. 6.1. Communication graph for the heterogeneous systems.

The distributed observer (6.8), (6.11) is implemented for $i = 1 \cdots 4$, The observer and adaptive gains are chosen as $c = 25, \Gamma_{si} = 15$. For the initial leader's state $\zeta_0(0) = (1 \quad 1)^T$, the error between observer and leader's state along with the Frobenius norm $\| S - \hat{S}_i \|_F$ is given in Fig.6.2 and Fig.6.3 and Fig.6.4, respectively. It can be seen from these figures that the introduced distributed observer converges to the leader's state.

The solution of the output regulator equation (6.4) for the given heterogeneous systems (6.61) is

$$\begin{aligned}
\Pi_1 &= \begin{matrix} 0 & 1 \end{matrix}, \Gamma_1 = \begin{matrix} 0 & 0.2 \end{matrix} \\
\Pi_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \Gamma_2 = \begin{matrix} -0.8 & 0 \end{matrix} \\
\Pi_3 &= \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \Gamma_3 = \begin{matrix} -1.5 & 0 \end{matrix} \\
\Pi_4 &= \begin{pmatrix} 0.36 & 0.48 \\ 0.64 & -0.48 \\ 0.96 & 1.28 \end{pmatrix}, \Gamma_4 = \begin{matrix} -0.192 & 0.144 \end{matrix}
\end{aligned} \tag{6.62}$$

For the following choice of the weight matrices Q,R the resulting optimal state feedback gain using LQR method for (6.61) is

$$\begin{aligned}
&Q_1 = 100, R_1 = 1, K_{11} = -10 \\
&Q_2 = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}, R_2 = 1, K_{12} = \begin{bmatrix} -10 & -10.19 \end{bmatrix} \\
&Q_3 = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}, R_3 = 1, K_{13} = \begin{bmatrix} -9.51 & -10.46 \end{bmatrix} \\
&Q_4 = 100\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, R_4 = 1, K_{14} = \begin{bmatrix} -10 & -12.66 & -6.29 \end{bmatrix}
\end{aligned} \tag{6.63}$$

Using (6.62) and (6.63) the local optimal output regulator control can be solved. Instead, the tracking control is obtained online by using the Algorithm 6.2. The convergence of the learning controller to their optimal values given by (6.62) and (6.63) for all the agents is given in Fig. 6.5. The evaluation of learned optimal tracking control along with the adaptive observer for the given multi-agent heterogeneous network is in Fig. 6.6.

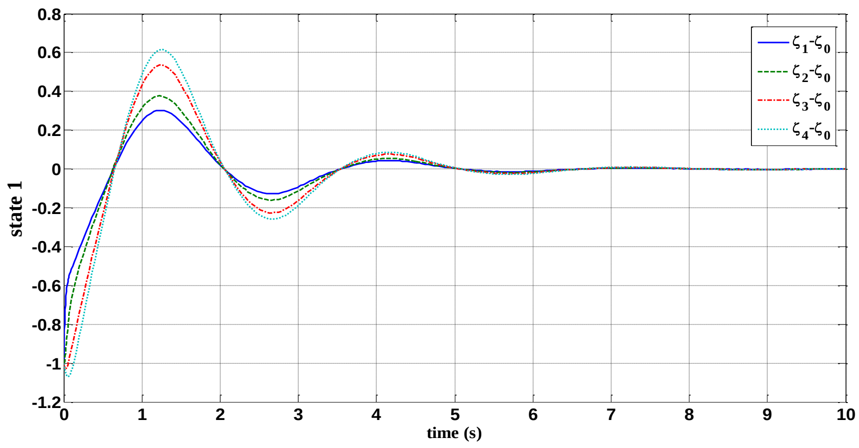

Fig. 6.2. Error between adaptive observer and leader's trajectory for state 1.

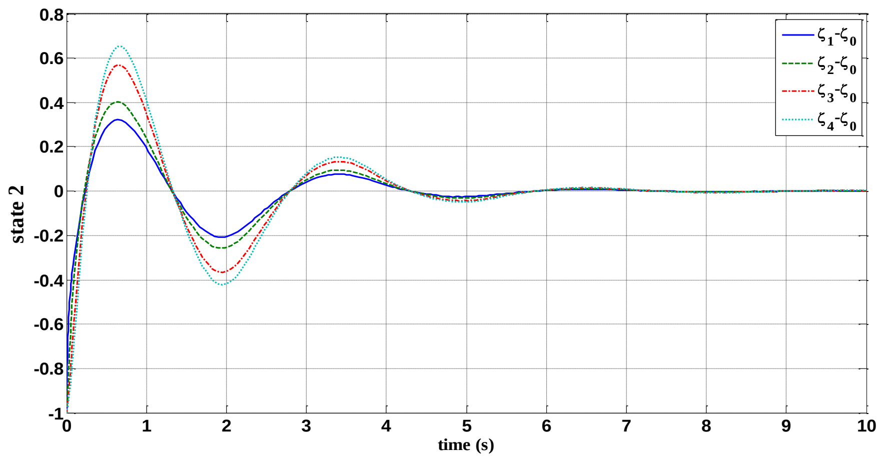

Fig. 6.3. Error between adaptive observer and leader’s trajectory for state 2.

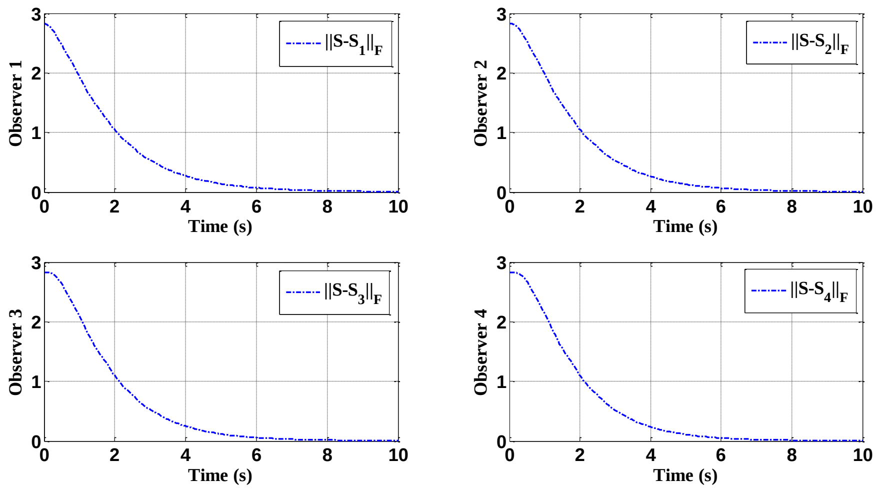

Fig. 6.4. Frobenius norm $\| S-\hat{S}_i \|_F$ for the adaptive observers.

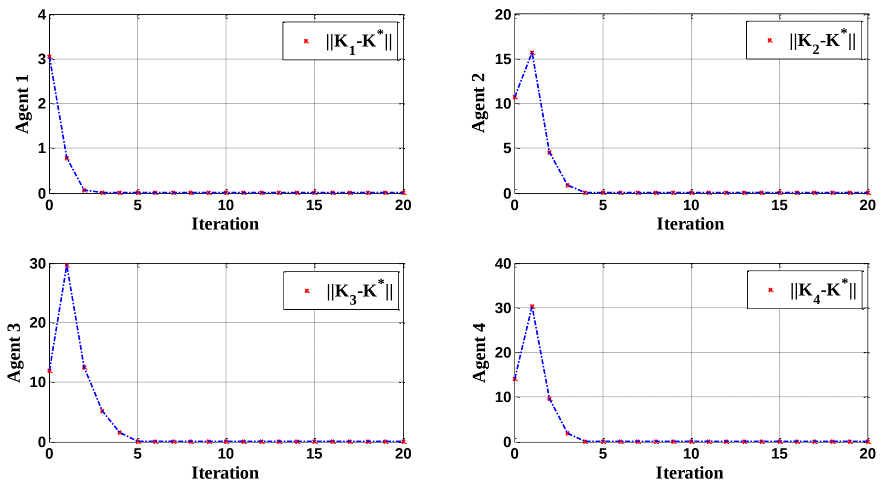

Fig. 6.5. Convergence of the learning controller to their optimal values.

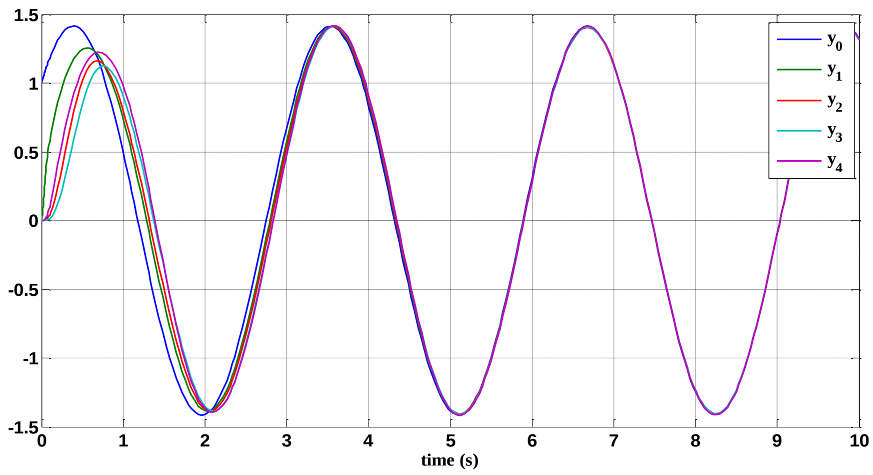

Fig. 6.6. Evaluation of the learned controller along with adaptive observer for all 4 heterogeneous agents given in (6.61).

It can be seen from these results that the introduced approach implicitly solves the output regulator equations (6.4) and solves problem 6.1 without requiring any knowledge of either agent's or leader's dynamics.

A novel approach is provided to design a model-free distributed controller to synchronize outputs of heterogeneous systems to the output of the leader. A local discounted performance function is defined for each agent which penalizes its own control effort and its tracking error. It is shown that minimizing these performance functions leads to solving AREs. It is also shown that the solutions found by solving AREs guaranteed synchronization provided that the discount factor is small enough. An adaptive distributed observer is designed to estimate the leader state and reinforcement learning is used to solve the AREs without requiring any knowledge of the dynamics of the agents. A simulation example is provided to show that the proposed approach in fact solves implicitly the output regulator equation for each agent (which is a necessary and sufficient condition to achieve output synchronization).

Chapter 7

OPTIMAL ASSISTIVE HUMAN-ROBOT INTERACTION USING REINFORCEMENT LEARNING

7.1. Introduction

Human robot interaction (HRI) is an area of increasing interest in robotic research. Its potential applications in industry, entertainment, teleoperation, household and healthcare, to name just a few, have led to increased studies to develop more flexible and efficient HRI systems. Unlike ordinary industrial robotics where the environment is structured and known, in HRI systems, the robots interact with humans who may potentially have very different skills and capabilities. Therefore, it is desired to develop human-robot systems that are capable of adapting themselves to the level of the skill of the human operator to assist the operator to accomplish a given task with minimum workload demands, and to achieve a good closed-loop behavior of the human-robot system.

Industrial robots are usually programmed to follow desired trajectories. Adaptive robot controllers using neural networks (NNs) have been widely used in the literature to provide highly effective controllers in yielding guaranteed trajectory following control for robot manipulators with unknown nonlinear dynamics, modeling inaccuracies, and disturbances. The use of NNs in feedback control systems was first proposed by Narendra [83]. Since then, NN control has been studied by many researchers. Recently, NN have entered the mainstream of control theory as a natural extension of adaptive control to systems that are nonlinear in the tunable parameters. The state of NN control is well illustrated by papers in the Automatica Special issue on NN control [84]. Overviews of the initial work in NN control are provided by [86], [87], [88], [90], [94], [18], [38], [49], [64], [65], [66], [67], [137], [32], [33] which highlighted a host of difficulties addressed for closed-loop control applications.

When the robot manipulator is in contact with an object or a human, it must be able to control not only positions, but also forces. Impedance control [37] provides an effective method for control of both position and force simultaneously in trajectory-following tasks. The objective of the impedance controller is to assign a prescribed dynamic behavior between the end effector position and end effector environment contact force. This method is inspired by how humans learn to adapt their arm impedance parameters to enable them to successfully perform contact tasks even in uncertain environments. Various impedance control methodologies have been developed in the literature to make a robot follow a desired trajectory while operating in physical contact with objects. The important feature in trajectory following is the tracking error dynamics. Therefore, impedance control generally has focused on making the tracking error dynamics behave like a prescribed robot impedance model. Adaptive impedance control techniques using NNs have been developed to tune the impedance model to be followed by the tracking error dynamics based on various considerations [19], [34], [133], [44], [35], [53].

All these mentioned adaptive NN based control methods and impedance control methods are based on tracking error dynamics, and/or making the error dynamics have a prescribed impedance characteristic. The objective of trajectory following with an error dynamics having prescribed impedance properties often restricts the applications of these approaches in HRI systems. For modern interactive HRI systems to be capable of performing a wide range of tasks successfully, it is required to include the effects of both the robot dynamics and the human dynamics. Human performance neuropsychological and human factors studies have shown that in coordinated motion with a robot, human learning has two components [12], [30], [101]. The operator learns a robot-specific inverse dynamics model to compensate for the nonlinearities of the robot [99][104], and simultaneously he learns a feedback control component that is specific to the successful performance of the task. These foundations can be incorporated in the design of the human-robot control system to include the effects of both the robot dynamics and the human dynamics, and their interactions in a task-specific outer control loop.

Recently, impedance control methods for HRI systems have been developed by some researchers, motivated by these human factor studies. One approach, namely human adaptive mechatronics, is presented in [31], [57], [58], [102]. That is, it has an inner-loop controller to make the robot behave like a virtual model and an outer-loop controller to make the HRI system stable based on the task. This method takes into account the skill differences of the operators by adjusting the impedance of the robot according to the identified operator’s model dynamics. Human modeling has been studied in the literature [105], [91]. Moreover, the impedance parameters in [31], [57], [58] are tuned based on a Lyapunov function to assure stability. But, stability is a bare minimum requirement for a controlled system, and it is also desired to tune the impedance parameters to optimize the long-term performance of the system. Sam Ge et.al [71], [115] developed an adaptive impedance method for HRI systems to find the optimal parameters of the robot impedance model.

In this chapter, a novel approach is presented to develop an intelligent HRI system with adjustable robot behavior that assists the human operator to perform a given task using the minimum effort and achieves an optimal performance for the human-in-the-loop system. The proposed method does not require the knowledge of the human model and does not need to estimate or to identify the human impedance characteristics. This makes it a biologically plausible learning algorithm. In accordance with human factors studies, the proposed method has two control loops. A robot-specific inner-loop is designed to make the robot with unknown dynamics behave like a simple prescribed robot impedance model as perceived by a human operator. This means the human does not need to learn an inverse dynamics model to compensate for robot nonlinearities. In contrast to most previous work, this is not a trajectory following objective and no information of the task performance is used in the inner loop. Next, a task-specific outer-loop is developed, taking into account the human transfer characteristics, to find the optimal parameters of the prescribed robot impedance model depending on the task. In the outer loop, the problem of finding the optimal parameters of the prescribed robot impedance model is formulated as a LQR problem such that both the tracking errors and the human operator effort are minimized. RL is used to solve the given LQR problem to obviate the requirement of the knowledge of the human model.

The contributions of this chapter are as follows.

1. An inner-loop controller is designed to make the nonlinear unknown robot dynamics behave like a prescribed robot impedance model. This is more general than standard trajectory following. The proposed inner-loop controller does not require either task information or the specific prescribed robot impedance model parameters. This enables us to decouple the design of the robot-specific inner loop from the design of the task-specific outer-loop controller.
2. The problem of designing the optimal parameters of the prescribed robot impedance model is transformed into an LQR problem in a task-specific outer-loop control design. These parameters are determined by minimizing a performance function in terms of the human control effort and the tracking error.

3. A reinforcement learning technique is employed to solve the task-specific LQR problem online in real time and without knowing the knowledge of the human model.
4. The proposed approach does not restrict the robot to a trajectory following task, because it leaves the task-specific details to the design of the outer-loop which incorporate the human operator.

The rest of the chapter is organized as follows. The next section presents the overall structure of the proposed control design method for the HRI systems. Both the inner-loop control design and the outer-loop control design are briefly discussed. Sections 8.3 and 8.4 discuss the inner-loop design and the outer-loop design, respectively, in details. Finally, Sections 8.5 and 8.6 present the simulation results and conclusion, respectively.

7.2. HRI control structure overview

In this section, the structure of the HRI control system developed in this chapter is overviewed. The proposed control structure is motivated by the human factors, which states that the human learns a robot-specific inverse dynamics model to compensate for the nonlinearities of the robot, and simultaneously a feedback control component that is specific to the successful performance of the task. Therefore, the HRI design here has two objectives. First, a robot torque controller is provided to avoid the need for the operator to learn a robot-specific model. Second, assistive inputs are provided to augment the operator’s control effort so that the operator performs a given task with minimum workload demands and maximum performance.

To achieve these goals, the proposed method has two control loops. The first loop is a robot-specific inner loop which does not require any information of the task. See

Fig. 7.1. The second loop is a task-specific outer loop which includes the human operator dynamics, the robot and the task performance details. See Fig. 7.2.

The robot-specific inner-loop controller is shown in Fig. 7.1. The objective is to make the unknown robot manipulator dynamics behave like a prescribed robot impedance model as perceived by a human operator. This avoids the need for the operator to learn a model of the specific robot system. The human only needs to interact with the simplified impedance model. To compensate for the unknown robot nonlinearities, an adaptive NN controller is employed. This is not the same as the bulk of the work in robot impedance control and NN control, which is directed towards making a robot follow a prescribed trajectory, and/or causing the trajectory error dynamics to follow a prescribed impedance model. No trajectory information is needed for the inner-loop design. This leaves the freedom to incorporate task information in an outer-loop design. The robot-specific inner-loop controller shown in Fig. 7.1 called model reference neuro-adaptive control. This is because an adaptive NN controller is developed to make the robot dynamics, from the human force to the robot motion, behave like the prescribed reference impedance model.

The task-specific outer-loop controller is shown in Fig. 7.2. Given the robot-specific inner-loop design, the optimal parameters of the prescribed robot impedance model are determined in this task-specific outer loop to guarantee motion tracking and also assist the human to perform the task with minimum effort. To this end, the problem of determining the optimal parameters of the prescribed robot impedance model is transformed into a LQR problem. This design must take into account the unknown human dynamics as well as the desired overall performance of the human-robot system, which depends on the task. Reinforcement learning is used to solve the LQR problem without any knowledge of the human operator dynamics.

Fig. 7.3 shows the overall schematic of the proposed two-loop control design method for the HRI system. The details of the inner-loop design and the outer-loop design are given in Sections 7.3 and 7.4, respectively.

7.3. Robot-specific inner-loop: A model reference neuro-adaptive controller

In this section, the design of the inner-loop controller in Fig. 7.1 is given. The aim of the inner-loop controller is to make the robot manipulator behave like a prescribed robot impedance model. The proposed inner-loop control method has two main differences from the existing adaptive impedance control methods. First, in contrast to trajectory following based methods, the proposed method is a robot-specific method that does not require a reference motion trajectory. That is, the proposed method minimizes the model-following error between the output of the prescribed robot impedance model and the motion of the robot without needing task information. Second, in the proposed method, the designed control torque does not require any knowledge of the prescribed robot impedance model. This enables us to decouple the design of the robot-specific inner-loop controller from the design of the task-specific outer-loop control design.

Consider the dynamical model of robot manipulator in Cartesian space [64]

$$M(q)\,\ddot{x} + C(q,\dot{q})\,\dot{x} + F_c(\dot{q}) + G(q) + \tau_d = \tau + K_h f_h \qquad (7.1)$$

with $M = J^{-T} M^* J^{-1}$, $C = J^{-T}(C^* - M^* J^{-1}\dot{J})J^{-1}$, $M = J^{-T} M^* J^{-1}$, $F_c = J^{-T} F^*$, $G = J^{-T} G^*$, $\tau = J^{-T}\tau^*$, where $q \in \mathbb{R}^n$ is the vector of generalized joint coordinates, n is the number of joints, $x \in \mathbb{R}^n$ is the end-effector Cartesian position, the control input force is $\tau = J^{-T}\tau^*$ with τ^* is the vector of generalized torques acting at the joints, $M^* \in R^{n\times n}$ is the symmetric positive definite mass (inertia) matrix, $C^*(q,\dot{q})\,\dot{q} \in \mathbb{R}^{n\times 1}$ is the vector of Coriolis and centripetal forces, $F_c^*(\dot{q}) \in \mathbb{R}^{n\times 1}$ is the Coulomb friction term, $G^*(q) \in \mathbb{R}^{n\times 1}$ is the vector of gravitational torques, τ_d is a general nonlinear disturbance , f_h is the human control effort, K_h is a gain and J is the Jacobian matrix.

Remark 7.1. Note that in the above dynamics, it is assumed that the human force is sensed by a force sensor and is amplified by a gain K_h before applying to the robot. For example, for the x-y table example in [101], the force generated by the hand to the grip is measured by a force sensor and is magnified before it is applied to the stage. As shown in Section 7.5, this gain is employed to help the human to minimize the tracking error and maximize the performance. It is shown in Remark 7.5 that if the amplification of the human force is not possible for a specific application, the proposed method can still be used. It is assumed that the robot manipulator dynamics are unknown.

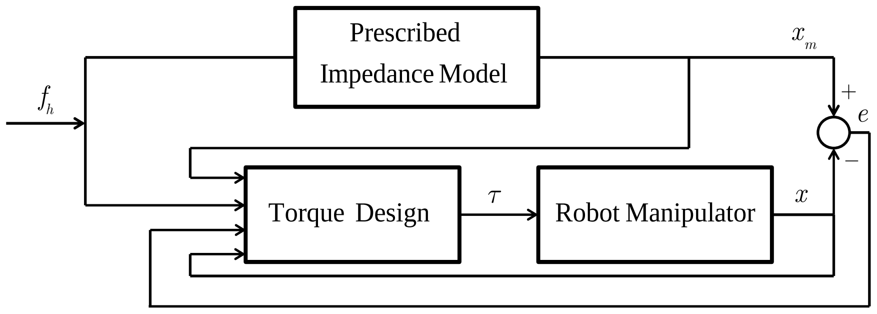

Fig. 7.1. The robot-specific inner-loop model-following control design

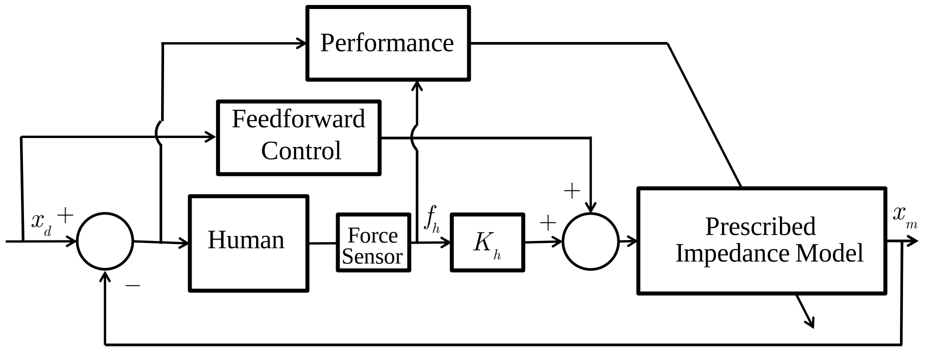

Fig. 7.2. The task-specific outer-loop control design

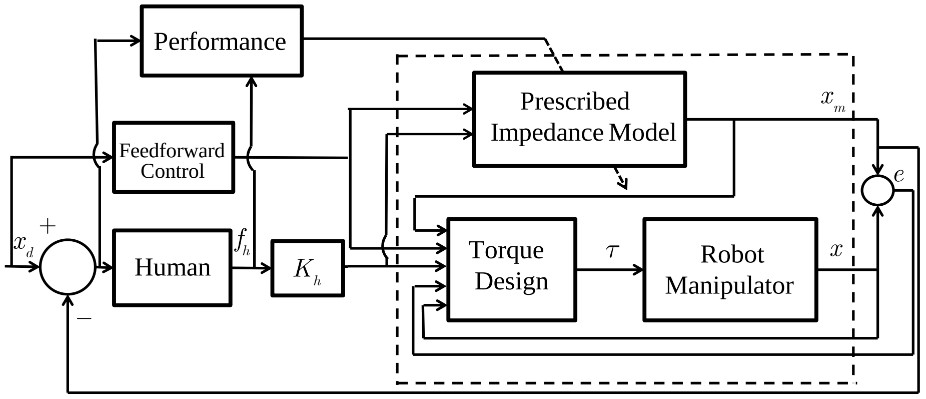

Fig. 7.3. The overall two-loop control design method for the adaptive HRI system

Consider the prescribed robot impedance model

$$\bar{M}\ddot{x}_m + \bar{B}\dot{x}_m + \bar{K}x_m = K_h f_h + \bar{l}(x_d) \equiv l(f_h, x_d) \tag{7.2}$$

in Cartesian space, where x_m is the output of the prescribed robot impedance model, $\bar{M}$, $\bar{B}$, and $\bar{K}$ are the desired inertia, damping, and stiffness parameter matrices, respectively. These parameters are specified in the task-specific outer control loop design in Section 7.4. The auxiliary input $\bar{l}(x_d)$ is a trajectory dependent input and is also designed in Section 7.4. No trajectory information is needed in this section.

Design Objective: The aim is to design the force τ in (7.1) to make the unknown robot dynamics (7.1) from the human force f_h to the Cartesian coordinates x behave like the prescribed robot impedance model (7.2). That is, it is desired to make the following model-following error go to zero.

$$e = x_m - x \tag{7.3}$$

Note this is not a trajectory-following error. Therefore, this is a model-following design, and not a trajectory-following design, in contrast to most work on robot torque control. No task information is required in this section. All task-specific details are taken into account in the next section.

Write the control torque as

$$\tau = \tau_1 + \overline{l}\,(x_d) \tag{7.4}$$

where the feedforward term $\overline{l}\,(x_d)$ is designed in the next section. Then, the robot dynamics (7.1) becomes

$$M(q)\,\ddot{x} + C(q,\dot{q})\,\dot{x} + F_c(\dot{q}) + G(\dot{q}) + \tau_d = \tau_1 + l(f_h, x_d) \tag{7.5}$$

It is now required to design a control torque τ_1 to make the robot behave like the prescribed robot impedance model (7.2). Consider the control torque

$$\tau_1 = \hat{W}^T\phi(\hat{V}^T z) + K_v r - v(\mathrm{t}) - l(f_h, x_d) \tag{7.6}$$

where $v(\mathrm{t})$ is a robustifying signal to be specified , K_v is the control gain, and

$$r = \dot{e} + \Lambda_1 e + \Lambda_2 \varepsilon \tag{7.7}$$

is the sliding mode error with

$$\varepsilon = \int_0^t e(\tau)\,d\tau \tag{7.8}$$

Finally,

$$\hat{h}(z) = \hat{W}^T\phi(\hat{V}^T z) \tag{7.9}$$

is a neural network (NN) with $z = [q, \dot{q}, \dot{x}_m, \ddot{x}_m, e, \dot{e}, \varepsilon]^T$ the input to the NN, $\hat{W}$ and $\hat{V}$ the NN weights and $\phi(z)$ the vector of activation functions. As is shown in the proof of Theorem 7.1, the NN controller in (7.6) is used to compensate for the unknown robot function h defined as

$$h(z) = M(q)(\ddot{x}_m + \Lambda_1\dot{e} + \Lambda_2 e) + C(q,\dot{q})(\dot{x}_m + \Lambda_1 e + \Lambda_2\varepsilon) + F_c(\dot{q}) + G(q) \tag{7.10}$$

The neural network universal approximation property specifies that any unknown continuous function can be approximated on a compact set using a two-layer NN to any arbitrary precision. That is, for the continuous function $h(z)$ on a compact set $z \in \Omega$, one has

$$h(z) = W^T \phi(V^T z) + \varepsilon(z) \tag{7.11}$$

where V is a matrix of first-layer weights, W is a matrix of second-layer weights, and ε is the NN functional approximation error. The ideal weight vectors W and V are unknown and is approximated online. Therefore, $h(z)$ is approximated as (7.9) with $\hat{W}$ and $\hat{V}$ the estimations of W and V, respectively. Define

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \tag{7.12}$$

and $\hat{Z}$ equivalently**.**

Assumption 7.1. The ideal NN weights are bounded by a constant scalar so that

$$\left\| Z \right\| \le Z_B \tag{7.13}$$

The following theorem shows that the proposed control input (7.4) with τ_1 given by (7.6) guarantees the boundness of the model-following error e and the NN weights.

Theorem 7.1. Consider the robot manipulator dynamics (7.1) and the prescribed robot impedance model (7.2). Let the control input be chosen as (7.4) and (7.6). Let Assumption 7.1 hold. Let the update rule for the NN weights be given by

$$\dot{\hat{W}} = F \hat{\phi} r^T - F \hat{\phi}' \hat{V}^T z r^T - k F \| r \| \hat{W} \tag{7.14}$$

$$\dot{\hat{V}} = G z (\hat{\phi}' \hat{W} r)^T r^T - k G \| r \| \hat{V} \tag{7.15}$$

where $\hat{\phi} = \phi(\hat{V}^T z)$, $\hat{\phi}' = d\phi(y) / dy \,|_{y=\hat{V}^T z}$, $F = F^T > 0$, $G = G^T > 0$ and $k > 0$ is a small design parameter. Let the robustifying term be

$$v(t) = -K_z(\|\hat{Z}\| + Z_B) \tag{7.16}$$

where $K_z > 0$. Then, $e(t)$ in (7.3) and the NN estimated weights are uniformly ultimately bounded (UUB).

Proof. By differentiating (7.3) with respect to time one has $\dot{e} = \dot{x}_m - \dot{x}$, or equivalently $\dot{x} = \dot{x}_m - \dot{e}$. Differentiating $\dot{x}$ gives $\ddot{x} = \ddot{x}_m - \ddot{e}$. Considering the sliding mode tracking error r defined in (7.7), one has $\dot{e} = r - \Lambda_1 e - \Lambda_2 \varepsilon$. Differentiating $\dot{e}$ gives $\ddot{e} = \dot{r} - \Lambda_1 \dot{e} - \Lambda_2 e$. Using these expressions in (7.1) yields

$$\begin{aligned} &M(q)(\ddot{x}_m - (\dot{r} - \Lambda_1 \dot{e} - \Lambda_2 e)) + C(q,\dot{q})(\dot{x}_m - (r - \Lambda_1 e - \Lambda_2 \varepsilon)) + \\ &F_c(\dot{q}) + G(q) + \tau_d = \tau + K_h f_h \end{aligned} \tag{7.17}$$

This gives the sliding mode error dynamics

$$M(q)\dot{r} = -C(q,\dot{q})r + h(q,\dot{q},\dot{x}_m,\ddot{x}_m,e,\dot{e},\varepsilon) + \tau_d - \tau - K_h f_h \tag{7.18}$$

with h defined in (7.10). The robot manipulator dynamics (7.1) is assumed to be unknown and therefore h in (7.18) is unknown and approximated online by (7.9). Then, the closed-loop filtered error dynamics (7.18) becomes

$$M(q)\dot{r} = -C(q,\dot{q})r + \hat{W}^T \phi(\hat{V}^T z) + \tau_d - \tau - K_h f_h + \tilde{h} \tag{7.19}$$

where $\tilde{h} = h - \hat{h}$ is the estimation error for approximating the function h. Substituting τ form (7.4) and (7.6) in (7.19) gives

$$M(q)\dot{r} = -C(q,\dot{q})r - K_v r + \tau_d + \tilde{h} + v(t) \tag{7.20}$$

The remainder of the proof is the same as [66] and thus is only outlined here. A Lyapunov function is defined as

$$L = \frac{1}{2} r^T M(q) r + tr(\tilde{W}^T F^{-1} \tilde{W}) + tr(\tilde{V}^T F^{-1} \tilde{V}) \tag{7.21}$$

where the weight estimation errors are $\tilde{W} = W - \hat{W}$, $\tilde{V} = V - \hat{V}$, and it is shown using (7.14)-(7.16) and (7.19) that the Lyapunov function derivative is negative outside a compact set. This guarantees the boundedness of the filtered tracking error r as well as the NN weights. Specific bounds on r and the NN weights are given in [66]. □

Note that the proposed controller (7.4), (7.6) is composed of four parts. The first part is a nonlinear compensator consisting of a NN controller to compensate for unknown function h defined in (7.10). The second part of the controller is a stabilizing PID controller that stabilizes the model-following error e. The third part is a robust term that is designed to achieve robustness against uncertainties. Finally, the last part is used to compensate for the input $l(f_h, x_d)$ of the prescribed robot impedance model.

Fig. 7.4 shows the detailed schematic of the proposed inner-loop controller. We call this model reference neuro-adaptive control because the neural network adaptive controller causes the robot dynamics to behave like the prescribed impedance model (7.2). This is in contrast to NN torque control work, which seek to make the robot motion $x(t)$ follow a prescribed trajectory.

Remark 7.2. Function $h(z)$ in (7.10) does not contain the parameters $\bar{M}$, $\bar{D}$ and $\bar{K}$ of the prescribed robot impedance model (7.2). This means that the NN does not need to estimate the impedance model. This is in contrast to methods that design the torque τ in (7.1) to guarantee following a desired trajectory, and demand that the trajectory tracking error dynamics follow a prescribed impedance model. Our design of the robot-specific inner-loop controller is independent from any task objectives. All task-specific information is considered in the outer-loop design in Section 7.4.

7.4. task-specific outer-loop adaptive impedance control

In this section the design of the outer task loop controller shown in Fig. 7.2 is detailed. It was shown in Section 7.3 that the robot-specific controller of Theorem 7.1 makes the nonlinear unknown robot behave like the simple prescribed robot impedance model (7.2) as perceived by the human operator. In this section, the parameters of the prescribed robot impedance model given in (7.2) are optimized to assist the human to perform a given task with minimum effort and to minimize a tracking error. To this end, the problem of optimizing the parameters of the prescribed robot impedance model is transformed into an LQR problem and then reinforcement learning is used to solve the given problem without requiring the human dynamics model.

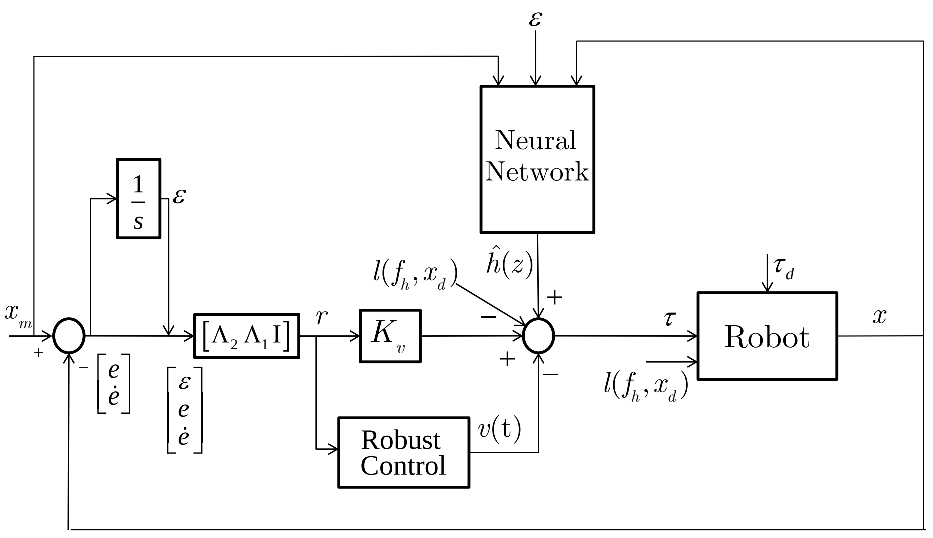

Fig 7.4. Model reference neuro-adaptive inner-loop controller

Design Objective: The aim of the task-specific outer-loop controller is to find the optimal values of the prescribed impedance parameters $\bar{B}$, $\bar{K}$ and the human gain K_h and the auxiliary input $\bar{l}(x_d)$ in (7.2) to minimize the human control effort f_h and optimize the tracking performance depending on the task.

Note that there are many types of tasks. We focus on tasks that require the human-robot system to follow a desired trajectory. This includes point-to-point motion control [22].

The dynamics of the human model and the interaction of the robot and the human are considered in this outer-loop control design. The human dynamics change during the task learning process. After learning, an expert human operator is characterized by a simple linear transfer characteristic. Therefore, the human impedance model is assumed to be

$$(K_d s + K_p) f_h = e_d \tag{7.22}$$

where K_d and K_p are unknown matrix gains. These parameters vary from one individual to another and depend on the specific task.

7.4.1 Task-specific outer-loop control method: An LQR approach

The block diagram of the outer loop task controller is sketched in Fig. 7.2 and shown in detail in Fig. 7.5. As shown in this figure, in addition to the adaptive impedance loop that specifies the optimal impedance parameters, an assistive feedforward input and a human force gain are employed to help the human to minimize the tracking error. The feedforward term $\bar{l}(x_d)$ in (7.2) is designed to make the steady-state tracking error go to zero. The human gain K_h and the optimal values of the prescribed impedance parameters $\bar{K}$ and $\bar{B}$ in (7.2) are determined to minimize the human effort and the tracking error for a given task.

In the following it is shown how the problem of finding optimal values of $\bar{B}$, $\bar{K}$ and K_h is transformed into an LQR problem, and how these parameters are obtained by solving an ARE.

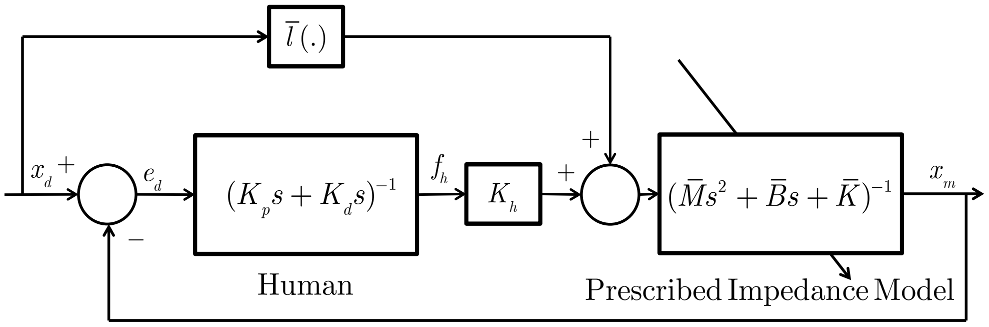

Fig. 7.5. Human-robot interface in the task-specific outer loop

Define the tracking error

$$e_d = x_m - x_d \in \mathbb{R}^n \tag{7.23}$$

and

$$\bar{e}_d = [e_d^T \ \dot{e}_d^T]^T = \bar{x}_m - \bar{x}_d \in \mathbb{R}^{2n} \tag{7.24}$$

with

$$\bar{x} = [x_m^T \ \dot{x}_m^T]^T \in \mathbb{R}^{2n} \tag{7.25}$$

and

$$\bar{x}_d = [x_d^T \ \dot{x}_d^T]^T \in \mathbb{R}^{2n} \tag{7.26}$$

Based on this tracking error, define the performance index

$$J = \int_t^{\infty} (\bar{e}_d^{\,T} Q_d \, \bar{e}_d + f_h^{\,T} Q_h \, f_h + u_e^{\,T} R \, u_e) \; d\tau \tag{7.27}$$

where $Q_d = Q_d^T > 0$, $Q_h = Q_h^T > 0$, $R = R^T > 0$, and u_e is the feedback control input which depends linearly on the tracking error $\bar{e}_d$ and the human effort f_h. Then,

$$u_e = K_1\,\bar{e}_d + K_2\,f_h \tag{7.28}$$

It is shown in Theorem 7.2 that the control input (7.28) has two components. The first component, i.e. K_1, tunes the prescribed impedance parameters $\bar{B}$, $\bar{K}$ and the second component, i.e. K_2, tunes the human control gain K_h.

Remark 7.3. Note that by minimizing the performance index (7.27), both the tracking error $\bar{e}_d$ and the human effort f_h are minimized.

By defining the augmented state

$$X = \begin{bmatrix} \bar{e}_d \\ f_h \end{bmatrix} \in \mathbb{R}^{3n} \tag{7.29}$$

the performance index (7.27) can be written as

$$J = \int_t^\infty (X^T Q\, X + u_e^{\ T} R\, u_e)\; d\tau \tag{7.30}$$

where $Q = \operatorname{diag}(Q_d, Q_h)$, and $u_e = K\,X$ with $K = [K_1 \quad K_2]$.

The dynamics of the system with augmented state (7.29) are now given. Using (7.2) one has

$$\bar{x} = \begin{bmatrix} 0 & I_{n\times n} \\ 0 & 0 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ I_{n\times n} \end{bmatrix} u \equiv A_q \bar{x} + B_q u \tag{7.31}$$

where $\bar{x}$ is defined in (7.25), and

$$u = \bar{M}^{-1}(-K_q \bar{x} + K_h f_h) + \bar{M}^{-1}\,\bar{l}\,(x_d) \tag{7.32}$$

with

$$K_q = [\bar{K} \quad \bar{B}] \tag{7.33}$$

On the other hand, based on the human model we have

$$(K_d s + K_p) f = h\, e_d \tag{7.34}$$

which can be written in time domain as

$$K_d \dot{f}_h + K_p f_h = h\, e_d \tag{7.35}$$

or equivalently

$$\dot{f}_h = -K_d^{-1} K_p\, f_h + h\, \bar{K}_d \bar{e}_d \tag{7.36}$$

where $K_{d,0} = [K_d^{-1} \;\; 0]$ and $\bar{e}_d$ is defined in (7.24).

The following theorem shows how the problem of finding the optimal parameters of the prescribed impedance model and the human gain are obtained by solving an LQR problem.

Theorem 7.2. Consider the prescribed robot impedance model (7.2). Based on dynamics in (7.31) and (7.36), define augmented matrices A and B by

$$A = \begin{bmatrix} A_q & 0 \\ K_{d,0} & -hK_d^{-1}K_p \end{bmatrix},\; B = \begin{bmatrix} B_q \\ 0 \end{bmatrix} \tag{7.37}$$

Define

$$K = [K_q \;\; K_h] \tag{7.38}$$

as the matrix of the impedance parameters and the human gain. Then, the optimal value of K which minimizes the performance index (7.27) is given by

$$K = -\bar{M}\, R^{-1} B^T P \tag{7.39}$$

where P is the solution to the ARE

$$0 = A^T P + P\, A + P\, B R^{-1} B^T P + Q \tag{7.40}$$

Then, the optimal feedback control is given by

$$u_e = \bar{M}^{-1}\bar{K}\,e_d + \bar{M}^{-1}\bar{B}\,\dot{e}_d + \bar{M}^{-1}K_h\,f_h \tag{7.41}$$

Proof. Manipulating (7.32) gives

$$u = \bar{M}^{-1}(K_q\bar{e}_d + K_h f_h) + M^{-1}\left(\bar{l}(x_d) - K_q\bar{x}_d\right) \equiv u_e + u_d \tag{7.42}$$

where $\bar{e}_d$ and $\bar{x}_d$ are defined in (7.24) and (7.26), and

$$u_e = \bar{M}^{-1}(K_q\bar{e}_d + K_h f_h) \tag{7.43}$$

is a feedback control input, and

$$u_d = M^{-1}\left(\bar{l}(x_d) - K_q\bar{x}_d\right) \tag{7.44}$$

is a feedforward control input. The steady state or feedforward term is used to guarantee perfect tracking. That is, in the steady state one has

$$\dot{\bar{x}}_d = A_q\bar{x}_d + B_q u_d \tag{7.45}$$

where $\bar{x}_d$ is defined in (7.26). Therefore,

$$\bar{l}(q_d) = \bar{M}\,u_d + K_q\bar{x}_d = \bar{M}\,B_q^{-1}(\dot{\bar{x}}_d - A_q\bar{x}_q) + K_q\bar{x}_d \tag{7.46}$$

Taking derivative of $\bar{e}_d$ and using (7.31) and (7.45), and some manipulations gives

$$\dot{\bar{e}}_d = A_q\bar{e}_d + B_q u_e \tag{7.47}$$

Using the augmented state (7.29), and using (7.47) and (7.36) one has

$$\dot{X} = \begin{bmatrix} \dot{\bar{e}}_d \\ \dot{f}_h \end{bmatrix} = \begin{bmatrix} A_q & 0 \\ K_{d,0} & -hK_d^{-1}K_p \end{bmatrix}\begin{bmatrix} \bar{e}_d \\ f_h \end{bmatrix} + \begin{bmatrix} \bar{B}_q \\ 0 \end{bmatrix} u_e \equiv AX + Bu_e \tag{7.48}$$

The control input u_e in terms of the augmented state can be written as

$$u_e = \bar{M}^{-1}\left(K_q\bar{e}_d + K_h f_h\right) = \bar{M}^{-1}K\,X \tag{7.49}$$

Finding the optimal feedback control (7.49) to minimize the performance index (7.27) subject to the augmented system (7.48) is an LQR problem and its solution is given by

$$u_e^* = -R^{-1}B^T PX \tag{7.50}$$

where P is the solution to the Riccati equation (7.40). Equating the right-hand sides of (7.49) and (7.50) yields

$$K = [K_q \;\; K_h] = -\bar{M}R^{-1}B^T P \tag{7.51}$$

This completes the proof. □

Remark 7.4. The K vector defined in (7.38) includes both the parameters (7.33) of the robot impedance model and the gain K_h of the human force. Therefore, the solution to the formulated LQR problem gives the optimal values of the prescribed impedance model parameters and the gain of the human operator force.

Remark 7.5. The outer-loop control design consists of two components: An adaptive impedance component which finds the optimal values of the parameters (7.33) of the prescribed impedance model, and an assistive component including the human force gain K_h and the feedformard term $\bar{l}(q_d)$ to help the human to minimize the tracking error.

Remark 7.6. The K vector defined in the above equation includes both the parameters of the robot impedance model and the gain K_h of the human force. Therefore, the solution to the formulated LQR problem gives the optimal values of the prescribed impedance model parameters and the gain of the human operator force. If the human gain cannot be magnified for a specific HRI application, i.e. if $K_h = 1$, then, one can set the coefficient of f_h in the control input as $\bar{M}^{-1}$ and then find $\bar{M}$ instead of K_h. That is, if $K_h = 1$ and $\bar{M}$ is unknown, then one has $K = [\bar{M}^{-1}K_q \;\; \bar{M}^{-1}] = -R^{-1}B^T P$, which gives unknown parameters of the impedance model.

7.4.2. Learning optimal parameters of the prescribed impedance model

Solving (7.40) requires the knowledge of the matrix A in (7.37) and consequently the knowledge of the human model. In order to obviate the requirement of knowledge of the human model, the IRL algorithm is used to solve the given LQR problem. To obtain the IRL Bellman equation for the given LQR problem, note that for time interval $\Delta t > 0$, the value function (7.27) satisfies

$$V(X(t)) = \int_t^{t+T} (X^T Q X + u_e{}^T R u_e)\; d\tau + V(X(t+T)) \tag{7.52}$$

It is well known that the value function for the LQR problem is quadratic in terms of the state of the system state. That is, $V(X) = X^T P\, X$. Using this quadratic form in (7.52) yields the IRL Bellman equation

$$X(t)^T P\; X(t) = \int_t^{t+\Delta t} \left[X(t)^T Q\;\; X(t) + u_e{}^T R\;\; u_e \right] d\tau + X(t+\Delta t)^T P\, X(t+\Delta t) \tag{7.53}$$

Using (7.53) for policy evaluation step and an update law in form of (7.50) to find an improved policy, the following IRL-based algorithm is obtained for solving (7.40).

Algorithm 7.1. Online IRL algorithm for the outer-loop control design

Initialization: Start with an admissible control input $u^0 = K_1{}^0 X$

Policy evaluation: Given a control policy u^i, find P^i using the Bellman equation

$$X(t)^T P^i X(t) = \frac{1}{2}\int_t^{t+\Delta t} \left[X(t)^T Q\;\; X(t) + (u_e{}^i)^T R\; (u_e{}^i) \right] d\tau + X(t+\Delta t)^T P^i\; X(t+\Delta t) \tag{7.54}$$

Policy improvement: update the control input using

$$u_e^{i+1} = -R^{-1} B_1^T P^i X \tag{7.55}$$

The policy evaluation and improvement steps (7.54) and (7.55) are repeated until the policy improvement step no longer changes the present policy, thus convergence to the optimal controller is achieved. That is, until $\| P^{i+1} - P^i \| \leq \varepsilon$ is satisfied, where ε is a small constant. The solution for P^i in the policy evaluation step (7.55) is generally carried out in a least squares (LS) sense. In fact (7.55) is a scalar equation and *P* is a symmetric *n*×*n* matrix with *n*(*n* + 1)/2 independent elements and therefore at least *n*(*n* + 1)/2 data sets are required before (7.55) can be solved using LS.

Note that Algorithm 7.1 solves ARE (7.40) and does not require knowledge of the A matrix which contains knowledge of the human dynamics. In fact, the information of A is embedded in the online measurement of system data.

7.5. Simulation results

In this section, the proposed reinforcement learning based optimized assistive control is applied to a two-link planar robot arm. The length of the links are $L_1 = 1m$ and $L_2 = 1m$. The masses of the rigid links are $m_1 = o.8kg$ and $m_2 = 2.3kg$. The gravitational acceleration is $g = 9.8\ m / s^2$.

In the following, it is shown in the first part how the proposed inner-loop control design method make the robot behave like a prescribed impedance model regardless of its impedance parameters. Then, in the second part, it is shown how to find optimal parameters of the prescribed impedance parameters to make the tracking error in an optimal manner.

7.5.2. Inner Loop

In this subsection it is shown how a robot two-link planar robot arm behaves like a given impedance model using the inner-loop control design method presented in Section 7.3. The human force is assumed sinusoidal in both directions in this section. The simulations are performed for two different sets of impedance gains to show that the robot behaves like the impedance model regardless of the impedance model parameters.

Case 1) The first matrix gains for the impedance model are chosen as

$$\bar{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \bar{B} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix},\ \bar{K} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Fig. 7.7 show that the trajectories of the prescribed impedance model are very close to the trajectories of the robot arm.

Case 2) The second matrix gains for the impedance model are chosen as

$$\bar{M} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},\ \bar{B} = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix},\ \bar{K} = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}.$$

Figs. 7.8-7.10 show that the trajectories of the prescribed impedance model are the same as the trajectories of the robot arm. Figs 7.5-7-10 confirm that the robot behaves like the impedance model regardless of the impedance parameters. This enables us to tune the impedance parameters in the outer loop to minimize the tracking error and the human workload.

7.5.2. Outer Loop

In this subsection, the results of the proposed outer-loop controller method are presented. The mass matrix for the prescribed impedance model is set to identity of appropriate dimension and it is assumed that desired trajectory to be followed by the robot is $x_d = [\sin(t), \cos(t)]$. It is assumed also that the human admittance parameters are $K_d = 10$ and $K_p = 20$ and $h = 1$. The matrices *A* and *B* then become

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & -2 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7.56}$$

Then, the offline solution toof the ARE is given as

$$P^* = \begin{bmatrix} 507.364 & 0.000 & 7.076 & -0.000 & 3.378 & -0.000 \\ 0.000 & 507.364 & 0.000 & 7.076 & -0.000 & 3.378 \\ 7.075 & 0.000 & 7.170 & -0.000 & 0.046 & 0.000 \\ -0.000 & 7.076 & -0.000 & 7.170 & -0.000 & 0.046 \\ 3.378 & -0.000 & 0.046 & -0.000 & 99.995 & -0.000 \\ -0.000 & 3.378 & 0.000 & 0.046 & -0.000 & 99.995 \end{bmatrix}$$

and

$$K^* = \begin{bmatrix} -70.758 & 0.000 & -71.704 & 0.000 & 0.458 & 0.000 \\ 0.000 & -70.758 & 0.000 & -71.704 & -0.000 & 0.458 \end{bmatrix} \tag{7.57}$$

and consequently, the optimal gain matrices for the prescribed impedance model and the human force gain becomes

$$\bar{B} = \begin{bmatrix} 71.704 & -0.000 \\ -0.000 & 71.704 \end{bmatrix}, \; \bar{K} = \begin{bmatrix} 70.758 & 0.000 \\ -0.000 & 70.758 \end{bmatrix}, \; K_h = \begin{bmatrix} 0.458 & 0.000 \\ 0.000 & 0.458 \end{bmatrix}$$

Figs. 7.11 and 7.12 show the results of implementing the proposed method using the obtained optimal gains for the prescribed impedance model. It can be seen that the system states follow the desired trajectories very fast because of the proper gains chosen for the prescribed impedance model. To compare these results to the results of another set of arbitrary gains, Figs. 7.13 and 7.14 show the results obtained if the gains of Case 1 in previous subsection are used. Comparing figures 7.11 and 7.12 to 7.13 and 7.14 confirms that the performance of the overall system for the set of optimal gains is by far better than those of Case 1.

It is now shown the proposed online IRL Algorithm 7.1 gives the same solution as the offline solution, but without requiring the knowledge of the human dynamics. To initialize the value function in Algorithm 7.1, we assume that $K_d = \bar{K}_d + \Delta K_d$ and $K_p = \bar{K}_p + \Delta K_p$, where $\bar{K}_d$ and $\bar{K}_p$ are nominal parameters for an expert human and ΔK_d and ΔK_p changes from one human to another. Note that matrix A_q in A is known and so we can solve the ARE with matrix A containing only nominal values of the human dynamics. This gives us a very appropriate initial value for the value function kernel matrix P.

Figs. 7.13 and 7.14 show the convergence of the control gain and the value function kernel matrix to their optimal values using online Algorithm 7.1. Note that Algorithm 7.1 converges fast after only two iterations because we initialized the value function kernel matrix in an appropriate way. The final gain and kernel matrix found after 16 iterations by Algorithm 7.1 are

$$P_{16} = \begin{bmatrix} 507.364 & 0.000 & 7.076 & -0.000 & 3.378 & -0.003 \\ 0.000 & 507.364 & 0.000 & 7.076 & -0.003 & 3.378 \\ 7.075 & 0.000 & 7.170 & -0.000 & 0.046 & 0.000 \\ -0.000 & 7.076 & -0.000 & 7.170 & -0.000 & 0.046 \\ 3.378 & -0.000 & 0.046 & -0.000 & 99.984 & -0.000 \\ -0.000 & 3.378 & 0.000 & 0.046 & -0.000 & 99.984 \end{bmatrix}$$

$$K_{16} = \begin{bmatrix} -70.758 & 0.000 & -71.704 & 0.000 & 0.461 & 0.000 \\ 0.000 & -70.758 & 0.000 & -71.704 & -0.000 & 0.461 \end{bmatrix} \tag{7.58}$$

By comparing (7.58) to (7.57), it is obvious that the solution found by Algorithm 7.1 is the same as one found using offline algorithm by solving the ARE.

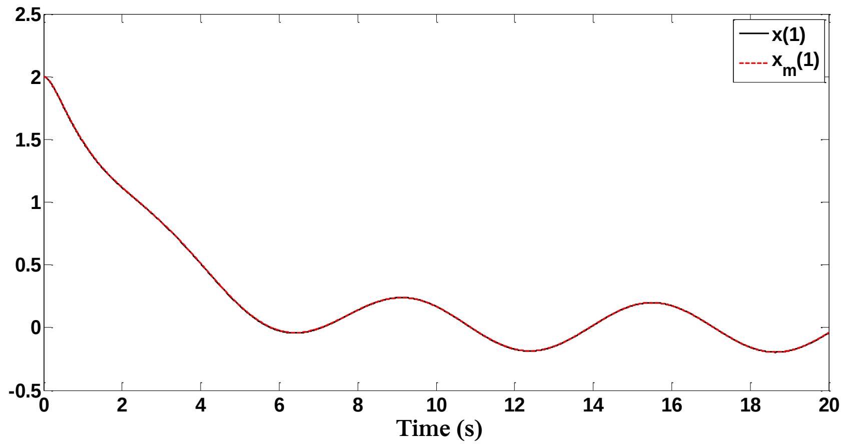

Fig. 7.5. The trajectory of the robot arm and the prescribed impedance model in x direction for Case 1.

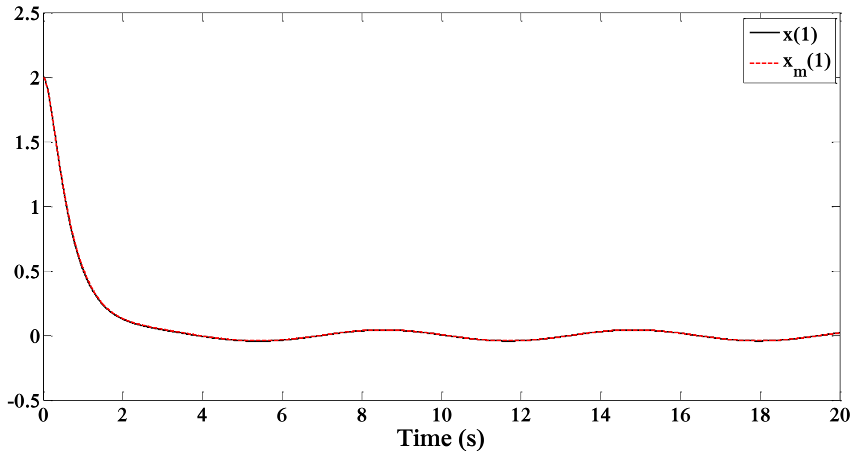

Fig. 7.6. The trajectory of the robot arm and the prescribed impedance model in y direction for Case 1.

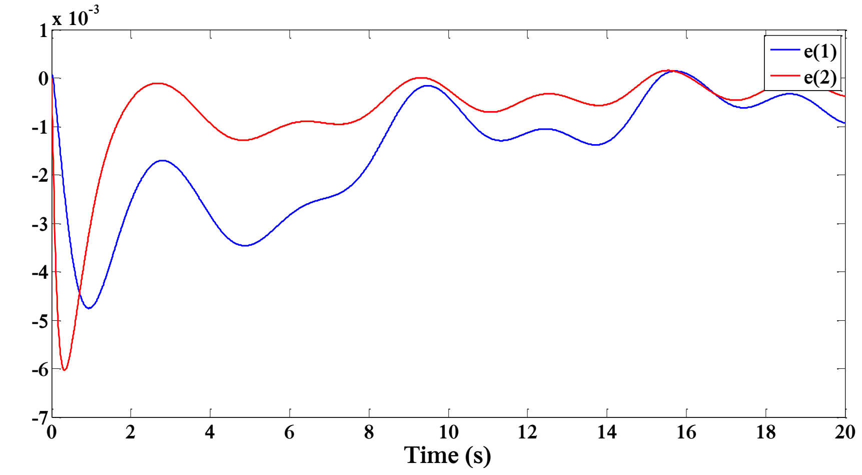

Fig. 7.7. The error between the trajectories of the robot arm and the prescribed impedance model for Case 1.

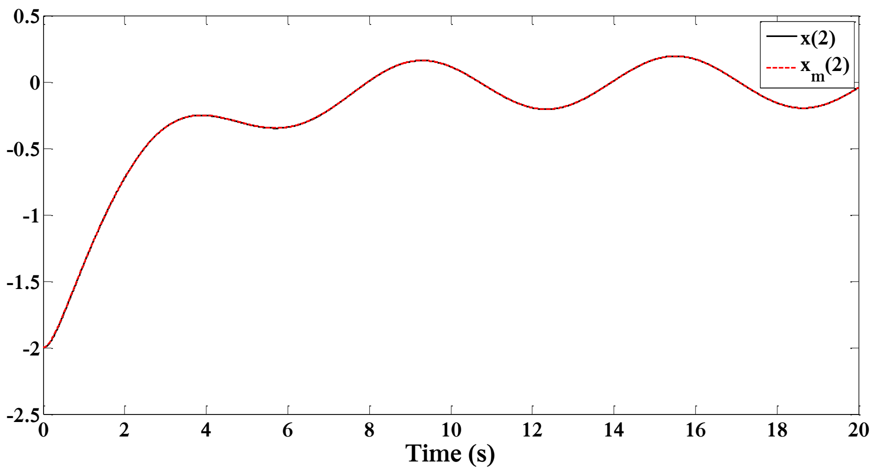

Fig. 7.8. The trajectory of the robot arm and the prescribed impedance model in x direction for Case 2.

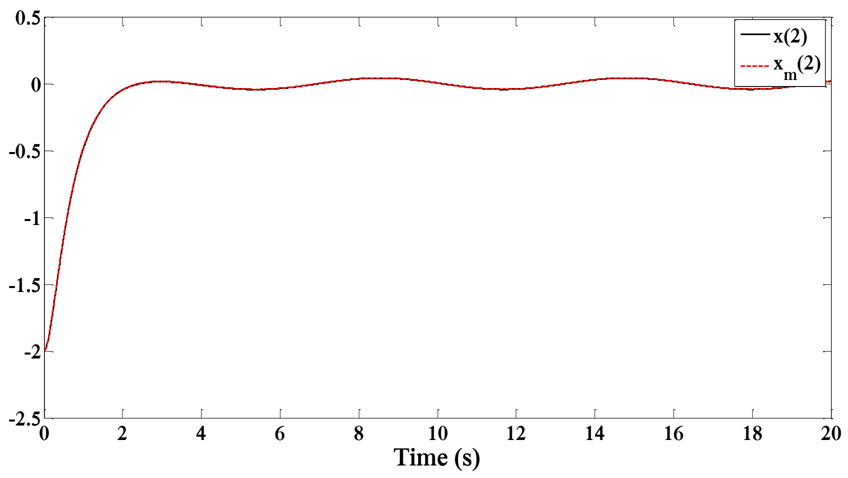

Fig. 7.9. The trajectory of the robot arm and the prescribed impedance model in y direction for Case 2.

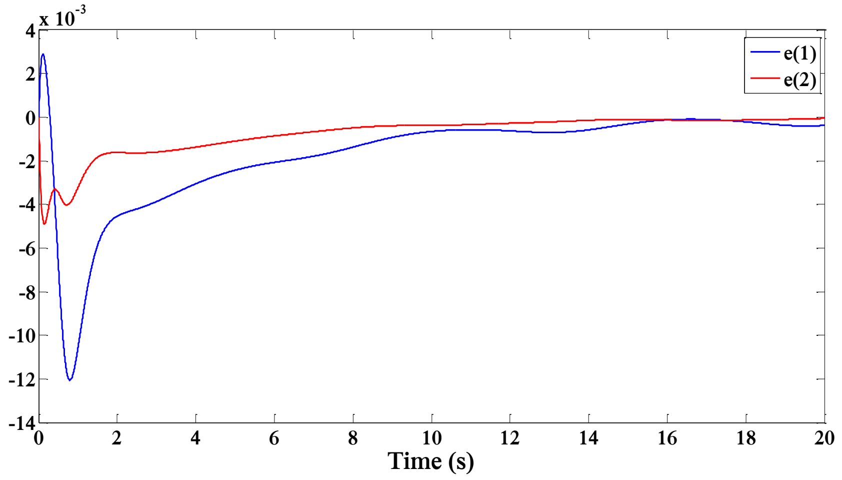

Fig. 7.10. The error between the trajectories of the robot arm and the prescribed impedance model for Case 2.

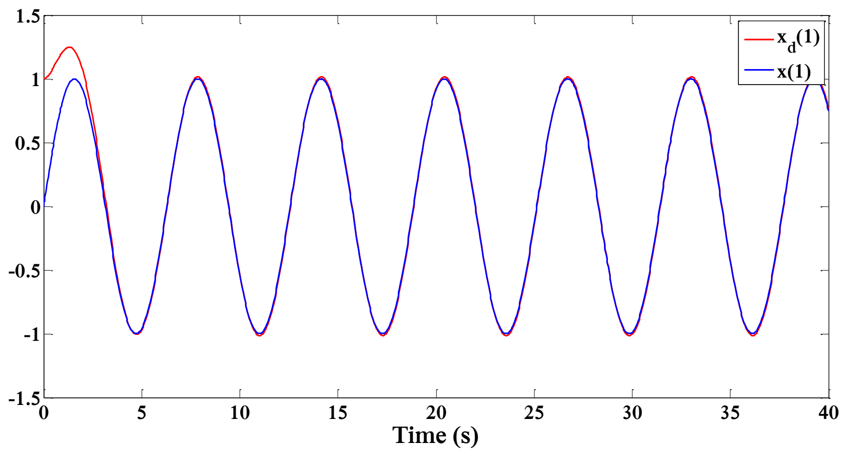

Fig. 7.11. The trajectory of the robot arm and the desired trajectory in x direction.

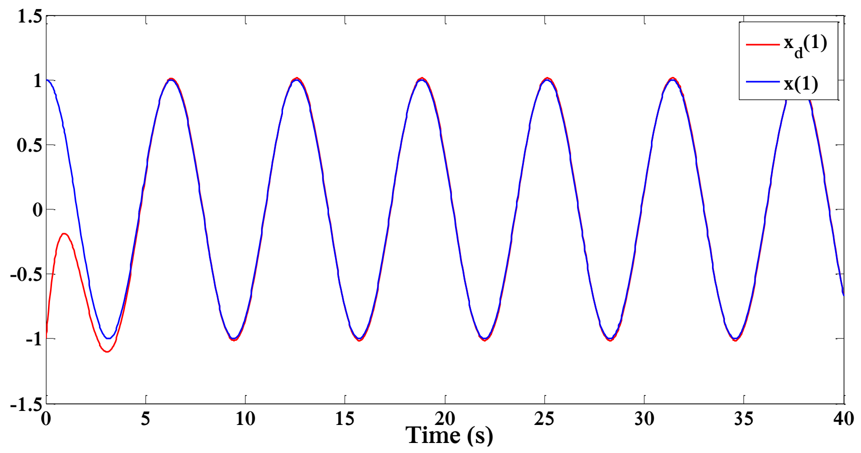

Fig. 7.12. The trajectory of the robot arm and the desired trajectory in y direction.

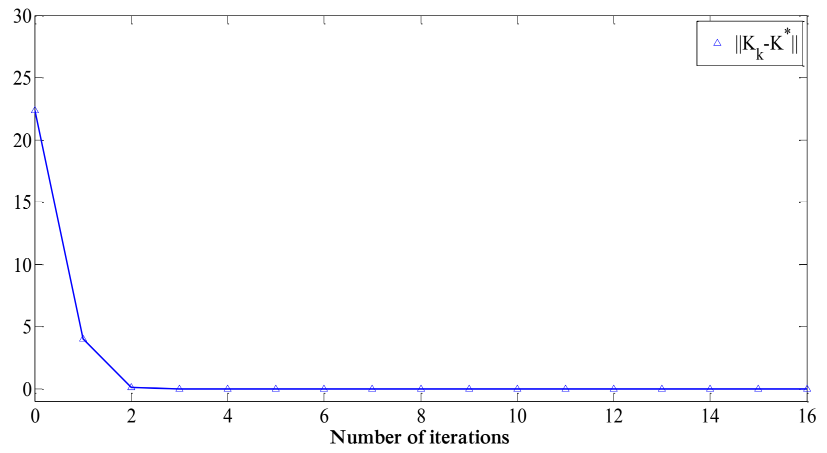

Fig. 7.13. Convergence of the prescribed impedance gains to their optimal values using online Algorithm 7.1.

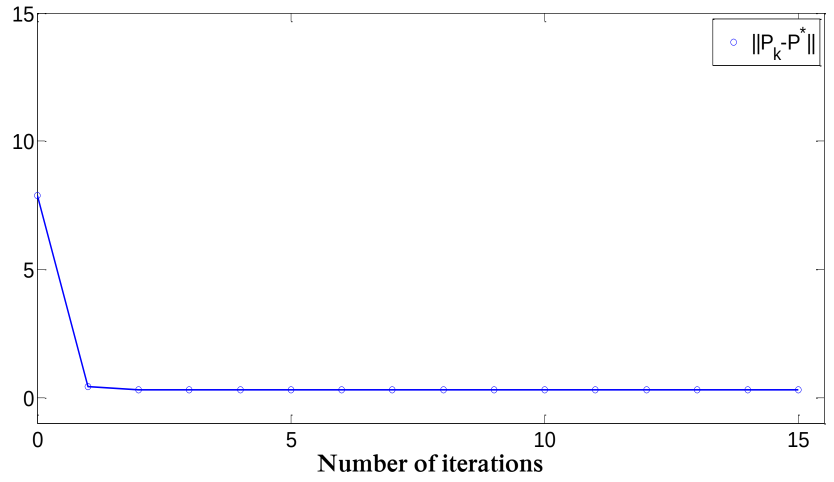

Fig. 7.14. Convergence of the kernel matrix parameters to their optimal values using online Algorithm 7.1.

An practical experiment is now conducted on a PR2 robot at the University of Texas at Arlington Research Institute. Fig.7.15 shows the PR2 robot and the experimental setup. In this experiment, the human operator holds the gripper of the PR2 to perform point-to-point motion between red and blue points along the y axis, as can be seen in Fig. 7.15. Human force is measured using an ATI Mini40 FT sensor attached between the gripper and forearm of the PR2. The controller is implemented using the real-time controller manager framework of the PR2 in ROS Groovy. The real-time loop on the PR2 runs at 1000Hz and communicates with the sensors and actuators on an EtherCAT network.

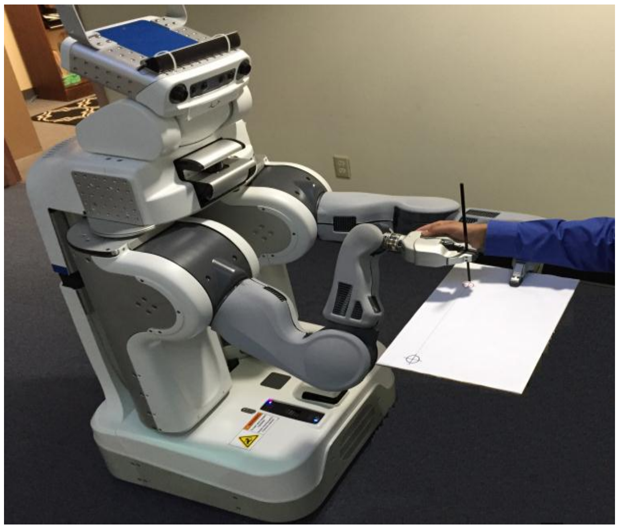
Fig. 7.15. PR2 robot and the experimental setup

The proposed controller is now implemented to this HRI system. Figs. 7.16-7.18 show the results of this experiment. Fig. 7.16 shows that the trajectory of the robot tracks the trajectory of the prescribed impedance model in the inner loop. Fig. 7.17 shows the outer-loop controller performance. At the beginning, the prescribed impedance model is initialized with a set of non-optimal parameters and thus the performance of the overall systems is not satisfactory. However, after a short time of interaction between the human and the robot, the outer-loop controller learns the optimal parameters for the prescribed impedance model and therefore the HRI system tracks the desired trajectory successfully. Fig. 7.18 shows how the human force is reduced after the learning is performed and the optimal set of the prescribed impedance model is found by the outer-loop controller.

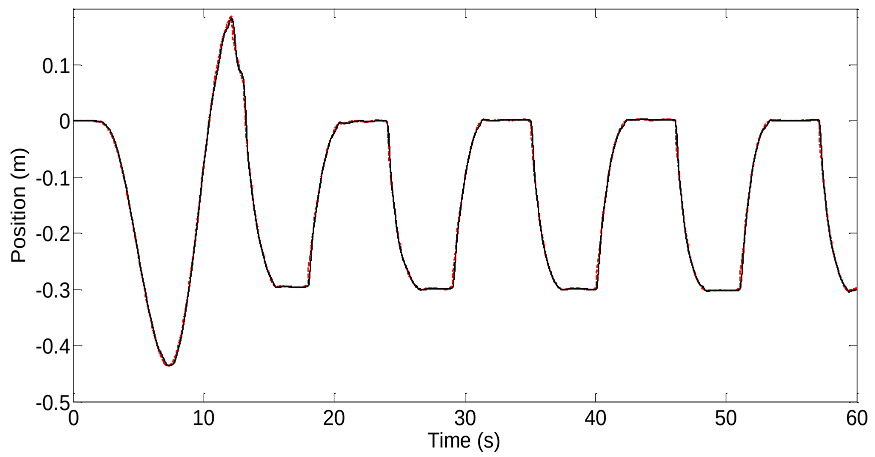

Fig. 7.16. The inner loop results: the trajectory of the prescribed impedance control (red) versus the robot trajectory (black).

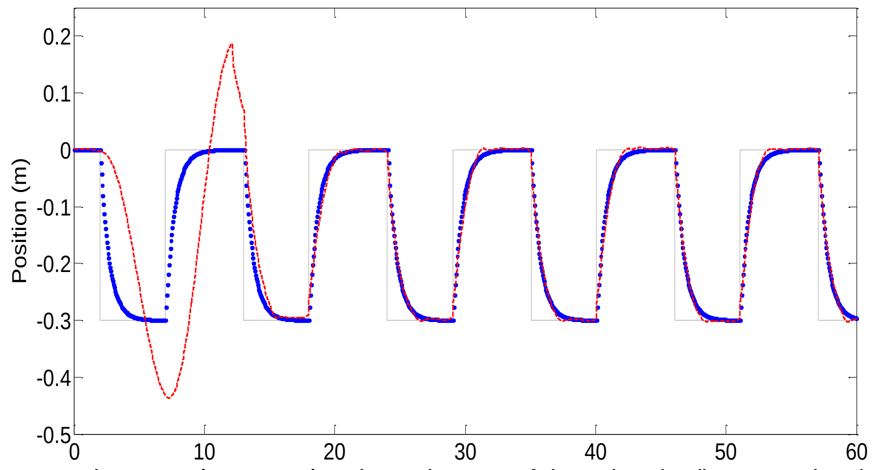

Fig. 7.17. The outer-loop results: the trajectory of the robot (red) versus the desired trajectory (blue).

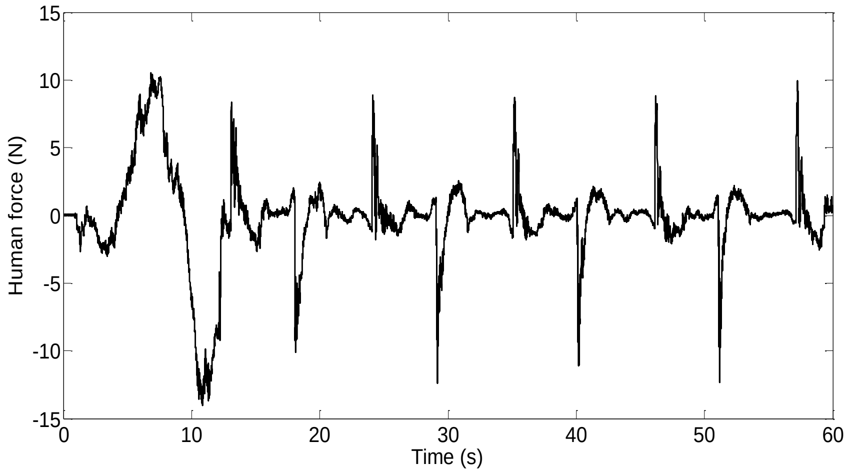

Fig. 7.18. The human force

7.6. Conclusion

A novel human-robot interaction control design method is presented inspired by the human factors studies. The proposed control structure has two control loops. The first loop is an inner control loop which makes the unknown nonlinear robot look like a prescribed robot impedance model. In contrast to the previous trajectory tracking based methods, the proposed inner loop does not require the knowledge of the task or the prescribed impedance parameters. This decomposes the robot-specific control design from the task specific design. The second loop is a task-specific loop which includes the human, the robot and their interaction and finds the optimal parameters of the prescribed impedance parameters to assist the human to perform the task with less effort and optimal performance.

Chapter 8

CONCLUSION AND FUTURE WORK

Reinforcement learning (RL) algorithms have been developed in this dissertation to solve the optimal tracking control problem (OTCP) for unknown continuous-time dynamical systems. This is in contrast to the existing model-free RL-based method methods which are limited to optimal regulation problems. Both on-policy and off-policy RL algorithms are employed. The proposed approach is extended to the design of a model-free RL-based solution to the optimal output synchronization of heterogenous muti-agent systems. This method does not require to explicity solve the output regulation equation and thus does not require any knowledge of the system dynamics. Moreover, in contrast to existing oprimal tracking solutions, the proposed approach can guarantee on the remaining control inputs on their permitted bounds during and after learning. An RL-bsed model-free output-feedback controller is also designed for the first time to solve both oprimal regulation and optimal tracking problems. Finally, a model-free RL based method is design for the human-robot interaction system to help the robot adapts itself to the level of the human skills. This assists the human operator to perform a given task with minimum workload demands and optimizes the overall human-robot system performance.

The following are some of the directions for continuation of this work.

1- Many practical systems such as advanced military robotic systems will be required to operate and co-operate in highly dynamic and challenging environments and make fast skilled decisions. The future goal is to design new classes of fast satisficing systems that deliver prescribed aspiration levels of satisfactory results. These new controllers will have a structure that consists of basic fast feedback loops that operate in normal situations, with additional control and decision loops that are recruited when risky decisions are detected. The proposed control structure will be used for control of military autonomous dynamic agents in highly constrained, data-overload, uncertain, and risky environments.

2- Engineering applications of multi-agent systems such as power and energy systems and robotic networks are prone to cyber-physical attacks. A cyber-physical attacker attempts to prevent the multi-agent system from accomplishing a desired functionally by injecting nonzero signals into the actuator input or sensor output signals of misbehaving/compromised agents. This has led to the emergence of new challenges in the design of RL-based secure distributed controllers for systems with uncertainties to sustain some notion of acceptable behavior of multi-agent systems under attack.

3- The design of model-free RL-based controllers for solving optimal OTCP for non-affine systems will be considered.

4- The design of model-free RL-based controllers for heterogeneous nonlinear multi-agent systems will be considered.

Appendix A

Proof of Theorem 3.4

Consider the following Lyapunov function

$$J(t) = V(t) + \frac{1}{2}\tilde{W}_1(t)^T \alpha_1^{-1} \tilde{W}_1(t) + \frac{1}{2}\tilde{W}_2(t)\alpha_2^{-1}\tilde{W}_2(t)^T \tag{A.1}$$

where $V(t)$ is the optimal value function. The derivative of the Lyapunov function is given by

$$\dot{J} = \dot{V} + \tilde{W}_1^T {\alpha_1}^{-1} \dot{\tilde{W}}_1 + \tilde{W}_2^T {\alpha_2}^{-1} \dot{\tilde{W}}_2 \tag{A.2}$$

Before evaluating (A.2), note that putting (3.67) and the tracking HJB (3.69) in the IRL tracking Bellman equation (3.74) gives

$$\begin{aligned}
\hat{e}_B(t) &= \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(X^T Q_T X + \hat{U} + \gamma \hat{W}_1(t)^T \phi - \hat{W}_1(t)^T \nabla\phi (F - G\lambda \tanh(\hat{D}))\, d\tau = \\
&= \int_{t-T}^{t} e^{-\gamma(\tau-t+T)}(\hat{U} - U - \gamma \hat{W}_1(t)^T \phi + \hat{W}_1(t)^T \nabla\phi (F - G\lambda \tanh(\hat{D})) + \gamma W_1(t)^T \phi - \\
&\quad W_1^T \nabla\phi (F - G\lambda \tanh(\hat{D})) + \varepsilon_{HJB})\, d\tau
\end{aligned} \tag{A.3}$$

where $\hat{U}$ is defined in (3.75) and is given by

$$\hat{U} = \hat{W}_2^T \nabla\phi\, G\lambda \tanh(\hat{D}) + \lambda^2 \bar{R} \ln(\underline{\mathbf{1}} - \tanh^2(\hat{D})) \tag{A.4}$$

and

$$U = W_1^T \nabla\phi\, G\lambda \tanh(D) + \lambda^2 \bar{R} \ln(\underline{\mathbf{1}} - \tanh^2(D)) \tag{A.5}$$

is the cost (3.15) for the optimal control input $u = -\lambda \tanh\left((1/2\lambda) R^{-1} G^T \nabla\phi^T W_1\right)$. Using (3.67) and some manipulations, (A.3) becomes

$$\begin{aligned}
\hat{e}_B(t) = -\Delta\phi^T \tilde{W}_1(t) + \int_{t-T}^{t} e^{-\gamma(\tau-t+T)} &(\hat{U} - U + W_1^T \nabla\phi (F - G\lambda \tanh(\hat{D})) - W_1^T \nabla\phi \times \\
&(F - G\lambda \tanh(D)) + \varepsilon_{HJB}) d\tau
\end{aligned} \tag{A.6}$$

Using (A.4) and (A.5) and some manipulations, $\hat{U} - U$ can be written as

$$\hat{U}-U=\hat{W}_2^T\nabla\phi\, G\,\lambda\tanh(\hat{D})+\tilde{W}_2^T\nabla\phi\, G\lambda\,\mathrm{sgn}(\hat{D})-W_1^T\nabla\phi\, G\,\lambda\tanh(D)-W_1^T\nabla\phi\times G\lambda\Big[\mathrm{sgn}(\hat{D})-\mathrm{sgn}(D)\Big]+\lambda^2\bar{R}\,(\varepsilon_{\hat{D}}-\varepsilon_D) \tag{A.7}$$

where $\varepsilon_{\hat{D}}$ and ε_D are some bounded approximation errors. Substituting (A.7) in (A.6) gives

$$\hat{e}_B(t)=-\Delta\phi^T\tilde{W}_1(t)+\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\tilde{W}_2^T M\, d\tau+E \tag{A.8}$$

where

$$M=\nabla\phi\;\, G\,\lambda\;(\tanh(\hat{D})-\mathrm{sgn}(\hat{D})) \tag{A.9}$$

and

$$E=\int_{t-T}^{t} e^{-\gamma(\tau-t+T)}\;\, W_1^T\nabla\phi\, G\,\lambda\,(\mathrm{sgn}(\hat{D})-\mathrm{sgn}(D))\;+\lambda^2\bar{R}\,(\varepsilon_{\hat{D}}-\varepsilon_D)+\varepsilon_{HJB}\;\, d\tau \tag{A.10}$$

Note that M and E are bounded.

We now evaluate the derivative of the Lyapunov function (A.2). For the first term of (A.2), one has

$$\dot{V}=W_1^T\nabla\phi(F-G\,\lambda\tanh(\hat{D}))+\varepsilon_0 \tag{A.11}$$

where

$$\varepsilon_0(x)=\nabla\varepsilon^T(F-G\,\lambda\tanh(\hat{D})) \tag{A.12}$$

According to Assumption 3.2 and the definition of G in (3.12), one has

$$\|G\|\le b_G \tag{A.13}$$

Using Assumption 3.4, (3.81) and (A.13), and taking norm of ε_0 in (A.12) yields

$$\|\,\varepsilon_0(x)\,\|\le b_{\varepsilon x}b_{F1}\|e_d\|+\lambda\, b_{\varepsilon x}b_G+b_{F2} \tag{A.14}$$

Using the tracking HJB equation (3.69), the first term of (A.11) becomes

$$W_1^T \nabla\phi\, F = -e_d^{\,T} Q\, e_d - U - \gamma W_1^T \phi + W_1^T \nabla\phi\, G\, \lambda \tanh(D) + \varepsilon_{HJB} \tag{A.15}$$

where $U > 0$ and it is defined in (A.5). Also, using $W_1 = \hat{W}_2 + \tilde{W}_2$, and the fact $x^T \tanh(x) > 0\ \forall x$, for the second term of (A.11) one has

$$W_1^T\, \nabla\phi\, G \lambda \tanh(\hat{D}) > \tilde{W}_2^T \nabla\phi\, G\, \lambda \tanh(\hat{D}) \tag{A.16}$$

Using (A.14)- (A.16) and Assumption 3.4, (A.11) becomes

$$\dot{V} < -\lambda_{\min}(Q) \left\|e\right\|^2 + k_1 \left\|e\right\| + k_2 - \tilde{W}_2^T \nabla\phi\, G\, \lambda \tanh(\hat{D}) \tag{A.17}$$

where $k_1 = b_{\varepsilon x} b_{F1}$ and $k_2 = 2\lambda\, b_G b_{\phi x} \left\|W_1\right\| + \gamma \left\|W_1\right\| \left\|\phi\right\| + \lambda\, b_{\varepsilon x} b_G + b_{F2} + \varepsilon_h$, and ε_h is the bound for ε_{HJB}.

For the second term of (A.2), using (A.8) in (3.59), $\dot{\tilde{W}}_1(t)$ becomes

$$\dot{\tilde{W}}_1(t) = -\alpha_1\, \Delta\bar{\phi}\, \Delta\bar{\phi}^T \tilde{W}_1(t) - \alpha_1 \frac{\Delta\bar{\phi}}{m} \int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \tilde{W}_2^T(\tau)\, M d\tau - \alpha_1 \frac{\Delta\bar{\phi}}{m} E \tag{A.18}$$

and therefore

$$\begin{aligned} \dot{J}_1 = \tilde{W}_1^T(t)\alpha_1^{-1}\dot{\tilde{W}}_1(t) = -\tilde{W}_1^T(t)\Delta\bar{\phi}\, \Delta\bar{\phi}^T \tilde{W}_1(t) - \tilde{W}_1^T(t) \frac{\Delta\bar{\phi}}{m} \int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \tilde{W}_2^T(\tau) M\, d\tau \\ - \tilde{W}_1^T(t) \frac{\Delta\bar{\phi}}{m} E \end{aligned} \tag{A.19}$$

For small enough reinforcement interval, the integral term of (A.19) can be approximated by the right-hand rectangle method (with only one rectangle) as

$$\int_{t-T}^{t} e^{-\gamma(\tau - t + T)} \tilde{W}_2^T(\tau)\, M\, d\tau \approx T e^{-\gamma T}\, M^T \tilde{W}_2(t) \tag{A.20}$$

Using (A.20) in (A.19) gives

$$\dot{J}_1 = -\tilde{W}_1^T(t)\Delta\bar{\phi}\, \Delta\bar{\phi}^T \tilde{W}_1(t) - \tilde{W}_1^T(t) \frac{\Delta\bar{\phi}}{m} E - \frac{T}{m} e^{-\gamma T} \tilde{W}_1^T(t)\, \Delta\bar{\phi}\, M^T \tilde{W}_2(t) \tag{A.21}$$

By applying the Young inequality to the last term of (A.21), one has

$$\frac{T}{m}e^{-\gamma T}\tilde{W}_1^T(t)\Delta\bar{\phi}\, M^T\tilde{W}_2(t) \le \frac{T^2e^{-2\gamma T}}{2\varepsilon}\tilde{W}_1^T(t)\Delta\bar{\phi}\,\Delta\bar{\phi}^T\tilde{W}_1(t) + \frac{\varepsilon}{2m^2}\tilde{W}_2(t)^T M\, M^T\tilde{W}_2(t) \qquad \text{(A.22)}$$

for every $\varepsilon > 0$. Using (A.22) in (A.21) yields

$$\dot{J}_1 \le -d\,\tilde{W}_1^T(t)\Delta\bar{\phi}\Delta\bar{\phi}^T\tilde{W}_1(t) - \tilde{W}_1^T(t)\frac{\Delta\bar{\phi}}{m}E + \frac{\varepsilon}{2m^2}\tilde{W}_2(t)^T M\, M^T\tilde{W}_2(t) \qquad \text{(A.23)}$$

where

$$d = 1 - \frac{T^2e^{-2\gamma T}}{2\varepsilon} \qquad \text{(A.24)}$$

Define T_0 as a constant that satisfies

$$T_0^2e^{-2\gamma T_0} = 2\varepsilon \qquad \text{(A.25)}$$

Then $d > 0$ if $T < T_0$.

Finally, for the last term of (A.2), using (3.79), and definitions $\hat{W}_1(t) = W_1 - \tilde{W}_1(t)$ and $\hat{W}_2(t) = W_1 - \tilde{W}_2(t)$, one has

$$\dot{J}_2 = \tilde{W}_2(t)^T\alpha_2^{-1}\dot{\tilde{W}}_2(t) = -\tilde{W}_2(t)^T Y\,\tilde{W}_2(t) + \tilde{W}_2(t)^T\lambda\nabla\phi\, G\,\tanh(\hat{D}) + \tilde{W}_2(t)^T k_3 \qquad \text{(A.26)}$$

where $k_3 = -\lambda\nabla\phi\, G\tanh((1/2\lambda)R^{-1}G^T\nabla\phi^T\hat{W}_1) + Y\,W_1 + \lambda\nabla\phi\, G\tanh^2(\hat{D})e_u$. Based on definitions of e_u, G in (3.77) and Assumptions 3.3 and 3.4, k_3 is bounded.

Using (A.17), (A.23) and (A.26) into (A.2), $\dot{J}$ becomes

$$\begin{aligned}\dot{J} < -\lambda_{\min}(Q)\|e\|^2 + k_1\|e\| + k_2 - d\,\tilde{W}_1^T(t)\Delta\bar{\phi}\,\Delta\bar{\phi}^T\tilde{W}_1(t) - \tilde{W}_1(t)^T\frac{\Delta\bar{\phi}}{m}E - \\ \tilde{W}_2(t)^T N\,\tilde{W}_2(t) + \tilde{W}_2(t)^T k_3\end{aligned} \qquad \text{(A.27)}$$

where

$$N = Y - \frac{\varepsilon}{2m^2} M M^T \tag{A.28}$$

If we choose T and Y such that d in (A.24) and N in (A.28) are bigger than zero, then $\dot{J}$ becomes negative, provided that

$$\|e\| > \frac{k_1}{2\lambda_{\min}(Q)} + \sqrt{\frac{k_1{}^2}{4\lambda_{\min}^2(Q)} + \frac{k_2}{\lambda_{\min}(Q)}} \tag{A.29}$$

$$\left\| \Delta\bar{\phi}^T \tilde{W}_1 \right\| > \frac{E}{d} \tag{A.30}$$

$$\left\| \tilde{W}_2 \right\| > \frac{k_3}{\lambda_{\min}(N)} \tag{A.31}$$

References

[1] Abdollahi, F., Talebi, H. A., & Patel, R. V. (2006). A stable neural network observer with application to flexible-joint manipulators, in Proc. 9th Int. Conf. Neural Inform. Process. (pp. 1910–1914).

[2] Abu-Khalaf M., & Lewis F. L. (2008). Neurodynamic Programming and Zero-Sum Games for Constrained Control Systems, IEEE Transactions on Neural Networks, 19(7), 1243-1252.

[3] Abou-Khalaf, M., & Lewis, F. L. (2005). Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica, 41, 779–791.

[4] Aliyu M.D.S. (2011). Nonlinear H_∞-Control, Hamiltonian Systems and Hamilton-Jacobi Equations, Boca Raton : CRC Press.

[5] Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2007). Model-free Q-learning designs for linear discrete-time zero-sum games with application to H_∞ control. Automatica, 43(3), 473–481.

[6] Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2008). Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 38(4), 943–949.

[7] Baird, L. C. III (1994). Reinforcement learning in continuous time: advantage updating. In Proc. of ICNN.

[8] Ball, A.J, Kachroo, P., & Krener, A.J. (1999). H_∞ Tracking Control for a Class of Nonlinear Systems, IEEE Transactions of Automatic Control, 44(6), 1202-1206.

[9] Barto, J. Si. A., Powell, W., & Wunch, D. (2004). Handbook of learning and approximate dynamic programming. John Wiley.

[10] Barbieri, E., & Alba-Flores, R. (2000). On the infinite-horizon LQ tracker," Systems & Control Letters, 40 (2), 77–82.

[11] Barbieri, E., & Alba-Flores, R. (2000). Real-time Infinite Horizon Linear-Quadratic Tracking Controller for Vibration Quenching in Flexible Beams. IEEE Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, (pp. 38-43).

[12] Baron, S., Kleinman, D.L., & Levison, W.H. (1970). An optimal control model of human response. Part II: Prediction of human performance in a complex task," Automatica, 6, 371–383.

[13] Basar, T., & Bernard, P. (1995). H∞ Optimal Control and Related Minimax Design Problems. Birkhäuser: Boston, MA.

[14] Basar, T., Olsder, G.J. (1999). Dynamic Noncooperative Game Theory (2nd edn). SIAM, Vol. 23, SIAM's Classic in Applied Mathematics: Philadelphia, PA.

[15] Bertsekas, D. P., & Tsitsiklis, J. N. (1996). Neuro-dynamic programming. MA: Athena Scientific.

[16] Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K. G, Lewis, F. L, & Dixon, W. E. (2012). A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems. Automatica, 49, 82–92.

[17] Bradtke, S. J., Ydestie, B. E., & Barto, A. G. (1994). Adaptive linear quadratic control using policy iteration. In: Proc. of ACC (pp. 3475–3476).

[18] Chen, F.C., & Khalil, H.K. (1992). Adaptive control of nonlinear systems using neural networks. International Journal of Control, 55 (6), 1299-1317.

[19] Cohen, M., & Flash, T. (1991). Learning impedance parameters for robot control using an associative search networks. IEEE Transactions on Robotics and Automation, 7, 382–390.

[20] De Persis, C, & Jayawardhana, B. On the internal model principle in the coordination of nonlinear systems. IEEE Transactions on Control of Network Systems, 1(3), 272–282.

[21] Devasia, S., Chen, D., & Paden., B. (1996). Nonlinear Inversion-based Output Tracking, IEEE Transactions of Automatic Control, 14(7), 930-942.

[22] Ding, H., & Wu, J. (2007). Point-to-point motion control for a high-acceleration positioning table via cascaded learning schemes. IEEE Transactions on Industrial Electronics, 54 (5), 2735–2744.

[23] Doya, K. (2000). Reinforcement learning in continuous time and space. Neural Computation, 12, 219-245.

[24] Dierks, T., & Jagannathan, S. (2009). Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics. In Joint 48th IEEE conference on decision and control and 28th Chinese control conference Shanghai, PR China (pp. 6750–6755).

[25] Dierks, T., & Jagannathan, S. (2010). Optimal control of affine nonlinear continuous-time systems. In Proc. Am. control conf. (pp. 1568–1573).

[26] Doyle, J.H., Glover, K., Khargonekar, P., & Francis, B. (1989). State-space solutions to standard H2 and H1 control problems. IEEE Transactions on Automatic Control, 34(8), 831–847.

[27] Engwerda, J. (2005). LQ Dynamic Optimization and Differential Games. John Wiley & Sons.

[28] Feng, Y., Anderson B.D., & Rotkowitz M. (2009). A game theoretic algorithm to compute local stabilizing solutions to HJBI equations in nonlinear H1 control. Automatica, 45(4), 881–888.

[29] Finlayson, B. A. (1990). The method of weighted residuals and variational principles. New York: Academic Press.

[30] Franklin, S., Wolpert, D.M., & Franklin, D.W. (2012). Visuomotor feedback gains upregulate during the learning of novel dynamics. Journal of Neurophysiology, 108, 467– 478.

[31] Furuta, K., Kado, Y., & Shiratori, S. (2006). Assisting control in Human Adaptive Mechatronics-single ball juggling. In: Proc. IEEE Int. Conf. Control Appl., (pp. 545–550).

[32] Ge, S.S., Lee, T.H., & Harris, C.J. (1998). Adaptive Neural Network Control of Robotic Manipulators, World Scientific, Singapore.

[33] Ge, S.S., Lee, T.H., & Wang, Z.P. (2001). Adaptive neural network control for smart materials robots using singular perturbation technique. Asian Journal of Control, 3(2), 143-155.

[34] Ge, S.S., Hang, C.C., Woon, L.C., & Chen, X.Q. (1998). Impedance control of robot manipulators using adaptive neural networks. International Journal of Intelligent Control Systems, 2(3), 433-452.

[35] Gribovskaya, E., Kheddar, A., & Billard, A. (2011). Motion Learning and Adaptive Impedance for Robot Control during Physical Interaction with Humans. IEEE Int. Conf. Robot. Autom., Shanghai, (pp. 4326-4333).

[36] Hardy, G. Littlewood, J. & Polya, G. (1989). Inequalities, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press.

[37] Hogan., N. (1985). Impedance Control: An Approach to Manipulation. I –Theory. II – Implementation. III – Applications. ASME Transactions on Journal of Dynamic Systems, Measurement, and Control, 107, 1–24.

[38] Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks, 3, 551–560.

[39] Hong, Y., Wang, X., & Jiang, Z.P. (2013). Distributed output regulation of leader–follower multi-agent systems. International Journal of Robust and Nonlinear Control, 23(1), 48–66.

[40] Howard, R. A. (1960). Dynamic programming and markov processes. Cambridge, MA: MIT Press.

[41] Huang, J. & Chen, Z. (2004). A general framework for tackling output regulation problem. IEEE Transactions on Automatic Control, 49(12), 2203-2218.

[42] Huang, J. Nonlinear output regulation, SIAM Press, Vol 1, Philadelphia 2004.

[43] Huang, Y., & Liu, D. (2014). Neural-network-based optimal tracking control scheme for a class of unknown discrete-time nonlinear systems using iterative ADP algorithm," Neurocomputing, 125, 46–56.

[44] Huang, L., Ge, S.S., & Lee, T.H. (2002). Neural Network Based Adaptive Impedance Control of Constrained Robots, Proceedings of the IEEE Int. Symp. Intell. Control Vancouver, (pp. 615-619).

[45] Ioannou, P., & Sun, J. (1996). Robust Adaptive Control. Prentice Hall, New Jersey.

[46] Isidori, A. (1994). H∞ control via measurement feedback for affine nonlinear systems. International Journal of Robust and Nonlinear Control, 4, 553–574.

[47] Isidori, A., Lin, W. (1998). Global L2-gain design for a class of nonlinear systems. Systems and Control Letters, 34(5), 245–252.

[48] Jadbabaie, A., Lin, J., & Morse, A.S (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control, 48, 988-1001.

[49] Jagannathan, S., & Lewis, F.L. (1996). Multilayer discrete-time neural net controller with guaranteed performance. IEEE Transactions on Neural Networks, 7(1), 107-130.

[50] Janardhanan, S., & Bandyopadhyay, B. (2006). Output Feedback Sliding-Mode Control for Uncertain Systems Using Fast Output Sampling Technique. IEEE Transactions on Industrial Electronics, 53 (5), 1677-1682.

[51] Jiang, Y., & Jiang, Z.P. (2012). Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. Automatica, 48, 2699–2704.

[52] Jiang, Y., & Jiang, Z.P. (2013). Robust adaptive dynamic programming with an application to power systems. IEEE Transactions on Neural Networks and Learning Systems, 24 (7), 1150-1156.

[53] Jung, S., & Hsia, T.C. (1998). Neural network impedance force control of robot manipulator. IEEE Transactions on Industrial Electronics, 45(3), 452–465.

[54] Khalil, H. K. (2002). Nonlinear systems (3rd ed.). Prentice Hall.

[55] Kiumarsi, B., Lewis, F.L., Modares, H., Karimpour, A., & Naghibi-Sistani, M.B. (2014).Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica, 50, 1167–1175.

[56] Kleinman, D.L. (1968). On an iterative technique for Riccati equation computations. IEEE Transactions on Automatic Control, 18 (1), 114-115.

[57] Kosuge, K., Furuta, K., & Yokoyama, T. (1987). Virtual internal model following control of robot arms. In: Proc. IEEE Int. Conf. Robot. Autom., (pp. 1549–1554).

[58] Kurihara, K., Suzuki, S., Harashima, F., & Furuta, K. (2004). Human adaptive mechatronics (HAM) for haptic system. In: Proc. 30th IEEE Annual Conf. Ind. Elect., (pp. 647-652).

[59] Lee, J.Y., Park, J.B., & Choi, Y.H. (2012). Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. Automatica, 48, 2850–2859.

[60] Lewis, F.L., Vrabie, D., & Syrmos, V. (2012). Optimal Control, Third edition, Wiley.

[61] Lewis, F.L., & Liu, D. (Eds.) (2012). Reinforcement learning and approximate dynamic programming for feedback control. New York: Wiley.

[62] Lewis, F.L., & Vamvoudakis, K. (2011). Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 41(1), 14–23.

[63] Lewis, F.L., Vamvoudakis, K., & Vrabie, D. (2013). Optimal adaptive control and differential games by reinforcement learning principles. London: Institution of Engineering and Technology.

[64] Lewis, F.L., Liu, K., & Yesildirek, A. (1995). Neural net robot controller with guaranteed tracking performance. IEEE Transactions on Neural Networks, 6 (3), 703-715.

[65] Lewis, F.L., Yesildirek, A., & Liu, K. (1996). Multilayer neural net robot controller with guaranteed tracking performance. IEEE Transactions on Neural Networks, 7(2), 388-399.

[66] Lewis, F.L., Jagannathan, S., & Yesildirek, A. (1999). Neural network control of robot manipulators and nonlinear systems. Taylor & Francis.

[67] Lewis, F.L., Campos, J., & Selmic, R. (2002). Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities, Society of Industrial and Applied Mathematics Press, Philadelphia.

[68] Lewis, F.L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits & Systems Magazine, 9, 32-50.

[69] Lewis, F.L., Zhang, H., Hengster-Movric, K., & Das, A. (2014). Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches, Springer-Verlag.

[70] Li, H., Liu, D., & Wang, D. (2014). Integral Reinforcement Learning for Linear Continuous-Time Zero-Sum Games With Completely Unknown Dynamics, IEEE Transactions on Automation Science and Engineering, 11(3), 706-714.

[71] Li, Y., Ge, S.S., Yang, C. (2011). Impedance Control for Multi-Point Human-Robot Interaction. In Proc. 8th Asian Control Conf. (ASCC), Kaohsiung, Taiwan, (pp. 1187-1192).

[72] Lin, S.T., & Tsai, H.C. (1997). Impedance control with online neural network compensator for dual-arm robots. Journal of Intelligent Robotic Systems, 18, 87–104.

[73] Liu, D., & Wei, Q. (2013). Finite-approximation-error-based optimal control approach for discretetime nonlinear systems. IEEE Transactions on Cybernetics, 43, 779–789.

[74] Liu, D., & Wei, Q. (2014). Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Transactions on Neural Networks and Learning Systems, 25(3), 621–634.

[75] Liu, D., Yang, X., & Li, H. (2013). Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics. Neural Computing and Applications, http://dx.doi.org/10.1007/s00521-012-1249-y.

[76] Liu, D., Huang, Y., & Wei, Q. (2013). Neural Network H_∞ Tracking Control of Nonlinear Systems Using GHJI Method, Advances in Neural Networks – ISNN 2013 Lecture Notes in Computer Science Volume 7952, 2013, 186-195.

[77] Lunze, J. (2012). Synchronization of heterogeneous agents. IEEE Transactions on Automatic Control, 57, 2885-2890.

[78] Luo, B., Wu, H.N., & Huang, T. (2014). Off-Policy Reinforcement Learning for H∞ Control Design. IEEE Transactions on Cybernetics, in press.

[79] Lyshevski, S.E. (1998). Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals. In Proceedings of American control conference (pp. 205–209).

[80] Mannava, A., Balakrishnan, S.N., Tang, L., & Landers, R.G. (2012). Optimal tracking control of motion systems. IEEE Transactions on Control Systems and Technology, 20(6), 1548–1556.

[81] Moerder, D.D., & Calise, A.J. (1985). Convergence of a numerical algorithm for calculating optimal output feedback gains. IEEE Transactions on Automatic Control, 30 (9), 900–903.

[82] Murray, J. J., Cox, C. J., Lendaris, G. G., & Saeks, R. (2002) Adaptive dynamic programming. IEEE Transactions on Systems Man, and Cybernetics, Part C: Applications and Reviews, 32, 140–153.

[83] Narendra, K.S., & Parthasarathy, K. Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, 1, 4-27.

[84] Narendra, K.S., & Lewis, F.L. (2001). Special Issue on Neural Network feedback Control, Automatica, 37 (8).

[85] Olfati-Saber, R., & Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control, 49, 1520-1533.

[86] Polycarpou, M.M. (1996). Stable adaptive neural control scheme for nonlinear systems. IEEE Transactions on Automatic Control, 41(3), 447-451.

[87] Polycarpou, M.M., & Ioannou, P.A. (1991). Identification and control using neural network models: design and stability analysis. Technical Report 91-09-01, Department of Electrical Engineering, University of South California.

[88] Polycarpou, M.M., & Ioannou, P.A. (1992). Neural networks as on-line approximators of nonlinear systems. Proc. IEEE Conf. Decision and Control, (pp. 7-12).

[89] Powell, W.B. (2007). Approximate dynamic programming: solving the curses of dimensionality. Wiley-Interscience. Sastry, S. (1991). Nonlinear systems: analysis, stability, and control. Springer-Verlag.

[90] Poznyak, A.S., Yu, W., Sanchez, E.N., & Perez, J.P. (1999). Nonlinear adaptive trajectory tracking using dynamic neural networks. IEEE Transactions on Neural Networks, 10(6), 1402-1411.

[91] Ragazzini, J.R. (1948). Engineering aspects of the human being as a servo-mechanism. Meeting of the American Psychological Association.

[92] Ren, W., Beard, R.W., & Atkins, E.M. (2007). Information consensus in multivehicle cooperative control. IEEE Control Systems Magazine, 27, 71-82.

[93] Ren W., & Beard, R.W. (2008). Distributed Consensus in Multi-Vehicle Cooperative Control, Springer-Verlag, London.

[94] Rovithakis, G.A. (2000). Performance of a neural adaptive tracking controller for multi-input nonlinear dynamical systems. IEEE Transactions on Systems, Man, Cybernetics, Part A, 30(6), 720-730.

[95] Si, J., Barto, A., Powell, W., & Wunsch, D. (2004). Handbook of Learning and Approximate Dynamic Programming, IEEE Press, USA.

[96] Slotine, J. E., & Li, W. (1991). Applied nonlinear control. Englewood Cliffs, NJ: Prentice Hall.

[97] Song, R., Xiao, W., & Wei, Q. (2013). Optimal Tracking Control for a Class of Nonlinear Time-Delay Systems with Actuator Saturation. Advances in Brain Inspired Cognitive Systems-Lecture Notes in Computer Science, 7888, 208-215.

[98] Song, R., Xiao, W., Zhang, H., & Sun, C. (2014). Adaptive dynamic programming for a class of complex-valued nonlinear systems. IEEE Transactions on Neural Networks and Learning Systems, 25(9), 1733-1739.

[99] Stulp, F., Buchli, J., Ellmer, A., Mistry, M., Theodorou, E.A., & Schaal, S. (2012). Model-Free Reinforcement Learning of Impedance Control in Stochastic Environments. IEEE Transactions on Autonomous Mental Development, 4(4), 330-341.

[100]Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning—an introduction. Cambridge, MA: MIT Press. Vamvoudakis, K., & Lewis, F. L. (2010). Online actor–critic algorithm to solve the continuous infinite-time horizon optimal control problem. Automatica, 46, 878–888.

[101]Suzuki, S., Furuta, K. (2012). Adaptive impedance control to enhance human skill on a haptic interface system. Journal of Control Science Engineering, 2012, 1-10.

[102]Suzuki, S., Kurihara, K., Furuta, K., Harashima, F., & Pan, Y. (2005). Variable Dynamic Assist Control on Haptic System for Human Adaptive Mechatronics. In: Proc. 44th IEEE Conf. Decision Control, and European Control Conf.. Seville, Spain, (pp. 4596-4600).

[103] Toussaint G.J., Basar T., & Bullo F. (2000). H∞-Optimal Tracking Control Techniques for Nonlinear Underactuated Systems. Proceedings of the 39& IEEE Conference on Decision and Control Sydney, (pp. 2078-2083).

[104]Tsuji, T., & Tanaka, Y. (2005). Tracking Control Properties of Human–Robotic Systems Based on Impedance Control. IEEE Transactions on Systems, Man, Cybernetics, Part A, 35(4), 523-535.

[105]Tustin, A. (1947). The nature of the operator's response in manual control and its implications for controller design. Journal of Institute of Electrical Engineers – Part IIA, 94, 190–202.

[106]Vamvoudakis, K., & Lewis, F.L. (2010). Online actor-critic algorithm to solve the continuous infinite-time horizon optimal control problem. Automatica, 46, 878–888.

[107]Vamvoudakis, K., Vrabie, D., & Lewis, F.L. (2013). Online adaptive algorithm for optimal control with integral reinforcement learning. International Journal of Robust and Nonlinear Control, http://dx.doi.org/10.1002/rnc, in press.

[108]Vamvoudakis, K., Lewis F.L. (2010). Online solution of nonlinear two-player zero-sum games using synchronous policy iteration. In Proceedings of 49th IEEE Conference on Decision and Control, Atlanta, (pp. 3040–3047).

[109]Vamvoudakis, K., Lewis, F.L. (2011). Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton–Jacobi equations. Automatica, 47(8), 1556–1569.

[110]Vamvoudakis, K., Lewis, F.L., & Hudas, G.R. (2012). Multiagent differential graphical games: Online adaptive learning solution for synchronization with optimality, Automatica, 48(8), 1581–1661.

[111]Van der Schaft A.J. (1992). L2-gain analysis of nonlinear systems and nonlinear state feedback H∞ control, IEEE Transactions on Automatic Control, 37(6), 770–784.

[112]Vrabie, D., & Lewis, F.L. (2009). Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. Neural Networks, 22, 237–246.

[113]Vrabie, D., Pastravanu, O., Abou-Khalaf, M., & Lewis, F.L. (2009). Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica, 45, 477–484.

[114]Vrabie, D., Vamvoudakis, K.G., & Lewis, F.L. (2013). Optimal adaptive control and differential games by reinforcement learning principles. London: Institution of Engineering and Technology.

[115]Wang, C., Li, Y. Sam Ge, S., Tee, K.P., & Lee, T.H. (2013). Continuous Critic Learning for Robot Control in Physical Human-Robot Interaction," In 13th Int. Conf. Control, Autom. Syst. (ICCAS 2013) Gwangju, Korea Oct. 20-23, (pp. 833-838).

[116]Wang, D., Liu, D., & Wei, Q. (2012). Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach. Neurocomputing, 78, 14–22.

[117]Wang F.Y., Zhang H., & Liu D. (2009). Adaptive dynamic programming: an introduction, IEEE Computational Intelligence Magazine, 39-47.

[118]Wang, L.Y., Li, C., Yin, G.G., Guo, L., & Xu, C.Z. (2011). State Observability and Observers of Linear-Time-Invariant Systems under Irregular Sampling and Sensor Limitations. IEEE Transactions on Automatic Control, 56 (112), 2639-2654.

[119]Watkins, C. (1989). Learning from delayed rewards. Ph.D. thesis, England: University of Cambridge.

[120]Wei, Q., Liu, D. (2013). Optimal tracking control scheme for discrete-time nonlinear systems with approximation errors. Advances in Neural Networks – Lecture Notes in Computer Science, 7952, 1-10.

[121] Wei, Q., Liu, D., Shi, G., & Y. Liu, Y. (2015). Multi-batter optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming. IEEE Transactions on Industrial Electronics, 62(7), 4203-4214.

[122] Wei, Q., Liu, D., & Shi, D. (2015). A novel dual iterative Q-learning method for optimal battery management in smart residential environments, IEEE Transactions on Industrial Electronics, 62(4), 2509-2518.

[123]Wei, Q., & Liu, D. (2014). Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming. IEEE Transactions on Industrial Electronics, 61(11), 6399–6408.

[124] Wei, Q., & Liu, D. (2014). Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. IEEE Transactions on Automation Science and Engineering, 11(4), 1020-1036.

[125]Werbos, P.J. (1989). Neural networks for control and system identification. In Proc. IEEE conf. of decision control, Tampa, FL (pp. 260–265).

[126]Werbos, P.J. (1992). Approximate dynamic programming for real time control and neural modeling. In D. A. White, & D. A. Sofge (Eds.), Handbook of intelligent control. Multiscience Press.

[127]Werner, H. (1996). Robust control of a laboratory flight simulator by nondynamic multirate output feedback. in Proc. IEEE Conf. Decision and Control, pp. 1575–1580.

[128]Wieland, P., Sepulchre, R., & Allgower F. (2011). An internal model principle is necessary and sufficient for linear output synchronization. Automatica, 47, 1068-1074.

[129]White, D.A., & Sofge, D.A. (Eds.) (1992). Handbook of Intelligent Control, New York: Van Nostrand Reinhold.

[130] Wu, H.N., & Luo, B. (2012). Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H∞ control. IEEE Transactions on Neural Networks and Learning Systems, 23 (12), 1884–1895.

[131]Wu, H.N., Luo, B. (2013). Simultaneous policy update algorithms for learning the solution of linear continuous-time H∞ state feedback control. Information Sciences, 222, 472–485.

[132]Xu, H., Jagannathan S., & Lewis, F.L. (2012). Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses, Automatica, 48, 1017–1030.

[133]Xu, G., Song, A. (2009). Adaptive Impedance Control Based on Dynamic Recurrent Fuzzy Neural Network for Upper-limb Rehabilitation Robot. IEEE Int. Conf. Control Autom. Christchurch, New Zealand, (pp. 1376-1381).

[134]Yang, T., Saberi, A., Stoorvogel, A.A., &. Grip, H.F. (2014). Output synchronization for heterogeneous networks of introspective right-invertible agents. International Journal of Robust and Nonlinear Control, 24(13), 1821–1844.

[135] Yu, W., DeLellis, P., Chen, G., Bernardo M., & Kurths, J. (2012). Distributed adaptive control of synchronization in complex networks. IEEE Transactions on Automatic Control, 57, 2153-2158.

[136]Zames, G. (1981). Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. IEEE Transactions on Automatic Control, 26(2), 301–320.

[137]Zbikowskim R., & Hunt, K.J. (1996). Neural Adaptive Control Technology, World Scientific, Singapore.

[138]Zhang, H., Cui, L., Zhang, X., & Luo, X. (2011). Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. IEEE Transactions on Neural Networks, 22, 2226–2236.

[139]Zhang, H., Feng, T., Yang G.H., & Liang, H. (2014). Distributed cooperative optimal control for multiagent systems on directed graphs: an inverse optimal approach, IEEE Transactions on Cybernetics, in press.

[140] Zhang, H., Lewis, F.L., & Das, A. (2011). Optimal design for synchronization of cooperative systems: state feedback, observer, and output feedback, IEEE Transactions on Automatic Control, 56(8), 1948–1952.

[141]Zhang, H., Liu, D., Luo, Y., & Wang, D. (2012). Adaptive dynamic programming for control: algorithms and stability. London: Springer-Verlag.

[142]Zhang, H., Wei, Q., & Liu, D. (2011). An iterative approximate dynamic programming method to solve for a class of nonlinear zero-sum differential games. Automatica, 47(1), 207–214.

[143]Zhang, H., Wei, Q., & Luo, Y. (2008). A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 38, 937–942.

Biographical Information

Hamidreza Modares received his B.Sc. from Tehran University, Tehran, Iran and his and M.Sc. from Shahrood University of Technology, Shahrood, Iran. Between 2006 and 2009, he joined the Shahrood University of Technology as a senior lecturer. Since 2012 he has been working as a research/teaching assistant at University of Texas at Arlington Research Institute (UTARI). His work on the design of optimal controllers using reinforcement learning resulted in several journal and conference papers. His main research interests include cyber-physical systems, reinforcement learning, distributed control, robotics, and pattern recognition.