Learning-Based Control: A Tutorial and Some Recent Results

Zhong-Ping Jiang[1], Tao Bian[2] and Weinan Gao[3]

[1] *Tandon School of Engineering, New York University, New York, USA; zjiang@nyu.edu*

[2] *Bank of America, New York, USA; tbian@nyu.edu*

[3] *College of Engineering and Science, Florida Institute of Technology, Florida, USA; wgao@fit.edu*

ABSTRACT

This monograph presents a new framework for learning-based control synthesis of continuous-time dynamical systems with unknown dynamics. The new design paradigm proposed here is fundamentally different from traditional control theory. In the classical paradigm, controllers are often designed for a given class of dynamical control systems; it is a model-based design. Under the learning-based control framework, controllers are learned online from real-time input–output data collected along the trajectories of the control system in question. An entanglement of techniques from reinforcement learning and model-based control theory is advocated to find a sequence of suboptimal controllers that converge to the optimal solution as learning steps increase. On the one hand, this learning-based design approach attempts to overcome the well-known "curse of dimensionality" and the "curse of modeling" associated with Bellman's Dynamic Programming. On the other hand, rigorous stability and robustness analysis can be derived for

Zhong-Ping Jiang, Tao Bian and Weinan Gao (2020), "Learning-Based Control: A Tutorial and Some Recent Results", Foundations and Trends® in Systems and Control: Vol. 8, No. 3, pp 176–284. DOI: 10.1561/2600000023.

the closed-loop system with real-time learning-based controllers. The effectiveness of the proposed learning-based control framework is demonstrated via its applications to theoretical optimal control problems tied to various important classes of continuous-time dynamical systems and practical problems arising from biological motor control, connected and autonomous vehicles.

The version of record is available at: http://dx.doi.org/10.1561/2600000023

1

Introduction

The idea of learning-based control can be traced back at least to the Ph.D. dissertation (Minsky, 1954), where Minsky for the first time introduced the concept of reinforcement learning (RL) motivated by the problem of gaining further insight into the learning, memorizing, and thinking processes in human brain. Borrowing the words from Sutton *et al.* (1992), RL is direct adaptive optimal control. The field of RL is vibrant and is far from being saturated as clearly shown in numerous review articles and books (Bertsekas, 2011, 2013; Schmidhuber, 2015; Silver, 2015; Sutton and Barto, 2018; Szepesvári, 2010). Sixty years later after Minsky's original work, Google DeepMind developed perhaps one of the most advanced artificial intelligence (AI) system based on RL, and defeated the human world champion in the game of Go (Silver *et al.*, 2016, 2017). Indeed, besides Google DeepMind's AI system, RL has demonstrated its advantage in multiple industry applications (Barto *et al.*, 2017; Lorica, 2017). The recent success of RL and related methods can be attributed to several key factors. First, RL is driven by reward signals obtained through the interaction with the environment. Different from other machine learning (ML) techniques, this learning architecture is especially useful when the learning objective is to find the optimal

behavior or policy over a time interval. Second, RL is closely related to the human learning behavior. It has been identified in a number of papers that the learning behavior in the frontal cortex and the basal ganglia is driven by the neuron spikes in dopamine neurons. These spikes encode the temporal difference error signal (Dayan and Balleine, 2002; Doya, 2002; Glimcher, 2011; Lo and Wang, 2006; Wang *et al.*, 2018; Wise, 2004), which is a key element in the RL theory (Sutton and Barto, 2018, Chapter 6). Hence, it is not surprising that we can achieve human-level intelligence through RL. Third, RL has a solid mathematical foundation. The main theoretical result behind RL is the dynamic programming (DP) theory (Bellman, 1957), which is a powerful tool for solving sequential decision making problems. The mathematical guarantee from DP theory gives the advantage of RL over other heuristic AI methods. Finally, RL can be incorporated with other ML and optimization methods to build a sophisticated learning system. For example, the learning performance of RL methods can be significantly improved by incorporating the recently developed deep neural network technique (Mnih *et al.*, 2015, 2016; Schmidhuber, 2015; Silver *et al.*, 2016, 2017). Because of these important features, RL and its extensions have become one of the most active research topics in AI and ML communities. Nonetheless, conventional RL theory exhibits some shortcomings. A common feature of most RL algorithms is that they are only applicable for discrete environments described by Markov decision processes (MDP) or discrete-time systems. To overcome this limitation, several researchers Baird, III (1993, 1994), Munos (2000), Doya (2000), Doya *et al.* (2002), van Hasselt and Wiering (2007), Theodorou *et al.* (2010), and van Hasselt (2012), have made significant efforts in adapting RL into the continuous environment, by discretizing and interpolating the time-state-action spaces. Alternatively, Bradtke and Duff (1994), Sutton *et al.* (1999), and Das *et al.* (1999) investigated RL for the semi-Markov process, a continuous-time dynamical system equipped with discrete state space. It should be mentioned that these methods may suffer from high computational burden when performing the discretization and approximation for continuous-time dynamical systems evolving in continuous state and action spaces. More recently, RL-based

methods, mostly known under the name of adaptive dynamic programming (ADP), have been developed for continuous learning environments (Russell and Norvig, 2010, Chapter 2) described by ordinary differential equations (ODEs) or stochastic differential equations (SDEs). Another limitation of traditional RL methods is that the stability and robustness of the controlled process is usually not considered. In fact, a common assumption in the convergence analysis of various RL methods is that the underlying MDP always has a steady state distribution (Bhatnagar *et al.*, 2009; Nedić and Bertsekas, 2003; Sutton *et al.*, 2000; Tsitsiklis, 1994; Tsitsiklis and Van Roy, 1997). However, few results have been proposed to guarantee this assumption, especially when there exist policies under which the MDP does not have steady state distribution. In contrast with these limitations, experimental results have demonstrated that biological systems exhibit the ability of learning complicated motor movements in an unstable environment composed with high-dimensional continuous state space (Adams, 1971; Shadmehr and Mussa-Ivaldi, 2012; Wolpert *et al.*, 2011). Traditional RL theory is insufficient in explaining this type of learning process.

The purpose of this tutorial is to present a learning-based approach to control dynamical systems from real-time data and to review some major developments in this relatively young field. Due to space limitation, we will focus on continuous-time dynamical systems described by ODEs and SDEs. With input–output data at hand, we can certainly opt for the indirect route as in model-based control theory, that is, first build a mathematical model and then design controllers for the practical system in question. This indirect method has proven successful for a variety of problems arising in the contexts of engineering and sciences. However, it is widely known that building precise mathematical models that can describe the motion of dynamical systems is time-consuming and costly. For certain classes of optimal control problems, especially when the dynamical systems under consideration are strongly nonlinear, it is very hard, if not impossible, to solve the Bellman equation. This observation has led Bellman (1957) to state: "Turning to the succor of modern computing machines, let us renounce all analytic tools." In this monograph, we aim to develop a framework for learning-based control theory that shows how to learn directly suboptimal controllers

from input–output data. Ultimately, these suboptimal controllers are expected to converge to the (unknown) optimal solution to the Bellman equation. Besides the benefit of direct vs indirect control methods, the learning-based control theory overcomes the curse of modeling tied to the traditional DP. There are three main challenges on the development of learning-based control. First, there is a need to generalize existing recursive methods, known under the names of policy iteration (PI) and value iteration (VI), from model-based to data-driven contexts when the system dynamics are completely unknown. Previous RL-based learning algorithms are not directly extendable to the setting of continuous-time dynamical systems, let alone convergence and sensitivity analyses. Second, as a fundamental difference between learning-based control and RL, stability and robustness are important issues that must be addressed for the safety-critical engineering systems such as self-driving cars. Therefore, there is a need to develop new tools and methods, beyond the present literature of RL, that can provide theoretic guarantees on the stability and robustness of the controller learned from real-time data collected online along the trajectories of the control system under consideration. Third, data efficiency of RL algorithms need be addressed for safety-critical engineering systems. In this monograph, we will address the first two issues and only discuss the third issue from the perspective of numerical and experimental studies by means of some case studies. The learning-based control theory as reviewed in this monograph is closely tied to the literature of safe RL and ADP, and is a new direction in control theory that is still in its infancy and especially so for continuous-time dynamical systems described by differential equations. For prior work of others on ADP-based optimal control, the reader may consult (Jiang and Jiang, 2017; Lewis and Vrabie, 2009; Lewis *et al.*, 2012b; Liu *et al.*, 2017; Luo *et al.*, 2014; Song *et al.*, 2015; Vrabie *et al.*, 2013; Wang *et al.*, 2009; Werbos, 1968) and many references therein. For recent developments in learning-based control for other types of systems and problems, see Antsaklis *et al.* (1991), Antsaklis and Rahnama (2018), Rahnama and Antsaklis (2019), Werbos (2013, 2014, 2018), Kiumarsi *et al.* (2017), He and Zhong (2018), Recht (2019), Bertsekas (2019), Kamalapurkar *et al.* (2018), Chen *et al.* (2019), Pang *et al.* (2020), and references therein.

The rest of the monograph is organized as follows. Section 2 describes the learning-based optimal control of continuous-time linear and nonlinear systems described by (ordinary or stochastic) differential equations. Section 3 is concerned with the learning-based optimal control of a class of large-scale dynamical systems. Section 4 deals with the learning-based adaptive optimal tracking with disturbance rejection, the so-called adaptive optimal output regulation problem, for classes of linear and nonlinear control systems. Applications of the presented learning-based control theory to autonomous vehicles and human motor control are given in Section 5. Finally, some concluding remarks and discussions on future work are provided in Section 6.

2

Learning-Based Control of Continuous-Time Dynamical Systems

2.1 Uncertain Linear Time-Invariant Systems

The linear quadratic regulator (LQR for short) problem was for the first time solved by Kalman (1960), and has been perceived as a key design method in automatic control. For a review of earlier results on LQR theory, the reader should consult Willems (1971), Kučera (1973), and Anderson and Moore (1989), to name a few. For applications of LQR and its extensions in different fields of engineering and sciences, the reader should consult Bryson (1994), Todorov and Jordan (2002), Lewis *et al.* (2004), and Kolm *et al.* (2014).

2.1.1 The Linear Quadratic Regulator Problem

In this subsection, we review some key properties of LQR control that will be explored in the synthesis of our RL and ADP algorithms and controllers. Consider the following linear time-invariant system:

$$\dot{x} = Ax + Bu, \tag{2.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and input of the system, and A and B are constant real-valued matrices with appropriate dimensions.

The finite-horizon LQR problem with respect to (2.1) consists in finding a control law u to minimize the following cost functional:

$$\mathcal{J}_{\tau-t_0}(u;x_0) = x^T(\tau)Mx(\tau) + \int_{t_0}^{\tau} r(t)dt, \quad x(t_0) = x_0, \tag{2.2}$$

where $r = x^TQx + u^TRu$, $M = M^T \geq 0$, $Q = Q^T \geq 0$, and $R = R^T > 0$.

From a RL perspective, (2.1) and (2.2) represent the interaction between the agent and the environment throughout a continuous-time interval $[t_0, \tau]$. At each time instant $t \geq t_0$, $x(t)$ represents the environment's state, on the basis of which the agent selects a control action $u(t)$. Then, as a consequent of $(x(t), u(t))$, the agent receives a cost $r(t)dt$, and the environment state also transits from $x(t)$ to $x(t) + \dot{x}(t)dt$. $r(t)$ here is directly observed from the environment, and generally we do not know its mathematical representation. This process continues until the terminal time τ, when a terminal cost $x^T(\tau)Mx(\tau)$ is received. As in RL, the goal of the agent here is to find an optimal controller, denoted $u^*(t)$, $t \in [t_0, \tau]$, such that the cumulative cost $\mathcal{J}_{\tau-t_0}$ is minimized.

Following the standard argument in LQR theory (Kučera, 1973), $\mathcal{J}_{\tau-t_0}$ achieves its minimum value $\min_u \mathcal{J}_{\tau-t_0} = x_0^T P(t_0)x_0$ under the optimal controller $u^*(t) = -K(t)x(t)$, with P and K defined as

$$\begin{aligned} K &= R^{-1}B^TP, \\ -\dot{P} &= \mathrm{Ricc}(P), \quad P(\tau) = M, \end{aligned} \tag{2.3}$$

where

$$\mathrm{Ricc}(P) := A^TP + PA - PBR^{-1}B^TP + Q.$$

(2.3) is known as the *differential matrix Riccati equation* (DMRE), which is a backward matrix differential equation, in the sense that given the terminal/boundary condition $P(\tau) = M$, (2.3) admits a unique real symmetric and positive definite solution on $[t_0, \tau]$, which can be solved backward in time (Kučera, 1973).

By fixing $t_0 = 0$ and letting τ go to infinity in (2.2), we have the following infinite-horizon cost:

$$\mathcal{J}(u;x_0) = \int_0^{\infty} r(t)dt. \tag{2.4}$$

Note that the terminal cost is removed because to ensure (2.4) is finite, x must converge to the origin as time goes to infinity, in which case the terminal cost defined in (2.2) is always 0.

Obviously, (2.4) may not be bounded under some controllers. Here, we say a controller u is *admissible*, if system (2.1) under u is globally asymptotically stable at the origin, and $\mathcal{J}(u; x_0) < \infty$.

It is well known that under stabilizability and observability assumptions (Kalman, 1960), $\mathcal{J}$ is minimized under the optimal controller $u^* = -K^*x$, where

$$K^* = R^{-1}B^T P^*, \tag{2.5}$$

$$0 = \mathrm{Ricc}(P^*). \tag{2.6}$$

(2.6) is called *algebraic Riccati equation* (ARE), implying both P^* and K^* are constant matrices. As we have expected, for any real symmetric and positive semidefinite $P(0)$, (2.3) is asymptotically stable at P^*, backward in time (Kučera, 1973; Shayman, 1986; Willems, 1971):

$$\lim_{t \to -\infty} P(t) = P^*.$$

In this monograph, we mainly focus on discussing ADP and RL methods in the context of infinite-horizon optimal control. For the infinite-horizon case, we can carry out the learning process continuously along a single learning path, because of the fact that the interaction between the agent and the environment does not stop naturally at some prescribed terminal time. Alternatively, we can also break the learning process into episodes. After finishing one learning episode, the system is reset to an initial state, from where the next learning episode starts. Both of these two setups are explored in the learning algorithms in the following sections.

2.1.2 Policy Iteration

In Kleinman (1968), Kleinman developed the continuous-time policy iteration algorithm to approximate P^*. Instead of solving the ARE (2.6) directly, Kleinman's algorithm aims at deriving a sequence $\{P_k\}_{k=0}^{\infty}$ from the following Lyapunov equation iteratively

$$0 = \mathrm{G}(K_k, P_k), \quad k = 0, 1, \ldots,$$

Algorithm 2.1 PI for continuous-time LQR

Initialize: Choose an arbitrary admissible control gain matrix K_0. Let $k \leftarrow 0$.

repeat

Policy Evaluation: Given K_k, solve for P_k from

$$0 = \mathrm{G}(K_k, P_k). \tag{2.7}$$

Policy Improvement: Update

$$K_{k+1} \leftarrow R^{-1}B^T P_k. \tag{2.8}$$

Let $k \leftarrow k+1$.

until $|P_k - P_{k+1}| < \epsilon$ (a small positive number)

where

$$\mathrm{G}(K,P) := (A-BK)^T P + P(A-BK) + K^T RK + Q,$$
$$K_{k+1} := R^{-1}B^T P_k.$$

Kleinman's algorithm starts from an admissible control gain matrix K_0, i.e., $A-BK_0$ is a Hurwitz matrix. Once P_k is solved, the cost associated with the controller $u_k = -K_k x$ is readily given in a closed form: $V_k(x) = x^T P_k x$. On top of P_k, the control gain matrix is then updated from K_k to K_{k+1}. It can be shown that P_k converges monotonically to P^*, in the sense that $x^T P_k x \geq x^T P_{k+1} x \geq \cdots$ for any $x \in \mathbb{R}^n$. The complete policy iteration algorithm is given in Algorithm 2.1. Different from (2.6), (2.7) is a linear matrix equation of P_k in each iteration, and hence it is much easier to solve for high-dimensional problems. In addition, as a Lyapunov equation, (2.7) implies that K_k is admissible.

A key observation here is that the PI is equivalent to the Newton-Raphson method (Dahlquist and Björck, 1973) in matrix space. Indeed, it is easily checked using (2.7) and (2.8) that

$$0=(A-BK_{k+1})^T(P_{k+1}-P_k)+(P_{k+1}-P_k)(A-BK_{k+1})+\mathrm{Ricc}(P_k),$$

which takes the same form as (6.3.1) in Dahlquist and Björck (1973). Hence, as a standard convergence result for Newton-Raphson method, P_k in the PI algorithm converges quadratically to the optimal solution P^*.

On top of Kleinman's seminal work, several extensions of Algorithm 2.1 have been proposed to solve AREs and LQR problems; see Sandell (1974), Banks and Ito (1991), Benner and Byers (1998), Benner *et al.* (2008), and Lanzon *et al.* (2008), to name a few.

In 2012, Jiang and Jiang (2012a) proposed the first continuous-time PI-based *off-policy* ADP method. Different from the traditional on-policy methods (Jiang and Jiang, 2011; Modares and Lewis, 2014; Murray *et al.*, 2002; Vamvoudakis, 2017; Vrabie *et al.*, 2009, 2013), the behavior policy that is used to generate data is not necessarily the same as the target policy to be evaluated in off-policy learning. A key idea in developing the off-policy method for continuous-time dynamical systems is to estimate the following *Hamiltonian* rather than P_k and K_k separately:

$$H_k(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} A^T P_k + P_k A + Q & P_k B \\ B^T P_k & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}.$$

Introducing the Hamiltonian provides at least two benefits. First, the two PI equations (2.7) and (2.8) in Algorithm 2.1 can be directly rewritten in terms of H_k as

$$0 = H_k(x,u_k) \quad \text{and} \quad u_{k+1} = \arg\inf_a H_k(x,a),$$

for all x, respectively. Note that matrices A, B, Q and R no longer appear explicitly in above equations. In addition, by taking the time derivative along the trajectories of system (2.1), we have

$$\frac{d}{dt}V_k(x) + r = H_k(x,u). \tag{2.9}$$

It is interesting to note that the left-hand side of Equation (2.9) is a continuous-time version of the temporal-difference (TD) error (Sutton and Barto, 2018). Indeed, since the left-hand side of Equation (2.9) involves a differential term, as opposed to the TD error, a more appropriate name should be the temporal-differential error. More importantly, this equation holds for any input u, as long as the system solution is well defined. This provides the possibility of developing off-policy learning.

Using the above two important features of H_k, an off-policy PI-based ADP algorithm is given in Algorithm 2.2. In this algorithm,

Algorithm 2.2 PI-based ADP for continuous-time LQR

Initialize: Choose a linear admissible controller u_0. Let $k \leftarrow 0$.
for each episode **do**
Collect data (x, u) and running cost r from the environment.
repeat
Given u_k, solve (V_k, H_k) from

$$0 = \int_{t_j}^{t_{j+1}} H_k(x, u_k)dt, \tag{2.10}$$

$$V_k(x)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} \left(H_k(x, u) - r\right) dt, \tag{2.11}$$

where $j = 0, 1, \ldots, J$.
Update $u_{k+1} \leftarrow \arg\inf_a H_k(x, a)$. Let $k \leftarrow k + 1$.
until $|V_k(x) - V_{k+1}(x)| < \epsilon|x|^2$, or hits the maximum iteration number in one episode
end for

(2.10) is deduced from the policy evaluation step (2.7) in Algorithm 2.1, and (2.11) is due to the continuous-time TD error. In particular, if we subtract (2.10) from (2.11) directly, the resulting equation reduces to the original ADP algorithm proposed by Jiang and Jiang (2012a).

Suppose the set of linear equations in the form of (2.10) and (2.11) are not degenerate for a sufficiently large J. Then we can solve V_k and H_k uniquely from (2.10) and (2.11) via standard least squares technique; see Jiang and Jiang (2012a) for detailed derivation. Once H_k is obtained, u_{k+1} is readily obtained. Note that the knowledge on A, B, Q and R is not required throughout the learning process. Despite the similarity of notations, u_k is completely different from u. The actual system input u is used to generate the online data, and hence corresponds to the behavior policy. On the other hand, u_k (including u_0) is the controller updated through the policy improvement, and hence is in fact the target policy. That being said, u_k is a function of x, while u is just a control input (function of time) that theoretically can even be an open-loop state-independent signal. This separation of u from u_k classifies

Algorithm 2.2 together with its extensions (see sections below) as an off-policy method.

To ensure the non-degeneration assumption on (2.10) and (2.11) is satisfied, we can inject an exploration noise, such as sinusoidal waves or harmonic signals, into the input u while generating the online data. Since Algorithm 2.2 is an off-policy method, injecting the exploration noise into the control input does not compromise the estimation accuracy of the target policy being learned, as long as the persistent excitation (PE) condition (Åström and Wittenmark, 1997; Tao, 2003) is satisfied. In fact, matrix P_k in the definition of V_k that is solved from (2.10) and (2.11) is exactly the same matrix in Algorithm 2.1. The technique of injecting an exploration noise has been widely used in RL to improve learning performance (Thrun, 1992) and adaptive control to guarantee the PE condition. As we shall see in the rest of this monograph, the non-degenerate assumption on the linear equations for a sufficiently large J will be repeatedly required, although under various formats, in different ADP and RL algorithms to ensure the convergence of these algorithms. In this monograph, we always use the term "PE condition" to denote this type of assumptions.

2.1.3 Value Iteration

In this subsection, we introduce the continuous-time VI algorithm that leads to the development of new RL and ADP methods for solving continuous-time LQR problems. Compared with PI, employing the VI has at least two advantages: (1) the knowledge of an initial admissible controller is not required to start the iteration; and (2) there is no need to solve the policy evaluation equation in each iteration. Due to these two advantages, VI has become the most widely used and best understood algorithm for solving discounted Markov decision problems (Puterman, 2005). Furthermore, VI methods for discrete-time, continuous-state-action space systems can also be found in Lancaster and Rodman (1995, Section 17.5) and Bertsekas (2005, Proposition 4.4.1), for the setting of linear systems; and in Bertsekas (2017), for a nonlinear extension.

To understand the VI in the continuous-time setting, we start with the continuous-time VI algorithm in integration form. For any real

symmetric and positive semidefinite matrix M, define the continuous-time DP mapping $\mathcal{T}_\tau$ as

$$x_0^T \mathcal{T}_\tau(M) x_0 = \inf_u \left\{ x^T(\tau) M x(\tau) + \int_0^\tau (x^T Q x + u^T R u) dt \right\}. \quad (2.12)$$

System (2.1) together with (2.12) formulates a finite-horizon linear quadratic optimization problem. Hence, from Section 2.1.1 we have $\mathcal{T}_\tau(M) = P(0)$, where P is the solution to (2.3) on $[0, \tau]$ with boundary condition $P(\tau) = M$. It is not difficult to see that the definition of $\mathcal{T}_\tau$ matches the definition of DP mapping for MDPs in Bertsekas (2007, Eq. (1.4)). In particular, the fixed-point equation and the monotonicity property still hold:

$$P^* = \mathcal{T}_\tau(P^*), \quad P_1 \geq P_2 \Rightarrow \mathcal{T}_\tau(P_1) \geq \mathcal{T}_\tau(P_2).$$

Once the DP operator $\mathcal{T}_\tau$ is defined, the continuous-time VI in integral form is naturally obtained:

$$P_{k+1} = \mathcal{T}_\tau(P_k), \quad k = 0, 1, \ldots. \quad (2.13)$$

Based on the asymptotic stability property of the DMRE, the convergence of P_k to P^* is easily obtained.

It is not difficult to see the similarity between (2.13) and the VI algorithm for discrete-time linear systems (Bertsekas, 2005, Proposition 4.4.1) and MDPs (Puterman, 2005, Theorem 6.3.1). By letting τ goes to 0, we develop the continuous-time VI in stochastic approximation form in Algorithm 2.3.

Obviously, (2.14) is just a discrete-time approximation of (2.3) (after reverting the time line) with step size h_k. Finding a suitable h_k is not an easy task in practice. If h_k is too small at the beginning of learning, then the convergence tends to be very slow and could even terminate before reaching the optimal value. On the other hand, if h_k is too large, then the algorithm may simply diverge. Because h_k is decreasing to 0, the convergence speed of Algorithm 2.3 is sub-linear, just as in other stochastic algorithms including the famous Robbins–Monro algorithm (Robbins and Monro, 1951). Nevertheless, since there is no need to solve a matrix equation in each iteration as in Algorithm 2.1, the computational complexity for each step in Algorithm 2.3 is not

Algorithm 2.3 VI for continuous-time LQR

Initialize: Choose $P_0 = P_0^T > 0$. $k \leftarrow 0$. $h_k > 0$, $\sum_{k=0}^{\infty} h_k = \infty$, $\lim_{k\to\infty} h_k = 0$.

repeat

Given P_k, update P_{k+1} from

$$\begin{aligned} P_{k+\frac{1}{2}} &\leftarrow P_k + h_k \,\mathrm{Ricc}(P_k), \\ P_{k+1} &\leftarrow P_{k+\frac{1}{2}} + Z_k. \end{aligned} \tag{2.14}$$

Let $k \leftarrow k+1$.

until $|P_k - P_{k+1}| < \epsilon$

high, and the VI algorithm is still an efficient method to solve the LQR control problem.

To ensure (2.14) does not diverge due to the large step size h_k at the early stage of learning, a projection term Z_k is introduced in Algorithm 2.3, with the following definition:

$$\begin{cases} Z_k = P_0 - P_{k+\frac{1}{2}}, & \text{increase q to q+1,} \quad \text{if } P_{k+\frac{1}{2}} \notin B_q, \\ Z_k = 0, & \text{otherwise,} \end{cases}$$

where $\{B_q\}_{q=0}^{\infty}$ is a sequence of bounded sets such that $B_0 \subseteq B_1 \subseteq \cdots \subseteq B_q \subseteq B_{q+1} \subseteq \cdots$, and B_∞ is the space of all real symmetric and positive semidefinite matrices. Note that the definition of Z_k is independent of the system dynamics. Z_k is designed to project $P_{k+\frac{1}{2}}$ back to P_0, whenever $P_{k+\frac{1}{2}}$ becomes non-positive definite or $|P_{k+\frac{1}{2}}|$ becomes unrealistically large. Different from PI, P_k in VI does not have the monotone convergence property. Hence, Z_k is used as a safeguard to control the approximation error caused by the large step size h_k during initial learning iterations. In addition, if we know P^* is in a known bounded set, then we can simplify the above design of Z_k by fixing B_q as this bounded set.

Different from Algorithm 2.1, instead of starting from an initial admissible controller, Algorithm 2.3 starts from an arbitrary positive definite real symmetric matrix representing the initial guess on the optimal value function.

Algorithm 2.4 VI-based ADP for continuous-time LQR

Initialize: Choose $V_0(x) = x^T P_0 x$, $P_0 = P_0^T > 0$. $k \leftarrow 0$.
for each episode **do**
Collect data (x, u) and running cost r from the environment.
repeat
Given V_k, solve H_k from

$$V_k(x)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_k(x,u) - r)\, dt, \tag{2.15}$$

where $j = 0, 1, \ldots, J$.
Update

$$V_{k+1}(x) \leftarrow V_k(x) + h_k \inf_a H_k(x,a) + x^T Z_k x, \quad \forall x \in \mathbb{R}^n. \tag{2.16}$$

Let $k \leftarrow k + 1$.
until $|V_k(x) - V_{k+1}(x)| < \epsilon |x|^2$, or hits the maximum iteration number in one episode
end for

The convergence of Algorithm 2.3 is guaranteed by the asymptotic stability property of the DMRE; see Bian and Jiang (2016b,c, 2018) for more details. In addition, we have shown in Bian and Jiang (2019a) that Algorithm 2.3 is robust to external disturbances with a linear L^2 gain (van der Schaft, 2017). More importantly, this L^2 gain can be assigned to be arbitrarily small, by tuning Q and R matrices in the running cost. This important property allows us to apply the VI method in environments with various types of noises and disturbances, with guaranteed convergence result.

The stochastic approximation setup allows us to develop continuous-time VI-based RL and ADP methods. On the basis of Algorithm 2.3, a VI-based ADP algorithm is given in Algorithm 2.4. Similar to Algorithm 2.3, Algorithm 2.4 starts from an arbitrary positive definite real symmetric matrix. This dramatically increases the wide usability of Algorithm 2.4 in practice.

In addition, Algorithm 2.4 is an off-policy method. Each iteration of Algorithm 2.4 contains two steps, aiming at updating H_k and V_k, respectively. In the first step, given V_k, the Hamiltonian H_k is solved from (2.15). As in Algorithm 2.2, by injecting the exploration noise in the input u, the PE condition can be satisfied in the sense that the set of linear equations (2.15) are not degenerate for a large enough J. It is worth noting that, in this case, (2.15) in fact defines a linear mapping from V_k to H_k that is independent of the learning iterations. As a result, we can directly map V_k to H_k through this mapping, without solving the linear equation (2.15) repeatedly in each iteration. In the second step, once H_k is obtained, V_{k+1} is directly obtained from (2.16), which is just an alternative representation of (2.14) based on H_k. The convergence analysis of V_k to V^* has been given in Bian and Jiang (2016b).

Finally, it is interesting to note that Algorithm 2.4 shares some similarity with TD learning. Indeed, when h_k is sufficiently small, we easily see from Algorithm 2.4 that for any $t \geq 0$ and $x(t)$,

$$\begin{aligned} & h_k \inf_u H_k(x(t), u) \\ & \quad = h_k \inf_u \left\{ (Ax + Bu)^T P_k + P_k(Ax + Bu) + x^T Q x + u^T R u \right\} \\ & \quad \approx \inf_u \left\{ V_k(x)|_t^{t+h_k} + \int_t^{t+h_k} r ds \right\}, \end{aligned} \tag{2.17}$$

which is consistent with the definition of TD(0) algorithm in RL (Sutton and Barto, 2018).

Besides the papers mentioned above in this section, other off-policy ADP papers on different types of optimal control problems for continuous-time linear systems have been given in Jiang and Jiang (2013a,b, 2014b, 2017), Bian *et al.* (2014, 2015, 2016), Gao and Jiang (2016a), Gao *et al.* (2018), Kiumarsi *et al.* (2017), Vamvoudakis (2017), Vamvoudakis and Ferraz (2018), Rizvi and Lin (2019), Yang *et al.* (2019), Chen *et al.* (2019), Pang *et al.* (2019, 2020), and Pang and Jiang (2020), to name a few. Generalizations of PI and VI based learning control results to nonlinear systems can be found in Vrabie *et al.* (2013), Liu *et al.* (2017), and Jiang and Jiang (2017) and many references therein.

2.2 Uncertain Linear Stochastic Systems

The methods reviewed in the previous section are designed for learning in the noise-free environment. Obviously this assumption is too restrictive, since in most real-world systems, stochastic noise is not only unavoidable, but plays a critical role in the learning system (Section 5.1.4).

Generally speaking, learning in stochastic environments is a difficult task, as the learning algorithm is required to be robust to stochastic noises (Bian and Jiang, 2019a) to some degree, so that the convergence to the optimal solution can be guaranteed.

In this section, we review some preliminary results on RL and ADP methods for continuous stochastic environments. In addition, we also point out some key difficulties that need to be addressed when developing stochastic ADP methods in continuous time.

2.2.1 Continuous-Time Stochastic Optimal Control

Suppose w is a (q_1+q_2)-dimensional standard Brownian motion. Denote $\mathcal{F}_t$ as the σ-field generated by $w(s)$, $0 \le s \le t$. Consider the following linear stochastic system with additive and multiplicative noises:

$$dx = (Ax + Bu)dt + \sum_{i=1}^{q_1}(A_i x + B_i u)dw_i + \sum_{i=1}^{q_2}\sigma_i dw_i, \tag{2.18}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are $\{\mathcal{F}_t\}$-adapted random processes representing the state and the input of the system, respectively; $x(0) = x_0$ is deterministic; A, B, A_i and B_i are constant real matrices; and σ_i are constant real vectors.

Borrowing the definition in Haussmann (1971), we say a controller u is admissible, if it is Lipschitz continuous, and an invariant probability measure π exists such that $\mathbb{E}_\pi[|x|^2] < \infty$.

In the setting of stochastic optimal control, several types of cost functionals can be considered, each of which has its own advantages and usage in practice. In this section, we discuss three cost functionals defined on the infinite horizon. All of these costs are associated with the same running cost $x^TQx + u^TRu$, where Q and R are real symmetric and positive definite matrices.

Ergodic Control

The objective of ergodic control is to minimize (with probability one)

$$\rho(u) = \limsup_{T\to\infty} \frac{1}{T}\int_0^T r(t)dt,$$

or to minimize

$$\bar{\rho}(u) = \limsup_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}_0[r(t)]dt,$$

where $r = x^TQx + u^TRu$, $\mathbb{E}_0$ is the expectation conditional on $\mathcal{F}_0$.

It has been shown by Wonham (1967), Kleinman (1969), and Haussmann (1971) that if $\sum_{i=1}^{q_2}\sigma_i\sigma_i^T > 0$, then there exists an ergodic stationary probability measure π on $\mathbb{R}^n \times \mathbb{R}^m$ for system (2.18) under an admissible linear controller $u = -Kx$. Kleinman (1969) has shown that in this case $\bar{\rho}(u)$ and $\rho(u)$ are equal to $\mathbb{E}_\pi[r] = \sum_{i=1}^{q_2}\sigma_i^TP\sigma_i$ (with probability one in the later case), where P is the solution to the following Lyapunov equation for stochastic systems:

$$0 = \mathrm{G}(K, P),$$

where

$$\begin{aligned}\mathrm{G}(K,P) := {} & (A-BK)^TP + P(A-BK) + Q \\ & + \sum_{i=1}^{q_1} A_i^TPA_i + K^T\left(\sum_{i=1}^{q_1} B_i^TPB_i + R\right)K.\end{aligned}$$

In particular, the *ergodic controller* that minimizes ρ and $\bar{\rho}$ is $u^* = -K^*x$, and $\bar{\rho}(u^*) = \rho(u^*) = \sum_{i=1}^{q_2}\sigma_i^TP^*\sigma_i$, with K^* and P^* defined as

$$0 = \mathrm{Ricc}(P^*), \tag{2.19}$$

$$K^* = \left(\sum_{i=1}^{q_1} B_i^TP^*B_i + R\right)^{-1}\left(P^*B + \sum_{i=1}^{q_1} A_i^TP^*B_i\right)^T, \tag{2.20}$$

where

$$\begin{aligned}\mathrm{Ricc}(P) := {} & A^TP + PA + Q \\ & + \sum_{i=1}^{q_1} A_i^TPA_i - (K^*)^T\left(\sum_{i=1}^{q_1} B_i^TPB_i + R\right)K^*.\end{aligned}$$

Note that under certain assumptions, the non-degeneration assumption $\sum_{i=1}^{q_2} \sigma_i \sigma_i^T > 0$ in the above ergodic control problem can be relaxed (Haussmann, 1971; Zakai, 1969).

The ergodic control is one of the most widely studied stochastic optimal control problems. For instance, the ergodic controller characterizes the long-term behavior of the underlying processes, and hence is especially useful in developing Monte-Carlo samplers and RL algorithms. In addition, the ergodic property is closely related to some important properties (e.g., aperiodicity and recurrence) of stochastic processes (Meyn and Tweedie, 1993a,b). For more details on the ergodic control-related methods for continuous-time models, see Wonham (1967), Khas'minskii (1967), and Haussmann (1971, 1973), for linear diffusion processes, and Meyn and Tweedie (1993a,b), Borkar (2006), Arapostathis *et al.* (2012), and Khas'minskii (2012) for more general stochastic processes.

However, in many applications, especially when the noise is degenerated, the ergodic control costs ρ and $\bar{\rho}$ may not be good performance indicators, since the ergodic control problem cannot characterize the transient performance of the control system. Indeed, the values of $\bar{\rho}(u)$ and $\rho(u)$ are independent of the initial state x_0 for admissible u. In the extreme case where there is no stochastic noise in the system, all the stabilizing controllers for system (2.18) minimize the ergodic control costs, and $\inf_u \rho(u) = \inf_u \bar{\rho}(u) = 0$.

Discount-Optimal Control

Given $x_0 \in \mathbb{R}^n$, denote the *discounted cost* as

$$\mathcal{J}_\lambda(u; x_0) = \mathbb{E}_0 \left[\int_0^\infty e^{-\lambda t} r(t) dt \right], \quad \lambda > 0. \tag{2.21}$$

(2.21) is quite similar to the cost in deterministic LQR problem. The main difference here is that a discounting term $e^{-\lambda t}$ is introduced to add more weights on the running cost during the initial control period. Since the discounting term decreases to 0 as t goes to the infinity, the total cost value is finite under admissible controllers, even in the presence of additive noises. Different from ergodic control costs, the discounted cost also takes into account the initial state of the controlled system.

It has been shown in Kushner (1967, Theorem 4) that $\mathcal{J}_\lambda$ is minimized under the *discount-optimal* controller $u^* = -K^*x$, where K^* follows the same definition in (2.20), and P^* satisfies

$$\lambda P^* = \mathrm{Ricc}(P^*). \tag{2.22}$$

Moreover, the minimum cost is

$$\inf_u \mathcal{J}_\lambda(u;x) = x^T P^* x + \frac{1}{\lambda}\sum_{i=1}^{q_2} \sigma_i^T P^* \sigma_i.$$

Note that if we introduce a new matrix $A' = A - \frac{1}{2}\lambda I$ where I is an identity matrix with appropriate dimension, then (2.22) reduces to (2.19) in which A is replaced by A'. Hence, the analysis on (2.19) can be directly carried over to (2.22).

Different from the ergodic control problem, both the transient performance and the stationary distribution are taken into account in the discount-optimal control problem. As we can see, λ represents the tradeoff between the contributions of the initial state and the steady state distribution to the total cost.

However, introducing the discounting factor λ brings up the concern about stability guarantee for the closed-loop system. Because of the discounting term $e^{-\lambda t}$ in the discounted cost, (2.22) may still be finite even if the closed-loop system is not stable. Nevertheless, this stability challenge can be resolved by applying the PI approach with a carefully selected initial controller (Bian and Jiang, 2016a).

It should be mentioned that the discount-optimal control problem has been extensively studied in the context of MDPs and RL. In the continuous-time setting, the discounted cost has also attracted attention from many practitioners, since the discounting feature makes it a more realistic cost candidate in optimal controller design. Take the finance as an example. Participants in the finance field show more interests in the short-term profit/loss. In fact, according to the arbitrage pricing theory (Ross, 1976), it is necessary to include the discounting factor in portfolio optimization as it reflects the risk-free rate in the market. For more details on discount optimal control-related methods, see Kushner (1967), Pliska (1986), and Bensoussan and Frehse (1992), to name a few.

Bias-Optimal Control

An alternative way to bypass the stability issue in discount-optimal control is to consider the bias-optimal control.

Given $x_0 \in \mathbb{R}^n$, denote the *biased cost* as

$$\mathcal{J}_b(u; x_0) = \mathbb{E}_0 \left[\int_0^\infty (r - \bar{\rho}(u)) dt \right]. \tag{2.23}$$

Borrowing the definition in Mahadevan (1996) for MDPs, we call a controller u *bias-optimal*, if it is admissible and minimizes both $\bar{\rho}(u)$ and the biased cost $\mathcal{J}_b(u; x_0)$. Note from (2.23) that $\mathcal{J}_b$ represents the accumulated difference between r and $\bar{\rho}(u)$. If the initial state x_0 is a random variable following the steady state distribution associated with u, then $\mathcal{J}_b(u; x_0) = 0$ by the law of large numbers. In other words, $\mathcal{J}_b(u; x_0)$ characterizes the relative difference in the total cost gained from starting in state x_0 as opposed to some other states (Mahadevan, 1996). Due to this feature, the biased cost has been extensively studied for MDPs (Mahadevan, 1996; Puterman, 2005; Schwartz, 1993; Tsitsiklis and Van Roy, 1999, 2002; Yu and Bertsekas, 2009). An extension to non-degenerate diffusions can be found in Arapostathis *et al.* (2012, Chapter 3, 2014).

It has been shown in the literature (Arapostathis *et al.*, 2014; Bian and Jiang, 2016a) that the *bias-optimal* controller is $u^* = -K^* x$, where K^* is defined by (2.20) and (2.19). To see this, note that u^* minimizes $\bar{\rho}(u)$ by definition. Moreover, compared with other controllers which also minimize $\bar{\rho}(u)$, one can easily see that only u^* minimizes $\mathcal{J}_b$, by following standard DP arguments. Finally, the minimum cost is $\mathcal{J}_b(u^*; x_0) = x_0^T P^* x_0 - \mathbb{E}_\pi[x^T P^* x]$.

It is not difficult to see that the three stochastic optimal control problems discussed above share almost the same ARE. This is not a coincidence, as they are not independent from each other. To be more specific, one can show (Bian and Jiang, 2016a) that for any admissible $u = -Kx$,

$$\lim_{\lambda \to 0} \left| \mathcal{J}_\lambda(u; x) - \frac{1}{\lambda} \bar{\rho}(u) - \mathcal{J}_b(u; x) \right| = 0.$$

A similar result has been obtained for MDPs (Puterman, 2005, Corollary 8.2.4).

In the rest of this section, we focus on RL and ADP algorithms used to solve the ergodic control problem. Extensions to the other two stochastic optimal control problems are straightforward given that they have similar AREs.

2.2.2 Policy Iteration

The PI algorithm for stochastic LQR problems was first developed by Kleinman (1969) to solve an optimal stationary controller, and then was extended to more general problems by different authors (Damm, 2004; Damm and Hinrichsen, 2001). Although these papers did not consider the discounted cost, it is a trivial extension to apply existing results to solve the discount-optimal control problem.

The PI algorithm for solving the ergodic control problem follows the exact same structure as its deterministic counterpart in Algorithm 2.1, except that $\mathrm{G}(\cdot)$ and K_k follow different definitions; see Kleinman (1969), Damm and Hinrichsen (2001), Damm (2004), and Bian *et al.* (2016) for details.

To develop PI-based ADP to solve continuous-time stochastic LQR problems, we consider the following special case of (2.18):

$$\begin{aligned} dx &= Axdt + B(u+v)dt, \\ vdt &= \sum_{i=1}^{q_1}(A_i x + B_i u)dw_i + \sum_{i=1}^{q_2}\sigma_i dw_i. \end{aligned} \tag{2.24}$$

Note that all the stochastic noises are passed into the system through the input channel B in (2.24). Here v can be viewed as a bundle of white noises in the input channel. As a result, the running-cost from the environment in this case is defined as $r = x^T Qx + (u+v)^T R(u+v)$,

By following the same procedure as in the deterministic case, we introduce the following value function and Hamiltonian:

$$V_k(x) = x^T P_k x, \quad H_k(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} F_{k,11} & F_{k,12} \\ F_{k,12}^T & F_{k,22} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix},$$

Algorithm 2.5 PI-based ADP for continuous-time stochastic LQR

Initialize: Choose a linear admissible controller u_0. $k \leftarrow 0$.
for each episode **do**
Collect data (x, u, v) and running cost r from environment.
repeat
Given u_k, solve $(V_k, H_k, \bar{\rho}_k)$ from

$$0 = \int_{t_j}^{t_{j+1}} H_k(x, u_k)dt,$$
$$V_k(x(t))|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} \left(H_k(x, u + v) + \bar{\rho}_k - r\right) dt,$$

where $j = 0, \ldots, J$.
Update $u_{k+1} \leftarrow \arg\min_a H_k(x, a)$. Let $k \leftarrow k + 1$.
until $|V_k(x) - V_{k+1}(x)| < \epsilon|x|^2$, or hits the maximum iteration number in one episode
end for

where

$$F_{k,11} = A^T P_k + P_k A + \sum_{i=1}^{q_1} A_i^T P_k A_i + Q,$$
$$F_{k,12} = P_k B + \sum_{i=1}^{q_1} A_i^T P_k B_i, \quad F_{k,22} = \sum_{i=1}^{q_1} B_i^T P_k B_i + R.$$

The extra terms in H_k compared with the deterministic case are due to the presence of stochastic noise. It is not hard to check that H_k inherits all the properties of the Hamiltonian in the deterministic case. As a result, we can derive the PI-based stochastic ADP algorithm (Algorithm 2.5) by mimicking the same structure in Algorithm 2.2.

Compared with Algorithm 2.2, an extra term $\bar{\rho}_k$ is introduced in Algorithm 2.5. This term is the Itô's correction term caused by the additive noise in (2.24). One can show (Bian and Jiang, 2016a) that both P_k and $\bar{\rho}_k$ converge to P^* and $\bar{\rho}(u^*)$ in the ergodic control problem.

As in the deterministic case, we can solve V_k, H_k, and $\bar{\rho}_k$ simultaneously, provided that the PE condition is satisfied for a sufficiently large J. In fact, since the stochastic noise also serves as the exploration

Algorithm 2.6 VI-based ADP for continuous-time stochastic LQR

Initialize: Choose $V_0(x) = x^T P_0 x$, where $P_0 = P_0^T > 0$. $k \leftarrow 0$.
for each episode **do**
  Collect data (x, u, v) and running cost r from environment.
  repeat
    Given V_k, solve $(V_k, \bar{\rho}_k)$ from

$$V_k(x)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_k(x, u+v) + \bar{\rho}_k - r)\, dt,$$

    where $j = 0, \ldots, J$.
    Update

$$V_{k+1} \leftarrow V_k + h_k \inf_a H_k(x, a) + x^T Z_k x.$$

    Let $k \leftarrow k + 1$.
  until $|V_k(x) - V_{k+1}(x)| < \epsilon |x|^2$, or hits the maximum iteration number in one episode
end for

noise in this case, the PE condition is much easier to satisfy. Similar to Algorithm 2.2, Algorithm 2.5 is also an off-policy method. Once H_k is obtained, u_{k+1} is readily updated. The knowledge on system matrices is not required in this setting. For detailed convergence analysis for Algorithm 2.5, see Jiang and Jiang (2014a), Bian and Jiang (2016a), and Bian *et al.* (2016).

2.2.3 Value Iteration

The VI algorithm for solving the ergodic control problem can be developed in the same fashion as for its deterministic counterpart in Algorithm 2.3, except that $\mathrm{Ricc}(\cdot)$ is defined differently.

Now, under the assumption that the stochastic noise enters into the system through the input channel, we obtain the VI-based ADP algorithm in Algorithm 2.6 for system (2.24).

Compared with Algorithm 2.4, Algorithm 2.6 also contains an extra term due to the stochastic nature of the problem. Our analysis on Algorithm 2.4 can be carried over to Algorithm 2.6 with minor modifications. See Bian and Jiang (2016c, 2018) for detailed convergence analysis on Algorithm 2.6.

2.3 Uncertain Nonlinear Systems

Despite its wide-ranged applications in industry, LQR control has a restrictive requirement on the controlled system and the cost functional. Hence, it is usually not applicable in the situation where both the system model and the running cost take general nonlinear forms. In this section, we focus on developing RL and ADP methods for general nonlinear optimal control problems.

2.3.1 Optimal Control of General Nonlinear Systems

Consider a continuous-time nonlinear system taking the nonaffine form:

$$\dot{x} = f(x, u), \tag{2.25}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and f is locally Lipschitz and satisfies $f(0, 0) = 0$.

We aim at finding an optimal controller u^* to minimize the following infinite-horizon cost with respect to system (2.25):

$$\mathcal{J}(u; x_0) = \int_0^\infty r(x, u)dt, \tag{2.26}$$

where $x(0) = x_0$, and r is continuous and positive definite. Similar to the linear case, we say a controller u is admissible, if system (2.25) is asymptotically stable at the origin, and $\mathcal{J}(u; x_0)$ is finite. In addition, we say a feedback control policy μ admissible, if $u = \mu(x)$ is admissible.

Suppose at least one admissible controller exists. Then the minimum cost value exists, and we can denote the minimum value of $\mathcal{J}$ at x_0 as $V^*(x_0)$. By DP theory, if V^* and u^* are sufficiently smooth, then V^* satisfies the following Hamilton-Jacobi-Bellman (HJB) equation (Bellman, 1954) over $\mathbb{R}^n$:

$$0 = \inf_{a \in \mathbb{R}^m} \{\partial_x V^*(x) f(x, a) + r(x, a)\}, \quad V^*(0) = 0. \tag{2.27}$$

Moreover, $u^*(t) = \mu^*(x(t))$, where

$$\mu^*(x) = \arg \inf_{a \in \mathbb{R}^m} \{\partial_x V^*(x) f(x, a) + r(x, a)\}. \tag{2.28}$$

On the other hand, if (2.27) admits a solution V^*, with the infimum achieved at $a = \mu^*(x)$, then $V^*(x) = \inf_u \mathcal{J}(u; x)$, and u^* (or μ^*, resp.) is the corresponding optimal controller (or policy, resp.). In addition, if V^* is proper and u^* is continuous, then u^* is a stabilizing controller, and hence u^* is admissible.

For more details on DP and optimal control theory for continuous-time dynamical processes, see Bellman (1957), Fleming and Rishel (1975), Evans (2005), Liberzon (2012), Lewis *et al.* (2012a), and Clarke (2013), and references therein.

2.3.2 Policy Iteration

The idea of PI in continuous-time dynamical systems can be traced back to the early work of Leake and Liu (1967), under the name of successive approximation. The ingenious idea behind this paper is to define two mappings, one from the controller space to the value function space, and another one in the reverse direction. It can be shown that the composition of these two mappings forms a contraction, and its fixed point is the optimal value function. This naturally leads to a fixed-point iteration to solve the HJB equation. From the perspective of DP, the two mappings identified in Leake and Liu (1967) are actually the policy evaluation and policy improvement, respectively. However, the iterative method in Leake and Liu (1967) is not popular within the optimal control community, as it requires several technical assumptions that are hard to verify.

The first PI, successive approximation algorithm that was brought to the community's attention in solving nonlinear optimal control problems was developed in a series of papers by Saridis and his co-workers (Beard, 1995; Beard *et al.*, 1997; Saridis and Lee, 1979). Although these papers were developed only for affine nonlinear systems, i.e., the control input is assumed to appear linearly in the standard state model, they relaxed and improved several restrictive assumptions in Leake and Liu (1967).

Algorithm 2.7 PI for continuous-time nonlinear optimal control

Initialize: Choose an arbitrary admissible control policy μ_0 and compact set K. $k \leftarrow 0$

repeat

Policy Evaluation: Given μ_k, solve V_k from

$$0 = \partial_x V_k(x) f(x, \mu_k(x)) + r(x, \mu_k(x)), \quad V_k(0) = 0. \tag{2.29}$$

Policy Improvement: Update μ_{k+1} via

$$\mu_{k+1}(x) \leftarrow \arg\inf_a \{\partial_x V_k(x) f(x, a) + r(x, a)\}. \tag{2.30}$$

Let $k \leftarrow k + 1$

until $\|V_k - V_{k+1}\|_\infty < \epsilon$ on K

In Bian *et al.* (2014), we further extended the successive approximation method in Leake and Liu (1967) and Beard (1995) to general nonaffine nonlinear systems. The PI for general nonlinear systems is summarized in Algorithm 2.7. Obviously, solving (2.29) and (2.30) is much easier than solving (2.27), which is a nonlinear partial differential equation (PDE). Another observation is that V_k serves as a Lyapunov function in (2.29), implying μ_k is admissible. In addition, Kleinman's algorithm (Algorithm 2.1) becomes a special case of Algorithm 2.7. Moreover, we can rewrite (2.29) and (2.30) as

$$0 = (p_{k+1} - p_k) f(x, a') + p_k f(x, a') + r(x, a'),$$

where $p_k = \partial_x V_k(x)$ for all k, and $a' = \arg\inf_a \{p_k f(x, a) + r(x, a)\}$. If $f(x, a)$ and $r(x, a)$ are continuously differentiable in a, then similar to the PI for LQR problem, the above equation is a Newton-Raphson method that aims at solving a vector p from the nonlinear equation $0 = \inf_a \{p^T f(x, a) + r(x, a)\}$, pointwise in $\mathbb{R}^n$. Hence, the nonlinear PI algorithm is also expected to have quadratic convergence speed. For detailed convergence analysis of Algorithm 2.7, see Bian *et al.* (2014).

Unfortunately, the methods discussed above are plagued by some serious implementation issues in practical applications. First, since the value functions and control policies in PI (V_k and μ_k in Algorithm 2.7) could take general nonlinear forms, solving these functions could still be

computationally expensive, especially with high-dimensional state space. In addition, since the value functions and control policies converge to their optimal values pointwise, the stopping criteria in PI is in the supremum norm of $V_k - V_{k+1}$, and hence we must check the value of $V_k(x)$ for each x in K, which is impossible in practice. Nevertheless, these difficulties can be lifted by using various types of numerical approximations. For instance, Beard (1995) and Beard *et al.* (1997) adopted Galerkin's method in Saridis's successive approximation design to approximate the value function via a linear combination of basis functions. Here, we will discuss in details the approach of using the neural network (NN) approximation directly in the continuous-time setting. The main advantage of this approach is that it allows us to develop continuous-time RL and ADP methods directly with concrete convergence and stability analysis.

The first off-policy ADP methods for continuous-time nonlinear systems were developed in Jiang and Jiang (2014b) and Bian *et al.* (2014), by implementing the single-layer NN approximation in nonlinear PI algorithms. As in Section 2.1.2, before introducing these off-policy ADP methods, we first define the Hamiltonian in the setting of nonlinear systems:

$$H_k(x,u) = \partial_x V_k(x) f(x,u) + r(x,u), \quad \forall (x,u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

Following this definition, (2.29) and (2.30) can be rewritten as

$$0 = H_k(x, \mu_k(x)) \quad \text{and} \quad \mu_{k+1}(x) = \arg\inf_a H_k(x,a),$$

respectively. In addition, similar to the linear case (see Section 2.1.2), by taking the time derivative along the trajectories of system (2.25), we have the following continuous-time version of TD error for general nonlinear optimal control problems:

$$\dot{V}_k(x) + r(x,u) = H_k(x,u).$$

Once H_k is defined, we can introduce NNs to approximate the unknown functions in the above equations:

$$\hat{H}(w_k,x,u) \approx H_k(x,u), \quad \hat{V}(c_k,x) \approx V_k(x), \quad \hat{\mu}(\theta_k,x) \approx \mu_k(x).$$

Algorithm 2.8 PI-based ADP for continuous-time nonlinear optimal control problems

Initialize: Choose an arbitrary θ_0 such that $\hat{\mu}(\theta_0, \cdot)$ is admissible.
$k \leftarrow 0$.
for each episode **do**
Collect data (x, u) and running cost r from the environment.
repeat
Given θ_k, solve for w_k and c_k from

$$0 = \int_{t_j}^{t_{j+1}} \hat{H}(w_k, x, \hat{\mu}(\theta_k, x))dt, \tag{2.31}$$

$$\hat{V}(c_k, x)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (\hat{H}(w_k, x, u) - r)dt, \tag{2.32}$$

where $j = 0, \ldots, J$.
Update $\theta_{k+1} \leftarrow \arg\inf_\theta \hat{H}(w_k, x, \hat{\mu}(\theta, x))$. Let $k \leftarrow k + 1$.
until $|c_k - c_{k+1}| < \epsilon$, or hits the maximum iteration number in one episode.
end for

$\hat{H}$, $\hat{V}$, and $\hat{\mu}$ are three NNs uniquely determined by their NN weights (w_k, c_k, and θ_k). Then, instead of searching H_k, V_k, and μ_k from some infinite-dimensional functional spaces, we only need to find the NN weights that lead to the best approximators.

Summarizing the above, we present the off-policy PI-based ADP algorithm for general nonlinear systems in Algorithm 2.8.

A special choice of the NNs in our learning algorithm is the single layer network:

$$\hat{H}(w_k, x, u) = \sum_{i=1}^{N} w_{i,k}\psi_i(x, u), \quad \hat{V}(c_k, x) = \sum_{i=1}^{N} c_{i,k}\phi_i(x),$$
$$\hat{\mu}(\theta_k, x) = \sum_{i=1}^{N} \theta_{i,k}\varphi_i(x),$$

where ψ_i, ϕ_i, and φ_i represent the neurons in the network (Park and Sandberg, 1991), and N is the size of the network. In this case, (2.31) and (2.32) become a set of linear equations in w_k and c_k. Note that if

we subtract (2.31) from (2.32) directly, the resulting equation is exactly the ADP algorithm proposed by Bian *et al.* (2014) and Jiang and Jiang (2014b). If these linear equations are non-degenerate for a sufficiently large J, then we can solve w_k and c_k uniquely from (2.31) and (2.32) using linear regression. As in the ADP design for LQR control, this non-degeneration assumption can be satisfied by injecting the exploration noise into the system input.

Besides the different problem formulation, a key difference between Algorithms 2.2 and 2.8 is that the least square solution is usually different from the true solution of the policy evaluation in Algorithm 2.2. This difference is caused by the NN approximation errors, which do not appear in the ADP methods for LQR problems. Because of this approximation error, the learning algorithm converges to the optimal solution only over a pre-selected compact set in supremum norm, when both J and N go to the infinity; see Jiang and Jiang (2014b) and Bian *et al.* (2014) for detailed convergence analysis.

2.3.3 Value Iteration

In Section 2.1.3, we have reviewed the VI for continuous-time LQR control problems. In this subsection, following the same steps in Section 2.1.3, we develop VI and its online implementation for continuous-time nonlinear optimal control problems.

We start with the definition of continuous-time DP mapping. For any positive semidefinite function V defined on $\mathbb{R}^n$, we can define the continuous-time DP mapping $\mathcal{T}_\tau$ as

$$\mathcal{T}_\tau(V)(x_0) = \inf_u \left\{ V(x(\tau)) + \int_0^\tau r(x,u)dt \right\}, \quad x(0) = x_0. \tag{2.33}$$

The right-hand side of (2.33) is a finite-horizon optimal control problem. Then, we can connect V and $\mathcal{T}_\tau(V)$ through the following finite-horizon HJB equation:

$$-\partial_t v(x,t) = \inf_{a \in \mathbb{R}^m} \{\partial_x v(x,t) f(x,a) + r(x,a)\}, \tag{2.34}$$

for all $(x,t) \in \mathbb{R}^n \times [0,\tau]$. Given the boundary condition $v(x,\tau) = V(x)$, one can easily derive $\mathcal{T}_\tau(V)(x) = v(x,0)$, provided that (2.34) admits a

classical solution. Moreover, the minimizer in (2.33) is defined as

$$u(t) = \mu(x,t) = \arg \inf_{a \in \mathbb{R}^m} \{\partial_x v(x,t) f(x,a) + r(x,a)\}.$$

Since f and r are time homogeneous, the finite-horizon cost is stationary, and hence we can safely shift the time interval $[0,\tau]$ in (2.33) and (2.34) to any interval of length τ, without altering the definition of $\mathcal{T}_\tau(V)$ and μ.

Once the DP operator $\mathcal{T}_\tau$ is defined, the continuous-time VI in integral form is naturally obtained as

$$V_{k+1} = \mathcal{T}_\tau(V_k), \quad k = 0, 1, \ldots.$$

On the basis of the above updating equation, we can show (Bian and Jiang, 2016c) that the solution to (2.34) converges backward in time to the infinite-horizon optimal value function V^* uniformly on any compact set:

$$\lim_{t \to -\infty} v(\cdot, t) = V^*(\cdot),$$

given that the boundary condition V is positive semidefinite and proper, and (2.27) and (2.34) are well-defined.

Given the convergence of v, we can follow the steps in Section 2.3.2 to represent (2.34) and the TD error in terms of the Hamiltonian:

$$\partial_s v(x,s) = \inf_{a \in \mathbb{R}^m} H_s(x,a), \quad \dot{v}_s(x) + r(x,u) = H_s(x,u). \tag{2.35}$$

In the above equations, x is the trajectory of system (2.25) generated by u. To avoid confusion, here we use t and s to represent the actual system time and the time evolution of the HJB equation (2.34), respectively. In addition, we reversed the time in (2.34), so that v is evolving forward in s in our learning algorithm.

Based on the above formulation, an off-policy VI-based ADP algorithm for continuous-time nonlinear optimal control problems is introduced in Algorithm 2.9. Similar to Algorithm 2.8, we introduce NNs ($\hat{v}$ and $\hat{H}$) in Algorithm 2.9 to approximate the unknown functions in (2.35). Moreover, in order to cope the iterative updating scheme of NN weights, the stochastic approximation is used in Algorithm 2.9 to approximate the updating equation of $\hat{v}$. Similar to (2.31) and (2.32), if

Algorithm 2.9 VI-based ADP for continuous-time nonlinear optimal control problems

Initialize: Choose an arbitrary c_0 such that $\hat{v}(c_0, \cdot)$ is positive semidefinite and proper. $k \leftarrow 0$.

for each episode **do**

Collect data (x, u) and running cost r from the environment.

repeat

Given c_k, solve for w_k from

$$\hat{v}(c_k, x)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (\hat{H}(w_k, x, u) - r)dt, \tag{2.36}$$

where $j = 0, \ldots, J$.

Update c_{k+1} via

$$\hat{v}(c_{k+1}, x) = \hat{v}(c_k, x) + h_k \inf_{a \in \mathbb{R}^m} \hat{H}(w_k, x, a). \tag{2.37}$$

until $|c_k - c_{k+1}| < \epsilon$, or hits the maximum iteration number in one episode.

end for

we use single layer NNs in (2.36) and (2.37), then w_k and c_{k+1} can be solved via linear regression in each iteration, provided the PE condition is satisfied for a sufficiently large J.

Because of the NN approximation errors, the convergence result holds over a compact set as in Algorithm 2.8. Detailed convergence proof of Algorithm 2.9 is given in Bian and Jiang (2016c).

Finally, interested readers are referred to Jiang and Jiang (2015b), Gao and Jiang (2016b), Kiumarsi *et al.* (2017), Liu *et al.* (2017), Gao and Jiang (2018), and Pang and Jiang (2020) for additional papers on off-policy continuous-time nonlinear ADP methods.

2.3.4 Numerical Stability of PI- and VI-Based Algorithms

The algorithms reviewed in this monograph so far are mainly discussed under ideal implementation scenarios. In real-world applications, one must consider the errors induced from the numerical calculations, such

as the numerical integration and numerical matrix inverse, when implementing these algorithms. In fact, the NN approximation error we mentioned previously may also be an alternative source of numerical errors.

When discussing the convergence of a numerical method, two criteria, i.e., the consistency and the stability, must be considered. Consistency here means that, in each learning iteration, we can make the residual error arbitrarily small, by increasing the precision of numerical calculation and NN approximation. This condition is relatively easy to check, and indeed all the ADP algorithms discussed before satisfy this condition. Stability, on the other hand, is a much more serious issue. Roughly speaking, a numerical algorithm is stable, if it does not magnify the errors (due to truncation, round-off, etc.) during the numerical process. From a control-theoretic point of view, this is equivalent to saying the algorithm is to some extent robust to the numerical errors.

We have shown in Bian and Jiang (2019a) that the VI algorithm for the LQR control problem indeed shows a promising robustness property to different types of disturbances in the learning algorithm. In addition, one can even tune the input-to-state stable (ISS) gain (Sontag, 2008) of the VI algorithm, by selecting the weighting matrices in the cost properly. Besides the numerical error, we see in Section 2.2.3 that VI is also robust to stochastic noise. In the setting of nonlinear optimal control, we can see from (2.34) that if the numerical error induced from one iteration is sufficiently small, it can be dominated by the running cost r. Then, the ADP algorithm with numerical errors can be considered as an algorithm learning in an ideal scenario but under a different cost. As a result, the ADP algorithm will converge to a suboptimal solution, and the sub-optimality is characterized by the time integration of the numerical error.

PI algorithms are essentially Newton-Raphson methods, and hence should also inherit the numerical stability property from standard Newton-Raphson method. However, it is not surprising that they may not be robust to stochastic noise (Spall, 2003, Chapter 1), especially when the value function corresponding to the initial controller is far from the optimal solution. Nevertheless, it is possible to address this issue via stochastic approximation (Kushner and Yin, 2003; Spall, 2003).

3

Learning-Based Control of Large-Scale Interconnected Systems

The RL and ADP methods reviewed so far are mainly designed for learning and control problems concerned with a centralized system. In modern engineering and natural systems, we often face large-scale interconnected systems that form a complex network with involved interactions. Examples include robotic networks, smart grids with thousands of distributed generators and an internet of connected, automated and human-driven vehicles. Finance and financial engineering industries provide other challenging examples such as the high frequency trading in an electronic market, where orders are managed by a limit order book (LOB). Every time an order is changed in the LOB, traders in the market will observe this change and take their corresponding actions. In this sense, all the traders in the market form a trading network through the LOB. As a result, it is of paramount theoretical and practical importance to investigate how the previously developed learning algorithms are scalable to decentralized dynamical networks. Instead of modeling the whole network as a centralized environment directly, decentralized RL and ADP methods conduct online learning for each subsystem in the network. The main difficulty in studying decentralized learning algorithms is that the behavior of a single system

or agent in the network may have direct or indirect influences to other systems or agents, which in turn pass these impacts back to this agent through loops in the network. As a result, besides the state and reward information received from the environment, each agent in the network also receives a dynamic disturbance from its neighbors. Such disturbance could be the power fluctuation from other DG units in the power grid, or the market impact of other traders in an electronic market. This leads to at least two basic questions that need to be answered while considering the decentralized RL and ADP. First, the stability and optimality of a dynamical network cannot be reduced down to checking the stability and optimality of individual systems. That is, even if each agent by itself is stable or made stable by feedback, there is no guarantee that the entire network with strong couplings is stable. To ensure the stability of the connected network, while still retaining good performance for each individual agent, a robust optimal control strategy is desirable. More specifically, we will model the interaction between two agents as dynamic uncertainty, which can be handled using the *small-gain* theory (Jiang and Liu, 2018; Jiang *et al.*, 1994). The second question is how to design a learning algorithm to achieve the robust optimality without knowing the model information. Different from traditional RL and ADP methods, the network interconnection must be considered in the decentralized learning algorithm.

3.1 Input-to-State Stability and Small-Gain Techniques

Before introducing the decentralized RL and ADP methods, we briefly review Sontag's concept of input-to-state stability (ISS) (Sontag, 2008) and an ISS small-gain theorem (Jiang *et al.*, 1996), that will serve as design ingredients in learning-based controller synthesis.

Consider the following system

$$\dot{x} = f(x, \Delta), \tag{3.1}$$

where f is a locally Lipschitz function vanishing at the origin, and Δ is an external disturbance input to the system. System (3.1) is said to be ISS, if the solutions $x(t)$ satisfy

$$|x(t)| \leq \max\{\beta(|x(0)|, t), \gamma_f(\|\Delta\|_\infty)\}, \tag{3.2}$$

where β is a $\mathcal{KL}$ function and γ_f is a $\mathcal{K}_\infty$ function.[1] Roughly speaking, (3.2) says that when $\Delta \equiv 0$, system (3.1) is globally asymptotically stable at the origin, and its transient behavior and rate of convergence are characterized by the $\mathcal{KL}$ function β. When Δ is bounded, then the solution to system (3.1) stays in a neighborhood of the origin that is characterized by the nonlinear gain function γ_f and the $\mathcal{L}^\infty$-norm of Δ. Hence, the ISS gain function γ_f serves as a quantifier of the robustness of (3.1) to the external disturbance.

Now, let us consider a special case where the disturbance Δ in (3.1) is an output from another ISS system:

$$\dot{\zeta} = g(\zeta, x), \quad \Delta := \Delta(x, \zeta), \tag{3.3}$$

where g and Δ are locally Lipschitz functions vanishing at the origin. The nonlinear system (3.3) is referred to as the *dynamic uncertainty*. In the literature of nonlinear control theory, the dynamic uncertainty can represent a large class of uncertainties, including the mismatch between the nominal model and the real plant when the order of the nominal model is lower than the order of the real plant. Similar to (3.1), we can also define a gain γ_g from the input x to the output Δ, called input-to-output stability (IOS) gain, to quantify the robustness of (3.3) to the external input x.

Note that even if both (3.1) and (3.3) are ISS, the interconnected system is not guaranteed to be stable. In Jiang *et al.* (1994), it is shown that, under the small-gain condition $\gamma_f \circ \gamma_g < Id$, the interconnected system remains IOS, and under certain observability condition, is ISS. For more details on the ISS and small-gain theory, see Sontag (2008), Karafyllis and Jiang (2011), Liu *et al.* (2014), and Jiang and Liu (2018) and numerous references therein.

3.2 Robust Optimal Control for Large-Scale Systems

In this subsection, we review some recent developments in robust optimal control theory for a continuous-time large-scale network composed of

[1] A class $\mathcal{K}$ function is a continuous mapping from $\mathbb{R}_+$ to $\mathbb{R}_+$ that is non-decreasing and vanishes at the origin. A class $\mathcal{K}_\infty$ function is an unbounded $\mathcal{K}$ function. A class $\mathcal{KL}$ function is a mapping from $\mathbb{R}_+ \times \mathbb{R}_+$ to $\mathbb{R}_+$ that is of class $\mathcal{K}$ in the first argument and converges to 0 in the second argument.

N systems:

$$\dot{\zeta}_i = g_i(\zeta_i, x_i), \tag{3.4}$$

$$\dot{x}_i = A_i x_i + B_i(z_i + \Delta_{1i}(\zeta_i, y)), \tag{3.5}$$

$$\dot{z}_i = F_i z_i + G_i(u_i + \Delta_{2i}(\zeta_i, z_i, y)), \tag{3.6}$$

$$y_i = x_i, \quad i = 1, 2, \ldots, N, \tag{3.7}$$

where g, Δ_{1i}, and Δ_{2i} are locally Lipschitz functions vanishing at the origin; (A_i, B_i) is stabilizable and G_i has full rank. $y = (y_1, y_2, \ldots, y_N)$ denotes the outputs of all subsystems in the network. For each ith subsystem, Δ_{1i} and Δ_{2i} represent the combined disturbances from its neighbors and the dynamic uncertainty (3.4).

When the network is decoupled (i.e., $\Delta_{1i} = \Delta_{2i} \equiv 0$), one way to stabilize system (3.5) and (3.6) is to use the backstepping technique (Krstić *et al.*, 1995) by breaking the controller design procedure into two steps. In the first step, we aim at designing a virtual controller z_i^* under which system (3.5) is ISS from the mismatch $z_i - z_i^*$ to x_i. In the second step, we design the real controller u that forces $z_i - z_i^*$ go to zero. As a consequence, system (3.5) and (3.6) is transformed to a cascade connection of an ISS system and a globally asymptotically stable (GAS)-system, and thus is GAS at the origin.

Obviously, the choice of z_i^* and u is not unique. To retain good transient performance of the controlled system, each network node solves a cascaded optimal control problem composed with the following two costs while conducting the backstepping design:

$$\mathcal{J}_{1,i}(x_i; z_i) = \int_0^\infty (x_i^T Q_{1,i} x_i + z_i^T R_{1,i} z_i) dt, \tag{3.8}$$

$$\mathcal{J}_{2,i}(z_i, z_i^*; u_i) = \int_0^\infty ((z_i - z_i^*)^T Q_{2,i}(z_i - z_i^*) + u_i^T R_{2,i} u_i) dt, \tag{3.9}$$

where $Q_{1,i}$, $Q_{2,i}$, $R_{1,i}$, and $R_{2,i}$ are symmetric positive definite matrices with appropriate dimensions. In the absence of Δ_{1i}, system (3.5) and cost (3.8) form a standard LQR control problem, and the optimal virtual controller z_i^* can be derived as a linear function in x_i. Once z_i^* is obtained, (3.9) becomes a quadratic cost as (3.8). Denoting $\tilde{z}_i = z_i - z_i^*$, we can derive from (3.6) that

$$\dot{\tilde{z}}_i = -\partial_x z_i^*(x_i)(A_i x_i + B_i(z_i + \Delta_{1i})) + F_i z_i + G_i(u_i + \Delta_{2i}). \tag{3.10}$$

Without considering the interconnections, system (3.10) and cost (3.9) again form an LQR control problem, from which we can solve the optimal controller u_i^*.

As we have argued before, finding an optimal controller is not enough in large-scale networks. Even if each node in the network is made stable, the interconnection between these nodes may cause instability in the network. To handle this problem, we will adopt *robust optimal* controllers. A controller u_i is called robust optimal if it is optimal for the costs (3.8) and (3.9) and the nominal system (3.5) and (3.6) in the absence of Δ_{1i} and Δ_{2i}, and is robustly stabilizing in the presence of these disturbances.

The main challenge facing robust optimal controller design is to guarantee the stability of the large-scale network. The small-gain theory provides a powerful tool to solve this problem simply by analyzing the composition of the gains along each simple cycle in the network. One way to fulfill the small-gain condition is to assign an appropriate gain, if possible, by means of feedback, for the subsystem (3.5)–(3.6). This is widely known as the gain assignment technique (Jiang and Mareels, 1997; Jiang *et al.*, 1994; Praly and Wang, 1996). As we have shown in Jiang and Jiang (2012b, 2013a) and Bian *et al.* (2015), the ISS gain of system (3.5) and (3.6) under the optimal controller depends on the cost matrices in (3.8) and (3.9). Hence, if the cost functionals are properly chosen, the small-gain condition can be fulfilled, therefore guaranteeing the stability of the large-scale network.

Finally, note that system (3.4)–(3.7) can be extended to higher-order systems where a repeated use of backstepping is required.

3.3 Decentralized Learning-Based Controllers for Large-Scale Systems

Recently, some preliminary results have been proposed on the design of robust optimal control policies via robust adaptive dynamic programming (for short RADP) algorithms (Bian and Jiang, 2018; Bian *et al.*, 2015; Jiang and Jiang 2012a, 2013a,b, 2014b, 2017). The idea of RADP was originally introduced by Jiang and Jiang (2011). Different from the traditional ADP methods, RADP addresses the presence of dynamic

uncertainty in linear and nonlinear systems. By employing the RADP algorithm, neither the system dynamics nor the system order need to be known exactly. Robust optimal control laws are learned directly from real-time input/output data along the trajectories of the control system.

3.3.1 PI-Based RADP for Continuous-Time Large-Scale Systems

The PI-based off-policy RADP algorithm is given in Algorithm 3.1. Since the robust optimal control problem discussed in Section 3.2 involves two cascaded costs, it is natural to use a two-phase design in Algorithm 3.1.

The first phase aims at solving z_i^* with respect to system (3.5) and cost (3.8). Following similar analysis in Section 2.1.2, we introduce the value function $V_{1,i,k}$ and the Hamiltonian $H_{1,i,k}$ for each subsystem i:

$$V_{1,i,k}(x) = x^T P_{1,i,k} x,$$
$$H_{1,i,k}(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} A_i^T P_{1,i,k} + P_{1,i,k} A_i + Q_{1,i} & P_{1,i,k} B_i \\ B_i^T P_{1,i,k} & R_{1,i} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}.$$

However, different from (2.11), the disturbance term $\Delta_{1,i}$ appears in (3.12). In addition, the running cost in (3.12) is defined as

$$r_{1,i} = x_i^T Q_{1,i} x_i + (z_i + \Delta_{1,i})^T R_{1,i} (z_i + \Delta_{1,i})$$

since the disturbed input $z_i + \Delta_{1,i}$ is fed to the environment. Once $H_{1,i,k}$ is obtained, the virtual controller $z_{i,k}^*$ can be directly updated.

The second phase of the robust optimal control design aims at solving u_i^* with respect to the error system (3.10) and cost (3.9), with z_i^* approximated by $\hat{z}_i^*$. Different from the first phase, the RADP design is more complicated in this phase, as the online data from both (3.5) and (3.6) are involved in the learning process. As a result, the value function $V_{2,i,k}$ and the Hamiltonian $H_{2,i,k}$ in the second learning phase are defined as

$$V_{2,i,k}(z) = z^T P_{2,i,k} z,$$
$$H_{2,i,k}(z,u,x,\Delta) = \begin{bmatrix} z \\ u \\ x \\ \Delta \end{bmatrix}^T \begin{bmatrix} M_{i,k,11} & M_{i,k,12} & M_{i,k,13} & M_{i,k,14} \\ M_{i,k,12}^T & M_{i,k,22} & 0 & 0 \\ M_{i,k,13}^T & 0 & 0 & 0 \\ M_{i,k,14}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ u \\ x \\ \Delta \end{bmatrix},$$

Algorithm 3.1 PI-based RADP for continuous-time large-scale systems

Initialize: Choose an admissible linear virtual controller $z_{i,0}^*$. Let $k \leftarrow 0$.

For each agent i, collect data $(x_i, z_i + \Delta_{1,i}, z_i, u_i + \Delta_{2,i})$, and running costs $r_{1,i}$ and $r_{2,i}$ from environment.

repeat

Given $z_{i,k}^*$, solve $(V_{1,i,k}, H_{1,i,k})$ from

$$0 = \int_{t_j}^{t_{j+1}} H_{1,i,k}(x_i, z_{i,k}^*)dt, \tag{3.11}$$

$$V_{1,i,k}(x_i)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_{1,i,k}(x_i, z_{i,\Delta}) - r_{1,i})dt, \tag{3.12}$$

where $z_{i,\Delta} = z_i + \Delta_{1,i}$ and $j = 0, \ldots, J$.

Update $z_{i,k+1}^* \leftarrow \arg\inf_z H_{1,i,k}(x_i, z)$. Let $k \leftarrow k + 1$.

until $|V_{i,k}(x) - V_{i,k+1}(x)| < \epsilon|x|^2$.

Choose an admissible linear controller $u_{i,0}$. Let $\hat{z}_i^* \leftarrow z_{i,k+1}^*$, $k \leftarrow 0$.

repeat

Given $u_{i,k}$, solve $(V_{2,i,k}, H_{2,i,k})$ from

$$0 = \int_{t_j}^{t_{j+1}} H_{2,i,k}(\tilde{z}_i, u_{i,k}, 0, 0)\, dt, \tag{3.13}$$

$$V_{2,i,k}(\tilde{z}_i)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_{2,i,k}(\tilde{z}_i, u_{i,\Delta}, x_i, z_{i,\Delta}) - r_{2,i})\, dt, \tag{3.14}$$

where $\tilde{z}_i = z_i - \hat{z}_i^*$, $u_{i,\Delta} = u_i + \Delta_{2,i}$, and $j = 0, \ldots, J$.

Update $u_{i,k+1} \leftarrow \arg\inf_u H_{2,i,k}(\tilde{z}_i, u, 0, 0)$. Let $k \leftarrow k + 1$.

until $|V_{2,i,k}(z) - V_{2,i,k+1}(z)| < \epsilon|z|^2$.

where

$$M_{i,k,11} = F_i^T P_{2,i,k} + P_{2,i,k} F_i + Q_{2,i}, \quad M_{i,k,12} = P_{2,i,k} G_i,$$
$$M_{i,k,13} = P_{2,i,k}(K_i^T A_i - F_i K_i), \quad M_{i,k,14} = P_{2,i,k} K_i^T B_i, \quad M_{i,k,22} = R_{2,i}.$$

$M_{i,k,13}$ and $M_{i,k,14}$ in the above definition are cross product terms due to the interconnection between (3.5) and (3.6). However, they do not appear in (3.13) which is derived from the policy evaluation. Similar to

the first phase, the perturbed running cost in (3.14) is

$$r_{2,i} = (z_i - \hat{z}_i^*)^T Q_{2,i}(z_i - \hat{z}_i^*) + (u_i + \Delta_{2,i})^T R_{2,i}(u_i + \Delta_{2,i}).$$

Still, the second learning phase follows the same structure as in the first phase.

Again, if the PE condition is satisfied for a sufficiently large J, then the convergence of Algorithm 3.1 is guaranteed (Bian *et al.*, 2015; Jiang and Jiang, 2012b, 2013a). In fact, the disturbances $\Delta_{1,i}$ and $\Delta_{2,i}$ automatically serve as the exploration noise during the RADP learning. As a result, it is much easier to satisfy the PE condition, which leads to the convergence to the robust optimal solution.

3.3.2 VI-Based RADP for Continuous-Time Large-Scale Systems

The VI-based RADP scheme can also be developed in the two phase manner as in the above subsection. Alternatively, given that the VI algorithm is inherently robust in the sense of ISS, we can simultaneously update the two learning processes with respect to x_i and z_i subsystems together (Bian and Jiang, 2019b), since a cascaded ISS system is still ISS and hence retains the convergence property. The VI-based off-policy RADP algorithm is summarized in Algorithm 3.2, where the step sizes $h_{1,i,k}$, $h_{2,i,k}$, and the projection terms $Z_{1,k}$ and $Z_{2,k}$ follow the same definitions in Algorithm 2.4.

The two Hamiltonians $H_{1,i,k}$ and $H_{2,i,k}$ in Algorithm 3.2 share the same structure of the Hamiltonians in Section 3.3.1. In addition, the online data $z_{i,\Delta}$, $u_{i,\Delta}$, the disturbed running costs $r_{1,i}$, $r_{2,i}$, and $z_{i,k}^*$ all have the same definitions as the ones in Algorithm 3.1, except that $\hat{z}_i^*$ in $r_{2,i}$ is now replaced by $z_{i,k}^*$. It is worth mentioning that Algorithm 3.2 inherits all the advantages of VI-based ADP, including the less restrictive initial condition.

From Algorithms 3.1 and 3.2, we observe that the "strict-feedback" structure in the network plays a crucial role in the RADP design. This special structure has been used explicitly for recursive learning and also been exploited to assign arbitrarily the ISS gain of each subsystem. The latter ensures the cyclic-small-gain conditions for the robustness of stability of the closed-loop large-scale system.

Algorithm 3.2 VI-based RADP for continuous-time large-scale systems

Initialize: Choose $V_{1,i,0}(x) = x^T P_{1,i,0} x$ and $V_{2,i,0}(z) = z^T P_{2,i,0} z$, where $P_{1,i,0} = P_{1,i,0}^T > 0$ and $P_{2,i,0} = P_{2,i,0}^T > 0$ for all i. $k \leftarrow 0$.
For each agent i, collect data $(x_i, z_i + \Delta_{1,i}, z_i, u_i + \Delta_{2,i})$, and running costs $r_{1,i}$ and $r_{2,i}$ from environment.
repeat
Given $V_{1,i,k}$ and $V_{2,i,k}$, solve $H_{1,i,k}$ and $H_{2,i,k}$ from

$$V_{1,i,k}(x_i)|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_{1,i,k}(x_i, z_{i,\Delta}) - r_{1,i})dt,$$
$$V_{2,i,k}(\tilde{z}_{i,k})|_{t_j}^{t_{j+1}} = \int_{t_j}^{t_{j+1}} (H_{2,i,k}(\tilde{z}_{i,k}, u_{i,\Delta}, x_i, z_{i,\Delta}) - r_{2,i})dt,$$

where $\tilde{z}_{i,k} = z_i - z_{i,k}^*$, $j = 0, \ldots, J$.
Update

$$V_{1,i,k+1}(x) \leftarrow V_{1,i,k}(x) + h_{1,i,k} \inf_z H_{1,i,k}(x, z) + x^T Z_{1,k} x, \quad \forall x,$$
$$V_{2,i,k+1}(z) \leftarrow V_{2,i,k}(z) + h_{2,i,k} \inf_u H_{2,i,k}(z, u, 0, 0) + z^T Z_{2,k} z, \quad \forall z.$$

Let $k \leftarrow k + 1$.
until $|V_{1,i,k}(x) - V_{1,i,k+1}(x)| < \epsilon |x|^2$ and $|V_{2,i,k}(z) - V_{2,i,k+1}(z)| < \epsilon |z|^2$.

3.3.3 Extensions to Stochastic and Nonlinear Systems

When the stochastic noise is accessible, it is not difficult to combine the above RADP algorithms with the stochastic ADP algorithms developed in Section 2.2. Stochastic noises can still be properly handled during the backstepping process to guarantee the robust stability of the control system (Bian and Jiang, 2017; Bian *et al.*, 2016). Resulting stochastic RADP algorithms still follow the structure of Algorithms 3.1 and 3.2, with additional terms due to the stochastic noises. Interested readers can refer to Jiang and Jiang (2015a), Bian *et al.* (2016), and Bian and Jiang (2016a, 2018) for more details on various types of RADP algorithms in stochastic environment.

Extending RADP to nonlinear optimal control problems is more difficult. Since existing ADP and RL methods for continuous-time nonlinear systems are mainly based on NN approximation, the system stability can only be guaranteed over a predefined compact set, i.e., only semi-global stability is achieved. Analyzing the robustness of the nonlinear system and assigning the ISS gain in this case are much more difficult. As a result, only a few preliminary results have been developed to study this problem (Jiang and Jiang, 2014b).

Robustness of stability is an important research topic that deserves more attention in the interdisciplinary field of learning-based control. Besides the papers discussed above, the interested reader should consult Wang *et al.* (2017), Jiang and Jiang (2017), Wang and Mu (2019), Gao *et al.* (2019b), and the references therein.

4

Learning-Based Output Regulation

The output regulation problem concerns designing a feedback controller to achieve nonvanishing disturbance rejection and asymptotic tracking for dynamical control systems, in which both the disturbance and reference signals are generated by a class of autonomous systems, named exosystems; see Francis (1977), Isidori and Byrnes (1990), Marino and Tomei (2003), Huang (2004), Su and Huang (2012), Wang *et al.* (2010), and Ding (2013). There are two major approaches to addressing a typical output regulation problem: feedforward-feedback and internal model principle. By means of the internal model principle (Francis and Wonham, 1976; Huang and Chen, 2004), one can convert an output regulation problem into a stabilization problem of an augmented system composed of the plant and a dynamic compensator named as internal model.

Considering parametric uncertainty in the system model, some adaptive control algorithms were proposed for dealing with output regulation problems (Ding, 2006; Liu *et al.*, 2009; Serrani *et al.*, 2001). However, a general characteristic of these algorithms is that the designed controllers are not optimal. The optimal output regulation problem was first studied for linear systems with matched disturbances in

Johnson (1971). In Krener (1992), Krener advanced the solutions to nonlinear model-based optimal output regulation problems, by means of a feedforward-feedback controller. The feedforward control input relies on the solution of certain nonlinear regulator equations and the optimal feedback control input relies on the solution of certain Hamilton-Jacobi-Bellman (HJB) equation. This is a model-based approach as the optimal controller requires the accurate knowledge of the controlled plant and relies on the analytical solution of the HJB equation which is by no means easy, if not impossible, for nonlinear systems. For these reasons, solving adaptive optimal output regulation problems for uncertain linear and nonlinear systems is an extremely challenging, yet important, control task.

Over the last decade, a trend in adaptive optimal control is to invoke RL (Sutton and Barto, 2018) and ADP (Jiang and Jiang, 2017; Lewis and Vrabie, 2009) for feedback control of dynamical systems. While data-driven stabilization has been a focus in the early developments of ADP for dynamical systems, the extension to tracking control via ADP has quickly attracted attention of several researchers. For instance, Kamalapurkar *et al.* (2015) proposed a model-based approximate optimal tracking control approach to guarantee the ultimately bounded tracking of nonlinear systems. Ni *et al.* (2013) developed an adaptive tracker by ADP for nonlinear affine systems with the input function being an identity function. Adaptive optimal trackers have also been developed by introducing discounting factors to the infinite-horizon cost functions; see Modares and Lewis (2014) and Luo *et al.* (2016). However, a straightforward application of these ADP approaches to adaptive optimal output regulation problems is not promising, due to either assuming the a prior knowledge of system model, or the difficulty in ensuring the asymptotic tracking of reference signals and the perfect rejection of nonvanishing disturbances.

This section presents a novel RL and ADP framework for the adaptive optimal output regulation of linear, nonlinear and multi-agent systems, which integrates the ADP and output regulation theory; see Figure 4.1. Based on the traditional output regulation problem setting, an exosystem is introduced that generates both references and disturbances for the system/environment. The agent needs to interact with

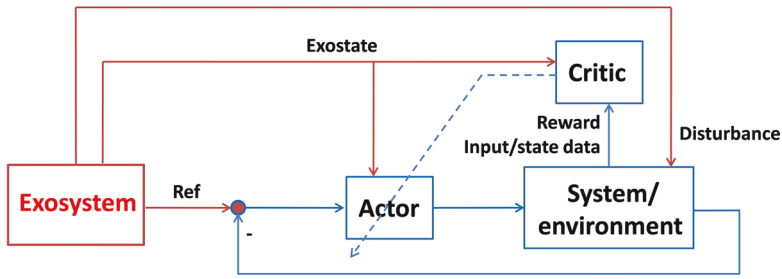


Figure 4.1: Configuration of an ADP-based control system for adaptive optimal output regulation.

both system and exosystem to seek the optimal control policy and the corresponding value function, and simultaneously achieve output regulation. Different from existing ADP for stabilization and tracking control problems, the value function is evaluated in the *critic* by not only reward, input and state data, but also the exostate. Based on the evaluated value function, the *actor* is able to update the control policy that is a function of both state and exostate. Notably, the adaptive optimal controller designed in this framework not only ensures asymptotic tracking of the closed-loop system, but also effectively asymptotically rejects nonvanishing disturbances. Moreover, the whole controller design process does not rely on the knowledge or identification of the system dynamics.

4.1 Uncertain Linear Systems

In this section, we aim to solve the problem of adaptive optimal tracking with disturbance rejection for continuous-time linear systems. We approach this task by taking advantage of techniques from two separately studied areas: ADP and output regulation theory. A data-driven learning-based algorithm is proposed on the basis of input and partial-state data.

4.1.1 Problem Formulation

Consider a class of continuous-time linear systems described by

$$\dot{x} = Ax + Bu + Dv, \tag{4.1}$$

$$\dot{v} = Ev, \tag{4.2}$$
$$e = Cx + Ju + Fv, \tag{4.3}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the control input, and $v \in \mathbb{R}^q$ the state of the exosystem (4.2). $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{r\times n}$, $D \in \mathbb{R}^{n\times q}$, $E \in \mathbb{R}^{q\times q}$, $F \in \mathbb{R}^{r\times q}$, and $J \in \mathbb{R}^{r\times m}$ are constant matrices. $d = Dv$ represents the exogenous disturbance, $y = Cx + Ju$ the output of the plant, $y_d = -Fv$ the reference signal and $e \in \mathbb{R}^r$ the tracking error. It is assumed that the measurement of v is unavailable to the designer.

Several assumptions are made on the system (4.1)–(4.3).

Assumption 4.1. The minimal polynomial of E is known and takes the form

$$\alpha_m(s) = \prod_{i=1}^{M}(s - \lambda_i)^{a_i} \prod_{j=1}^{N}(s^2 - 2\mu_j s + \mu_j^2 + \omega_j^2)^{b_j}, \tag{4.4}$$

with degree $q_m \leq q$, where a_i and b_j are positive integers and $\lambda_i, \mu_j, \omega_j \in \mathbb{R}$ for $i = 1, 2, \ldots, M, j = 1, 2, \ldots, N$.

Assumption 4.2. (A, B) is stabilizable.

Assumption 4.3. $\operatorname{rank}\begin{bmatrix} A - \lambda I & B \\ C & J \end{bmatrix} = n + r$, $\forall \lambda \in \sigma(E)$.

Under Assumption 4.1, we can always find a vector $w \in \mathbb{R}^{q_m}$ and a matrix $\hat{E} \in \mathbb{R}^{q_m \times q_m}$ such that

$$\dot{w}(t) = \hat{E}w(t), \tag{4.5}$$
$$v(t) = Gw(t), \quad \forall t \geq 0, \tag{4.6}$$

with $G \in \mathbb{R}^{q\times q_m}$ an unknown constant matrix.

Therefore, Equations (4.1) and (4.3) are equivalent to

$$\dot{x} = Ax + Bu + \hat{D}w, \tag{4.7}$$
$$e = Cx + Ju + \hat{F}w, \tag{4.8}$$

where $\hat{D} = DG$ and $\hat{F} = FG$.

Two output regulation problems are formulated as follows.

Definition 4.1. Linear Output Regulation Problem (LORP): Design a controller

$$u = -Kx + Lw \tag{4.9}$$

such that (1) the closed-loop system is exponentially stable with $\sigma(A - BK) \subset \mathbb{C}^-$. (2) the tracking error $e(t)$ asymptotically converges to 0, where $K \in \mathbb{R}^{m\times n}$ is a feedback control gain matrix and $L \in \mathbb{R}^{m\times q_m}$ is a feedforward control gain matrix.

Definition 4.2. Linear Optimal Output Regulation Problem (LOORP): Design a controller (4.9) to solve the LORP. Moreover, (4.9) is optimal with respect to some predefined cost.

To begin with, let us review a sufficient condition on the solvability of LORP.

Theorem 4.1 (Francis, 1977)**.** Under Assumptions 4.1 and 4.2, choose a K such that $\sigma(A - BK) \subset \mathbb{C}^-$. The LORP is solvable by the controller (4.9) if there exist $X \in \mathbb{R}^{n\times q_m}, U \in \mathbb{R}^{m\times q_m}$ solving the following regulator equations:

$$X\hat{E} = AX + BU + \hat{D}, \tag{4.10}$$

$$0 = CX + JU + \hat{F}, \tag{4.11}$$

with

$$L = U + KX. \tag{4.12}$$

Under the controller (4.9) that solves the LORP, for any initial state $x(0)$ and $w(0)$, one can satisfy $\lim_{t\to\infty} u(t) - Uw(t) = 0$ and $\lim_{t\to\infty} x(t) - Xw(t) = 0$.

It should be mentioned that the LOORP includes both asymptotic tracking and transient performance of the linear control system in question. To this end, we solve the static optimization Problem 1 to find the optimal solution (X^*, U^*) to regulator equations (4.10)–(4.11) and the dynamic optimization Problem 2 to find the optimal feedback control policy; see Krener (1992).

Problem 4.1.

$$\begin{aligned} \min_{(X,U)} \quad & \mathrm{Tr}(X^T\bar{Q}X + U^T\bar{R}U) \\ \text{subject to} \quad & (4.10)\text{–}(4.11), \end{aligned} \tag{4.13}$$

where $\bar{Q} = \bar{Q}^T > 0$ and $\bar{R} = \bar{R}^T > 0$.

Letting $\bar{x} = x - X^*w$ and $\bar{u} = u - U^*w$, it is direct to obtain the following error system

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}, \tag{4.14}$$

$$e = C\bar{x} + J\bar{u}. \tag{4.15}$$

The optimal feedback controller $\bar{u} = -K^*\bar{x}$ is found by solving the following constrained minimization problem.

Problem 4.2.

$$\begin{aligned} \min_{\bar{u}} \quad & \int_0^\infty (\bar{x}^T Q\bar{x} + \bar{u}^T R\bar{u})dt \\ \text{subject to} \quad & (4.14) \end{aligned}$$

where $Q = Q^T \geq 0, R = R^T > 0$, with $(A, \sqrt{Q})$ observable.

Therefore, when the system parameters are known, the LOORP is solved if we design a controller $u = -K^*x + L^*w$ where:

(1) K^* is computed by solving Problem 4.2.

(2) $L^* = U^* + K^*X^*$, where (X^*, U^*) is the minimizer of Problem 4.1.

Problem 4.2 is a standard LQR problem. By linear optimal control theory, the optimal feedback gain K^* is

$$K^* = R^{-1}B^TP^*, \tag{4.16}$$

where $P^* = P^{*T} > 0$ is the unique solution to the following ARE

$$A^TP + PA + Q - PBR^{-1}B^TP = 0. \tag{4.17}$$

4.1.2 Optimal Output Regulator Design

We first present a solution to regulator equations with known parameters. Then, we develop an adaptive learning strategy to solve X^*, U^* and to find data-based approximation of optimal values P^* and K^* under Assumptions 4.1–4.3 when the system matrices $A, B, \hat{D}$ are unknown.

Define two maps $\mathcal{S}$: $\mathbb{R}^{n\times q_m} \to \mathbb{R}^{n\times q_m}$ and $\bar{\mathcal{S}}(X,U)$: $\mathbb{R}^{n\times q_m} \times \mathbb{R}^{m\times q_m} \to \mathbb{R}^{n\times q_m}$ by

$$\begin{aligned} \mathcal{S}(X) &= X\hat{E} - AX, \\ \bar{\mathcal{S}}(X,U) &= XE - AX - BU, \quad X \in \mathbb{R}^{n\times q_m}, U \in \mathbb{R}^{m\times q_m}. \end{aligned} \tag{4.18}$$

Pick two constant matrices $X_1 \in \mathbb{R}^{n\times q_m}$ and $U_1 \in \mathbb{R}^{m\times q_m}$ such that $CX_1 + JU_1 + \hat{F} = 0$. Then we select $X_i \in \mathbb{R}^{n\times q_m}$ and $U_i \in \mathbb{R}^{m\times q_m}$ for $i = 2, \ldots, h+1$ such that all the vectors $\operatorname{vec}\left(\begin{bmatrix} X_i \\ U_i \end{bmatrix}\right)$ for $i = 2, \ldots, h+1$ form a basis for $\ker(I_{q_m} \otimes [C\ J])$, where $h = (n+m-r)q_m$ is the dimension of the null space of $I_{q_m} \otimes [C\ J]$.

A general solution to (4.11) can be described by a sequence of $\alpha_i \in \mathbb{R}$, with $i = 2, \ldots, h+1$, as

$$(X,U) = (X_1,U_1) + \sum_{i=2}^{h+1} \alpha_i (X_i,U_i). \tag{4.19}$$

Then, (4.10) implies

$$\bar{\mathcal{S}}(X,U) = \bar{\mathcal{S}}(X_1,U_1) + \sum_{i=2}^{h+1} \alpha_i \bar{\mathcal{S}}(X_i,U_i) = \hat{D}. \tag{4.20}$$

Equations (4.19)–(4.20) can be rewritten as

$$\mathcal{A}\chi = b, \tag{4.21}$$

where

$$\mathcal{A} = \begin{bmatrix} \operatorname{vec}\left(\begin{bmatrix} X_2 \\ U_2 \end{bmatrix}\right) & \ldots & \operatorname{vec}\left(\begin{bmatrix} X_{h+1} \\ U_{h+1} \end{bmatrix}\right) & -I \\ \operatorname{vec}(\bar{\mathcal{S}}(X_2,U_2)) & \ldots & \operatorname{vec}(\bar{\mathcal{S}}(X_{h+1},U_{h+1})) & 0 \end{bmatrix},$$

$$\chi^T = \left[\alpha_2, \ldots, \alpha_{h+1}, \left(\text{vec}\left(\begin{bmatrix} X \\ U \end{bmatrix}\right)\right)^T\right],$$

$$b = \begin{bmatrix} \text{vec}\left(\begin{bmatrix} X_1 \\ U_1 \end{bmatrix}\right) \\ -\text{vec}(\bar{\mathcal{S}}(X_1, U_1)) + \hat{D} \end{bmatrix}.$$

The interested reader can consult Odekunle *et al.* (2020) for the details.

4.1.3 Data-Driven Adaptive Design of Optimal Output Regulators

Defining $\bar{x}_i = x - X_i w$ for $i = 0, 1, 2, \ldots, h+1$ with $X_0 = 0_{n \times q_m}$, we have

$$\begin{aligned} \dot{\bar{x}}_i &= Ax + Bu + (\hat{D} - X_i \hat{E})w \\ &= A_j \bar{x}_i + B(K_j \bar{x}_i + u) + (\hat{D} - \mathcal{S}(X_i))w, \end{aligned} \tag{4.22}$$

where $A_j = A - BK_j$. Then, by policy evaluation, we have

$$\begin{aligned} &\bar{x}_i(t+\delta t)^T P_j \bar{x}_i(t+\delta t) - \bar{x}_i(t)^T P_j \bar{x}_i(t) \\ &= \int_t^{t+\delta t} [\bar{x}_i^T (A_j^T P_j + P_j A_j)\bar{x}_i \\ &\quad + 2(u + K_j \bar{x}_i)^T B^T P_j \bar{x}_i + 2w^T (\hat{D} - \mathcal{S}(X_i))^T P_j \bar{x}_i] d\tau \\ &= -\int_t^{t+\delta t} \bar{x}_i^T (Q + K_j^T R K_j)\bar{x}_i d\tau + 2\int_t^{t+\delta t} (u + K_j \bar{x}_i)^T R K_{j+1} \bar{x}_i d\tau \\ &\quad + 2\int_t^{t+\delta t} w^T (\hat{D} - \mathcal{S}(X_i))^T P_j \bar{x}_i d\tau. \end{aligned} \tag{4.23}$$

For positive integer s, define

$$\begin{aligned} \delta_{\bar{x}_i \bar{x}_i} = [&\text{vecv}(\bar{x}_i(t_1)) - \text{vecv}(\bar{x}_i(t_0)), \text{vecv}(\bar{x}_i(t_2)) \\ &- \text{vecv}(\bar{x}_i(t_1)), \ldots, \text{vecv}(\bar{x}_i(t_s)) - \text{vecv}(\bar{x}_i(t_{s-1}))]^T, \end{aligned}$$

$$\Gamma_{\bar{x}_i\bar{x}_i} = \left[\int_{t_0}^{t_1} \bar{x}_i \otimes \bar{x}_i d\tau, \int_{t_1}^{t_2} \bar{x}_i \otimes \bar{x}_i d\tau, \ldots, \int_{t_{s-1}}^{t_s} \bar{x}_i \otimes \bar{x}_i d\tau\right]^T,$$

$$\Gamma_{\bar{x}_i u} = \left[\int_{t_0}^{t_1} \bar{x}_i \otimes u d\tau, \int_{t_1}^{t_2} \bar{x}_i \otimes u d\tau, \ldots, \int_{t_{s-1}}^{t_s} \bar{x}_i \otimes u d\tau\right]^T,$$

$$\Gamma_{\bar{x}_i w} = \left[\int_{t_0}^{t_1} \bar{x}_i \otimes w d\tau, \int_{t_1}^{t_2} \bar{x}_i \otimes w d\tau, \ldots, \int_{t_{s-1}}^{t_s} \bar{x}_i \otimes w d\tau\right]^T,$$

where $t_0 < t_1 < \cdots < t_s$ are positive time instants.

(4.23) implies the following linear equation

$$\Psi_{ij} \begin{bmatrix} \text{vecs}(P_j) \\ \text{vec}(K_{j+1}) \\ \text{vec}((\hat{D} - \mathcal{S}(X_i))^T P_j) \end{bmatrix} = \Phi_{ij}, \tag{4.24}$$

where

$$\Psi_{ij} = [\delta_{\bar{x}_i\bar{x}_i}, -2\Gamma_{\bar{x}_i\bar{x}_i}(I_n \otimes K_j^T R) - 2\Gamma_{\bar{x}_i u}(I_n \otimes R), -2\Gamma_{\bar{x}_i w}],$$
$$\Phi_{ij} = -\Gamma_{\bar{x}_i\bar{x}_i} \text{vec}(Q + K_j^T R K_j).$$

The uniqueness of solution to (4.24) is guaranteed under some rank condition as shown in Lemma 4.2, where the proof is shown in Gao and Jiang (2016a).

Lemma 4.2. For $i = 0, 1, \ldots, h+1$, if there exists a $s^* \in \mathbb{Z}_+$ such that for all $s > s^*$, for any sequence $t_0 < t_1 < \cdots < t_s$,

$$\text{rank}([\Gamma_{\bar{x}_i\bar{x}_i}, \Gamma_{\bar{x}_i u}, \Gamma_{\bar{x}_i w}]) = \frac{n(n+1)}{2} + (m + q_m)n, \tag{4.25}$$

then Ψ_{ij} has full column rank for all $j \in \mathbb{Z}_+$.

Equation (4.24) can be uniquely solved when matrix Ψ_{ij} is of full column rank, i.e.,

$$\begin{bmatrix} \text{vecs}(P_j) \\ \text{vec}(K_{j+1}) \\ \text{vec}((\hat{D} - \mathcal{S}(X_i))^T P_j) \end{bmatrix} = (\Psi_{ij}^T \Psi_{ij})^{-1} \Psi_{ij}^T \Phi_{ij}. \tag{4.26}$$

We can compute $\hat{D}$ for $i = 0$ and $\mathcal{S}(X_i)$ for $i = 1, 2, \ldots, h+1$. From Kleinman (1968), we obtain that $B = P_j^{-1} K_{j+1}^T R$, and $\bar{\mathcal{S}}(X_i, U_i)$. Thus, both $\mathcal{A}$ and b in (4.21) are computable.

Algorithm 4.1 ADP Learning Algorithm for Solving LOORP

Initialize: Compute matrices $X_0, X_1, \ldots, X_{h+1}$. Utilize $u = -K_0 x + \xi$ on $[t_0, t_s]$ with (bounded) exploration noise ξ and $\sigma(A - BK_0) \subset \mathbb{C}^-$. Select a threshold $\epsilon > 0$. Let $i \leftarrow 0, j \leftarrow 0$

repeat

Solve P_j, K_{j+1} from (4.26)

$j \leftarrow j + 1$

until $|P_j - P_{j-1}| \leq \epsilon$

$j^* \leftarrow j, i \leftarrow i + 1$

repeat

Solve $\mathcal{S}(X_i)$ from (4.26)

until $i = h + 1$

Find (X^*, U^*) by solving Problem 4.1

The ADP-based Algorithm 4.1 for dealing with LOORP is presented as follows. The convergence of this algorithm and the stability of the closed-loop systems are shown in Theorems 4.3 and 4.4.

Theorem 4.3. If (4.25) holds, given a stabilizing $K_0 \in \mathbb{R}^{m \times n}$, the sequences $\{P_j\}_{j=0}^{\infty}$, $\{K_j\}_{j=1}^{\infty}$ obtained from solving (4.26) converge to P^* and K^*, respectively.

Theorem 4.4. Considering the continuous-time linear system (4.1)–(4.3), let $u = -K_{j^*} x + L_{j^*} w$ be the approximated optimal control policy obtained from Algorithm 4.1. Then:

(1) The control policy exponentially stabilizes the closed-loop system.

(2) The tracking error $e(t)$ converges to 0 as t goes to infinity.

4.2 Nonlinear Strict-Feedback Systems

In this section, we present a data-driven control approach to address the problem of adaptive optimal output regulation for a class of nonlinear strict-feedback systems. ADP and nonlinear output regulation theories are integrated to compute an adaptive near-optimal tracker without any *a priori* knowledge of the system dynamics.

4.2.1 Problem Formulation

Consider the class of strict-feedback nonlinear systems described by

$$\begin{aligned} \dot{\xi}_1 &= \bar{f}_1(\xi_1, w) + \xi_2, \\ \dot{\xi}_2 &= \bar{f}_2(\xi_1, \xi_2, w) + \xi_3, \\ &\vdots \\ \dot{\xi}_n &= \bar{f}_n(\xi_1, \xi_2, \ldots, \xi_n, w) + \nu, \\ e &= \xi_1 - q_d(w) \end{aligned} \tag{4.27}$$

with an exosystem described by

$$\dot{w} = Ew \tag{4.28}$$

where $\xi = [\xi_1, \ldots, \xi_n]^T \in \mathbb{R}^n$ is the state, $\bar{f}_1, \ldots, \bar{f}_n$ and q_d are sufficiently smooth functions vanishing at the origin. $\nu := \xi_{n+1} \in \mathbb{R}$ is the input. $e \in \mathbb{R}$ is the tracking error. $w \in \mathbb{W}$ is the exostate with $\mathbb{W} \subset \mathbb{R}^p$ a compact and invariant set with respect to the exosystem (4.28). The class of signals $\mathcal{I}(\mathbb{W})$ consists of solutions $w(t) = w(t, w(0))$ of the exosystem starting at $w(0) \in \mathbb{W}$. As in conventional output regulation problems, it is assumed that all the eigenvalues of matrix $E \in \mathbb{R}^{p \times p}$ are simple with zero real part.

By nonlinear output regulation theory, the corresponding nonlinear output regulator equation is

$$\begin{aligned} \left(\frac{\partial \boldsymbol{\xi_1}(w)}{\partial w}\right) Ew &= \bar{f}_1(\boldsymbol{\xi_1}(w), w) + \boldsymbol{\xi_2}(w), \\ \left(\frac{\partial \boldsymbol{\xi_2}(w)}{\partial w}\right) Ew &= \bar{f}_2(\boldsymbol{\xi_1}(w), \boldsymbol{\xi_2}(w), w) + \boldsymbol{\xi_3}(w), \\ &\vdots \\ \left(\frac{\partial \boldsymbol{\xi_n}(w)}{\partial w}\right) Ew &= \bar{f}_n(\boldsymbol{\xi_1}(w), \ldots, \boldsymbol{\xi_n}(w), w) + \boldsymbol{\nu}(w), \\ \boldsymbol{\xi_1}(w) &= q_d(w) \end{aligned} \tag{4.29}$$

where, for $i = 1, 2, \ldots, n$, the solutions $\boldsymbol{\xi_i}(w)$ and $\boldsymbol{\nu}(w)$ are sufficiently smooth functions vanishing at the origin with $\partial \boldsymbol{\xi}_i(w)/\partial w \in \mathbb{R}^{1 \times p}$.

Define the state and input transformations by $x_i = \xi_i - \boldsymbol{\xi_i}(w), i = 1, 2, \ldots, n$, and $u = \nu - \boldsymbol{\nu}(w)$. The error system is

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, w) + x_2, \\ \dot{x}_2 &= f_2(x_1, x_2, w) + x_3, \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \ldots, x_n, w) + u \end{aligned} \tag{4.30}$$

where, for $i = 1, 2, \ldots, n$,

$$f_i(x_1, \ldots, x_i, w) = \bar{f}_i(\xi_1, \ldots, \xi_i, w) - \bar{f}_i(\boldsymbol{\xi_1}, \ldots, \boldsymbol{\xi_i}, w).$$

Note that (4.30) can be rewritten in a compact form

$$\dot{x} = f(x, w) + g(x, w)u \tag{4.31}$$

where $x = [x_1, \ldots, x_n]^T$. f and g are two locally Lipschitz functions satisfying $f(0, w) = 0$ for any $w \in \mathbb{W}$.

Choose a cost criterion associated with (4.28) and (4.31) by

$$J(x_0, w_0, u) = \int_0^\infty (Q(x) + ru^2)dt \tag{4.32}$$

where Q: $\mathbb{R}^n \to \mathbb{R}$ is positive definite and proper, r is a positive constant, and $x_0 = x(0)$ and $w_0 = w(0)$ are initial conditions.

Letting $\boldsymbol{\xi}(w) = [\boldsymbol{\xi_1}(w), \ldots, \boldsymbol{\xi_n}(w)]^T$, the nonlinear model-based optimal tracking control problem is formulated as follows.

Problem 4.3. Given the nonlinear plant (4.27) with exosystem (4.28), find a controller

$$\begin{aligned} \nu^*(\xi, w) &= u^*(x, w) + \boldsymbol{\nu}(w) \\ &:= u^*(\xi - \boldsymbol{\xi}(w), w) + \boldsymbol{\nu}(w) \end{aligned} \tag{4.33}$$

such that:

(1) For any $w_0 \in \mathbb{W}$, the trajectory of (4.27)–(4.28) with (4.33) starting from any initial state exists and is bounded for all $t \geq 0$.

(2) The tracking error satisfies $\lim_{t\to\infty} e(t) = 0$.

(3) The performance index (4.32) achieves its minimum.

Let $\bar{\mathcal{P}}$ denote the set of all continuously differentiable functions from $\mathbb{R}^n \times \mathbb{W}$ to $\mathbb{R}$. Each function $V \in \bar{\mathcal{P}}$ has the property that, for any fixed $\bar{w} \in \mathbb{W}$, $V(x, \bar{w})$ is a positive definite and proper function on $\mathbb{R}^n$. Similar to Jiang and Jiang (2015b), we make the following assumption.

Assumption 4.4. There exists a $V^* \in \bar{\mathcal{P}}$ solving the following HJB equation

$$\left(\frac{\partial V}{\partial w}\right) Ew + \left(\frac{\partial V}{\partial x}\right) f + Q - \frac{1}{4r}\left[\left(\frac{\partial V}{\partial x}\right) g\right]^2 = 0. \tag{4.34}$$

For any continuously differentiable function V and any $u \in \mathbb{R}$, define

$$\mathcal{L}(V, u) = -\left(\frac{\partial V}{\partial w}\right) Ew - \left(\frac{\partial V}{\partial x}\right)(f + gu) - Q - ru^2.$$

The stability of the system (4.31) is discussed in the following theorem.

Theorem 4.5. For any $V^o(x, w) \in \bar{\mathcal{P}}$ and any $u^o(x, w)$ such that $\mathcal{L}(V^o, u^o) \geq 0, \forall (x, w) \in \mathbb{R}^n \times \mathbb{W}$, the equilibrium point $x = 0$ of system (4.31) with u^o is GAS for any $w \in \mathcal{I}(\mathbb{W})$.

Proof. See Gao and Jiang (2018).

Select a control policy based on the gradient of V^*:

$$u^* = -\frac{1}{2r} g^T \left(\frac{\partial V^*}{\partial x}\right)^T. \tag{4.35}$$

From (4.34), we see that the pair (V^*, u^*) satisfies that $\mathcal{L}(V^*, u^*) = 0$. Hence, the system (4.31) with u^* is GAS at the origin. For any other u that achieves $\lim_{t\to\infty} x(t) = 0$, we have

$$J(x_0, w_0, u) = V^*(x_0, w_0) + \int_0^\infty r(u - u^*)^2 dt.$$

Thus, u^* is the optimal control policy and

$$V^*(x_0, w_0) = \min_u J(x_0, w_0, u) = J(x_0, w_0, u^*), \quad \forall (x_0, w_0) \in \mathbb{R}^n \times \mathbb{W}. \tag{4.36}$$

4.2.2 A Novel Policy Iteration (PI) Approach for Solving HJB Equations

It is checkable from the previous section that the key strategy for addressing the Problem 4.3 is to solve both the regulator equation (4.29) and the HJB equation (4.34). Note that (4.34) is a PDE and the solution is positive semidefinite in (x, w), solving it analytically is by no means easy. We will propose a novel PI approach to solve (4.34) with assured convergence. Since the system (4.28) with (4.31) is not stabilizable, the concept of admissible control policy is relaxed as follows.

Definition 4.3. A control policy u: $\mathbb{R}^n \times \mathbb{W} \to \mathbb{R}$ is said to be *admissible* with respect to the cost (4.32) if the following properties hold:

(1) u is continuous on $\mathbb{R}^n \times \mathbb{W}$.

(2) System (4.31) with u is GAS at the origin for any $w \in \mathcal{I}(\mathbb{W})$.

(3) $J(x_0, w_0, u) < \infty$ for any $(x_0, w_0) \in \mathbb{R}^n \times \mathbb{W}$.

Then, we present a model-based PI method starting from an admissible u_1:

(1) *Policy evaluation:* For any integer $i > 0$, solve $V_i(x, w)$ with $V_i(0, w) = 0$ from

$$0 = \left(\frac{\partial V_i}{\partial w}\right) Ew + \left(\frac{\partial V_i}{\partial x}\right)(f + gu_i) + Q + ru_i^2. \tag{4.37}$$

(2) *Policy improvement:* Update the control policy by

$$u_{i+1}(x, w) = -\frac{1}{2r} g^T(x, w) \left(\frac{\partial V_i}{\partial x}(x, w)\right)^T. \tag{4.38}$$

The following theorem discusses the convergence of V_i and u_i generated by the PI method.

Theorem 4.6. Given an admissible control input u_1, consider V_i and u_{i+1} defined by (4.37)–(4.38). For any $i = 1, 2, \ldots$

(1) $V^*(x, w) \leq V_{i+1}(x, w) \leq V_i(x, w), \forall (x, w) \in \mathbb{R}^n \times \mathbb{W}$.

(2) u_{i+1} is admissible.

(3) For each fixed $(x, w) \in \mathbb{R}^n \times \mathbb{W}$, $\{V_i(x, w)\}_{i=1}^{\infty}$ and $\{u_i(x, w)\}_{i=1}^{\infty}$ converge monotonically to $V^*(x, w)$ and $u^*(x, w)$, respectively.

4.2.3 Data-Driven Nonlinear and Adaptive Optimal Tracking Controller Design

A two-phase data-driven learning strategy is presented to solve the nonlinear optimal tracking control problem with unknown system dynamics $\bar{f}_1, \ldots, \bar{f}_n$. We approximate the solution to regulator equation (4.29) in phase one, while the approximate optimal feedback controller is learned by an online PI approach in phase two. The following assumption indicates that the sets where the system is operating are known. Then, one can employ the function approximation technique on these sets.

Assumption 4.5. Suppose an initial controller $\nu(t) := \mu(t)$ is known for system (4.27) such that the closed-loop trajectory $(\xi(t), w(t))$ remains in a compact set $\mathbb{A} \times \mathbb{W} \subset \mathbb{R}^n \times \mathbb{W}$ for any $(\xi(0), w(0)) \in \mathbb{A}_0 \times \mathbb{W}$, where $\mathbb{A}_0 \subset \mathbb{A}$.

The data-driven approximate optimal tracking control problem is described as follows.

Problem 4.4. Given the nonlinear plant (4.27) with exosystem (4.28), design a controller such that, for any initial condition $(\xi(0), w(0)) \in \mathbb{A}_0 \times \mathbb{W}$:

(1) The trajectory of the closed-loop system is bounded for all $t \geq 0$.

(2) The tracking error is uniformly ultimately bounded (UUB) with arbitrarily small ultimate bound.

(3) The designed controller is a suboptimal controller.

Phase-One learning: solving regulator equations

We will approximate the solution to the regulator equation step by step. Let $\{\phi_j(w)\}_{j=1}^{\infty}$ be a sequence of linearly independent smooth basis functions on $\mathbb{W}$. By approximation theory, for $i = 1, 2, \ldots, N$, the solution $\boldsymbol{\xi_{i+1}}$ to regulator equations (4.29) can be approximated by $\hat{\boldsymbol{\xi}}_{\boldsymbol{i+1}}(\boldsymbol{w}) = \sum_{j=1}^{N_{\xi_{i+1}}} \hat{d}_{i+1,j}\phi_j(w)$, where $N_{\xi_{i+1}}$ is a sufficiently large integer, and $\hat{d}_{i+1,j}$ is a constant weight to be trained. For $i = 1, 2, \ldots, n$, let $\{\psi_{i,j}\}_{j=1}^{\infty}$ be a sequence of linearly independent smooth basis functions

on a compact set $\mathbb{D}_i$ containing $\{0\} \times \mathbb{W}$ and $\{[\xi_1 - \hat{\boldsymbol{\xi}}_1(w), \ldots, \xi_i - \hat{\boldsymbol{\xi}}_i(w), w^T]^T | \xi \in \mathbb{A}, w \in \mathbb{W}\} \subset \mathbb{D}_i$ with $\boldsymbol{\xi}_1 = \hat{\boldsymbol{\xi}}_1$. For $i = 2, \ldots, N+1$, letting $\hat{x}_i = \xi_i - \hat{\boldsymbol{\xi}}_{\boldsymbol{i}}$ and $\hat{x}_1 = x_1$, then by (4.27) and (4.29), we have

$$\begin{aligned} \dot{x}_i &= f_i(x_1, \ldots, x_i, w) + \xi_{i+1} - \boldsymbol{\xi_{i+1}} \\ &= f_i(\hat{x}_1, \ldots, \hat{x}_i, w) + \xi_{i+1} - \boldsymbol{\xi_{i+1}} + e_{fi} \end{aligned} \tag{4.39}$$

where $e_{fi} = f_i(x_1, \ldots, x_i, w) - f_i(\hat{x}_1, \ldots, \hat{x}_i, w)$.

(4.39) implies that

$$\begin{aligned} \frac{1}{2}\hat{x}_i^2\Big|_{t_k}^{t_{k+1}} = e_{xi,k} + \int_{t_k}^{t_{k+1}} \Bigg[\sum_{j=1}^{N_{f_i}} \hat{c}_{i,j}\psi_{i,j}(\hat{x}_1, \ldots, \hat{x}_i, w) \\ - \sum_{j=1}^{N_{\xi_{i+1}}} \hat{d}_{i+1,j}\phi_j(w) \Bigg] \hat{x}_i d\tau + \int_{t_k}^{t_{k+1}} \xi_{i+1}\hat{x}_i d\tau \end{aligned} \tag{4.40}$$

where $\sum_{j=1}^{N_{f_i}} \hat{c}_{i,j}\psi_{i,j} := \hat{f}_i$ with $N_{f_i} > 0$ and $\hat{c}_{i,j}$ constant weights. The approximation error $e_{xi,k}$ is

$$\begin{aligned} e_{xi,k} = \frac{1}{2}(\hat{x}_i^2 - x_i^2)\Big|_{t_k}^{t_{k+1}} + \int_{t_k}^{t_{k+1}} [(f_i - \boldsymbol{\xi_{i+1}})x_i + e_{fi}x_i]d\tau \\ - \int_{t_k}^{t_{k+1}} (\hat{f}_i - \hat{\boldsymbol{\xi}}_{\boldsymbol{i+1}})\hat{x}_i + \xi_{i+1}(x_i - \hat{x}_i)d\tau. \end{aligned}$$

Let $\{t_k\}_{k=0}^{l}$ be a strictly increasing sequence with l a sufficiently large positive integer, then the weights $\hat{c}_{i,j}$ and $\hat{d}_{i,j}$ can be solved in terms of least squares solutions (by minimizing $\sum_{k=0}^{l} e_{xi,k}^2$) if the following assumption holds

Assumption 4.6. There exist $l_0 > 0$ and $\delta > 0$, such that for any $l \geq l_0$, we have

$$\frac{1}{l}\sum_{k=0}^{l} \theta_{i,k}^T \theta_{i,k} \geq \delta I_{N_{f_i}+N_{\xi_{i+1}}} \tag{4.41}$$

where

$$\begin{aligned} \theta_{i,k} = \Bigg[\int_{t_k}^{t_{k+1}} \hat{x}_i\psi_{i,1}d\tau, \ldots, \int_{t_k}^{t_{k+1}} \hat{x}_i\psi_{i,N_{f_i}}d\tau, \\ - \int_{t_k}^{t_{k+1}} \hat{x}_i\phi_1 d\tau, \ldots, - \int_{t_k}^{t_{k+1}} \hat{x}_i\phi_{N_{\xi_{i+1}}} d\tau \Bigg]. \end{aligned}$$

Now, we are ready to present an online and data-driven algorithm, Algorithm 4.2, to approximate the solution to (4.29).

Algorithm 4.2 Online Algorithm Solving Regulator Equations

Initialize: Employ $\mu(t)$ and collect the input-state data. $i \leftarrow 1$.
repeat
Solve $\hat{c}_{i,1}, \ldots, \hat{c}_{i,N_{f_i}}$ and $\hat{d}_{i+1,1}, \ldots, \hat{d}_{i+1,N_{\xi_{i+1}}}$ from (4.40)
$i \leftarrow i+1$
until $i = n+1$

Theorem 4.7. Under Assumption 4.6, for $i = 2, \ldots, n+1$ and arbitrarily small constant $\epsilon > 0$, there exists a large enough integer $N_1^* > 0$ such that

$$\left| \sum_{j=0}^{N_{\xi_i}} \hat{d}_{i,j}\phi_j(w) - \boldsymbol{\xi_i}(w) \right| < \epsilon \tag{4.42}$$

for any $w \in \mathbb{W}$ if

$$N_1 := \min\{N_{f1}, \ldots, N_{fn}, N_{\xi_2} \ldots, N_{\xi_{n+1}}\} > N_1^*.$$

Phase-Two learning: solving HJB equations

To begin with, we rewrite the system (4.27) as

$$\dot{x} = f(x,w) + g(x,w)(\nu - \boldsymbol{\nu}(w)) \tag{4.43}$$
$$= f(x,w) + g(x,w)u_i(x,w) + g(x,w)v_i \tag{4.44}$$

where $v_i = \nu - \boldsymbol{\nu} - u_i$.

For each iteration $i \geq 1$, the derivative of $V_i(x,w)$ along the solutions to (4.44) satisfies

$$\begin{aligned}\dot{V}_i &= \frac{\partial V_i}{\partial w}Ew + \frac{\partial V_i}{\partial x}\left[f(x,w) + gu_i(x,w) + gv_i\right] \\ &= -Q(x) - ru_i^2(x,w) - 2ru_{i+1}(x,w)v_i. \end{aligned} \tag{4.45}$$

Integrating both sides of the equation on an interval $[t_k, t_{k+1}]$, it follows that

$$\begin{aligned}&V_i(x(t_{k+1}), w(t_{k+1})) - V_i(x(t_k), w(t_k)) \\ &= \int_{t_k}^{t_{k+1}} [-Q(x) - ru_i^2(x,w) - 2ru_{i+1}(x,w)v_i]d\tau. \end{aligned} \tag{4.46}$$

This implies that

$$V_i(\hat{x}(t_{k+1}), w(t_{k+1})) - V_i(\hat{x}(t_k), w(t_k)) = e_{Vi} + \int_{t_k}^{t_{k+1}} [-Q(\hat{x}) - ru_i^2(\hat{x}, w) - 2ru_{i+1}(\hat{x}, w)\bar{v}_i]d\tau \tag{4.47}$$

where $\bar{v}_i = \nu - \hat{\boldsymbol{\nu}}(w) - u_i(\hat{x}, w)$. From Theorem 4.7, $x - \hat{x} \to 0$ as $N_1 \to \infty$, by continuity, so is the error term e_{Vi}.

By approximation theory, the unknown functions $V_i(\hat{x}, w)$ and $u_{i+1}(\hat{x}, w)$ can be approximated by

$$\hat{V}_i(\hat{x}, w) = \sum_{j=1}^{N_2} \hat{s}_{i,j}\psi_{n,j}^V(\hat{x}, w), \tag{4.48}$$

$$\hat{u}_{i+1}(\hat{x}, w) = \sum_{j=1}^{N_3} \hat{z}_{i,j}\psi_{n,j}^u(\hat{x}, w) \tag{4.49}$$

where $\{\psi_{n,j}^V\}_{j=1}^{\infty}$ and $\{\psi_{n,j}^u\}_{j=1}^{\infty}$ are two sequences of linearly independent smooth basis functions on the compact set $\mathbb{D}_n$. $\hat{s}_{i,j}$ and $\hat{z}_{i,j}$ are weights with N_2 and N_3 two sufficiently large integers.

One can replace V_i and u_{i+1} in (4.47) by their approximations

$$\begin{aligned}&\sum_{j=1}^{N_2} \hat{s}_{i,j}[\psi_{n,j}^V(\hat{x}(t_{k+1}), w(t_{k+1})) - \psi_{n,j}^V(\hat{x}(t_k), w(t_k))] \\ &\quad + \int_{t_k}^{t_{k+1}} 2r \sum_{j=1}^{N_3} \hat{z}_{i,j}\psi_{n,j}^u(\hat{x}, w)\hat{v}_i d\tau + E_{i,k} \\ &= -\int_{t_k}^{t_{k+1}} [Q(\hat{x}) + r\hat{u}_i^2(\hat{x}, w)]d\tau \end{aligned} \tag{4.50}$$

where $\hat{u}_1(\hat{x}, w) = u_1(\hat{x}, w)$, $\hat{v}_i = \nu - \hat{\boldsymbol{\nu}}(w) - \hat{u}_i(\hat{x}, w)$, $E_{i,k}$ is the approximation error. Similar to the phase-one learning, the weights $\hat{s}_{i,j}$ and $\hat{z}_{i,j}$ are obtained in terms of least squares.

The online ADP Algorithm 4.3 for solving the HJB equation is provided as follows, while its convergence analysis is elucidated in the following Theorem 4.8.

Algorithm 4.3 Online ADP Algorithm

Initialize: Choose a sufficiently small threshold $\epsilon > 0$. Employ $\mu(t)$ and collect input-state data online. $i \leftarrow 1$.

repeat

Solve $\hat{s}_{i,1}, \ldots, \hat{s}_{i,N_2}$ and $\hat{z}_{i,1}, \ldots, \hat{z}_{i,N_3}$ from (4.50)

$i \leftarrow i + 1$

until $\sum_{j=1}^{N_2} |\hat{s}_{i,j} - \hat{s}_{i-1,j}|^2 \leq \epsilon$

Theorem 4.8. Under Assumption 4.6, for any arbitrary $\epsilon_1 > 0$, there exist positive integers i^*, N_1^*, N_2^* and N_3^* such that

$$\begin{aligned} \left| \sum_{j=1}^{N_2} \hat{s}_{i^*,j} \psi_{n,j}^V(\hat{x}, w) - V^*(x, w) \right| &\leq \epsilon_1, \\ \left| \sum_{j=1}^{N_3} \hat{z}_{i^*,j} \psi_{n,j}^u(\hat{x}, w) - u^*(x, w) \right| &\leq \epsilon_1 \end{aligned} \tag{4.51}$$

for any $(x, w) \in \mathbb{D}_n$, if $N_1 > N_1^*$, $N_2 > N_2^*$ and $N_3 > N_3^*$.

The following lemma provides a sufficient condition on the uniform boundedness and ultimate boundedness of the solution to (4.43).

Lemma 4.9. If there exists a $V \in \bar{\mathcal{P}}$ such that V and its derivative along the trajectories of (4.43) with some input ν satisfies

$$\begin{aligned} &\alpha_4(|x|) \leq V(x, w) \leq \alpha_5(|x|), \\ &\dot{V}(x, w) \leq -Q(x) + d, \quad \forall (x, w) \in \mathbb{D}, \end{aligned}$$

where $d > 0$ is a constant with α_4 and α_5 two functions of class $\mathcal{K}$. Let $\gamma > 0$ such that $B_\gamma(0) \times \mathbb{W} \subset \mathbb{D}$ and a function α_6 of class $\mathcal{K}$ such that $Q(x) > \alpha_6(|x|)$. Suppose $d < \alpha_6 \circ \alpha_5^{-1} \circ \alpha_4(\gamma)$. Then, for any $|x_0| \leq \alpha_5^{-1} \circ \alpha_4(\gamma)$ and $w_0 \in \mathbb{W}$, $(x(t), w(t))$ stays in $B_\gamma(0) \times \mathbb{W}$ for any $t \geq 0$. Moreover, $x(t)$ is UUB with the ultimate bound $\alpha_4^{-1} \circ \alpha_5 \circ \alpha_6^{-1}(d)$.

Theorem 4.10. Consider the nonlinear plant (4.27), the exosystem (4.28) in closed-loop with the approximate optimal controller obtained

by Algorithms 4.2–4.3, i.e.,

$$\begin{aligned}\nu_{i^*} &= \hat{u}_{i^*}(\hat{x}, w) + \hat{\boldsymbol{\nu}}(w) \\ &:= \hat{u}_{i^*}(\xi - \hat{\boldsymbol{\xi}}(w), w) + \hat{\boldsymbol{\nu}}(w),\end{aligned} \tag{4.52}$$

where i^* is defined in Theorem 4.8. Then, for any initial condition $(\xi(0), w(0)) \in \mathbb{A}_0 \times \mathbb{W}$, the following properties hold:

(1) The trajectory of the closed-loop system is bounded for any $t \geq 0$.

(2) The tracking error $e(t)$ is UUB with its ultimate bound approaching zero as i^*, N_1, N_2, N_3 all tend to infinity.

Combining Theorems 4.8 and 4.10, we see that the data-driven approximate optimal tracking control Problem 4.4 is solved through the controller (4.52) designed by Algorithms 4.2–4.3.

4.3 Multi-Agent Systems

The main purpose of this section is to address the cooperative adaptive optimal output regulation of linear multi-agent systems via ADP and the internal model principle. First, we develop a data-driven distributed control policy to ensure that each follower with uncertain models can achieve asymptotic tracking with disturbance rejection in an optimal sense. Second, we combine ADP and the internal model principle to generalize the result in Section 4.1 to the distributed adaptive optimal output regulation problem. Third, we consider a more practical situation where the leader model (or, the exosystem here) is subject to external disturbance. We will show that, under the proposed data-driven distributed control law, the closed-loop system is leader-to-formation stable (LFS) (see Tanner *et al.*, 2004).

4.3.1 Problem Formulation and Preliminaries

Consider the following class of linear multi-agent systems

$$\dot{v} = Ev + Hw, \tag{4.53}$$

$$\dot{x}_i = A_i x_i + B_i u_i + D_i v, \tag{4.54}$$

$$e_i = C_i x_i + F_i v, \quad i = 1, 2, \ldots, N \tag{4.55}$$

where $x_i \in \mathbb{R}^{n_i}, u_i \in \mathbb{R}$ and $e_i \in \mathbb{R}$ are the state, control input and tracking error of the ith subsystem (follower), respectively. $v \in \mathbb{R}^q$ is the state of the leader, modeled by the exosystem (4.53) which generates both the disturbance $D_i v$ and the reference signal $-F_i v$ (to be tracked by the output $y_i = C_i x_i$) of each follower. The leader is assumed to track a desired trajectory $y_0^* = -F_0 v^*(t)$ with the signal v^* satisfying $\dot{v}^* = Ev^*$. In this setting, the leader input is composed by two parts: $w = \hat{w} + \tilde{w}$, where $\hat{w}$ is a feedback control input $\hat{w} = -K_0(v - v^*)$ with $\sigma(E - HK_0) \subset \mathbb{C}^-$ and $\tilde{w}$ is an external disturbance input. Given the exosystem (4.53) and the plant (4.55), define a time-varying digraph $\mathcal{G}_{\rho(t)} = \{\mathcal{V}, \mathcal{E}_{\rho(t)}\}$. $\mathcal{V} = \{0, 1, \ldots, N\}$ is the node set with node 0 denoting the leader and the remaining N nodes being identified as followers described by (4.55). $\mathcal{E}_{\rho(t)} \subset \mathcal{V} \times \mathcal{V}$ refers to the edge set. Denote $\mathcal{N}_i(t)$ the set of all the nodes j such that $(j, i) \in \mathcal{E}_{\rho(t)}$. The adjacency matrix $\mathcal{A}_{\rho(t)} = [a_{ij}(t)] \in \mathbb{R}^{(N+1)\times(N+1)}$ is defined by $a_{ij}(t) > 0$ if $(j, i) \in \mathcal{E}_{\rho(t)}$ and otherwise $a_{ij}(t) = 0$.

Some standard assumptions are made on the system (4.53)–(4.55).

Assumption 4.7. All the eigenvalues of E are simple with zero real part.

Assumption 4.8. (A_i, B_i) is stabilizable, $\forall 1 \leq i \leq N$.

Assumption 4.9. $\text{rank} \begin{bmatrix} A_i - \lambda I & B_i \\ C_i & 0 \end{bmatrix} = n_i + 1$, $\forall \lambda \in \sigma(E)$, $\forall 1 \leq i \leq N$.

Assumption 4.10. There exists a subsequence $\{i_k\}$ of $\{i\colon i = 0, 1, \ldots\}$ with $t_{i_{k+1}} - t_{i_k} < T$ for some positive T such that each node $j = 1, 2, \ldots, N$ is reachable from node 0 in the union graph $\cup_{l=i_k}^{i_{k+1}-1} \mathcal{G}_{\rho(t_j)}$.

The LFS of multi-agent system (4.53)–(4.55) is defined as follows. Note that this definition is in light of IOS, which is slightly different from Tanner *et al.* (2004).

Definition 4.4. System (4.53)–(4.55) achieves LFS if there exist a function β of class $\mathcal{KL}$ and a function γ of class $\mathcal{K}$ such that, for any initial state error $\eta(0)$ and any measurable essentially bounded input $\tilde{w}$ and $t \geq 0$:

$$|e(t)| \leq \beta(|\eta(0)|, t) + \gamma(\|\tilde{w}\|) \tag{4.56}$$

where $e(t) = [e_1(t)\ e_2(t)\ \ldots\ e_N(t)]^T$.

Under Assumptions 4.7–4.9, LFS can be achieved by system (4.53)–(4.55) in closed-loop with a decentralized controller

$$u_i = -K_{xi}x_i - K_{zi}z_i, \tag{4.57}$$
$$\dot{z}_i = G_1 z_i + G_2 e_i, \quad \forall 1 \leq i \leq N \tag{4.58}$$

where the characteristic polynomial of G_1 is the same as the minimal polynomial of E, and the pair (G_1, G_2) is controllable. In this setting, the pair (G_1, G_2) incorporates an internal model of the matrix E, and (4.58) is an internal model of the ith follower. For $i = 1, 2, \ldots, N$, matrices K_{xi}, K_{zi} are chosen such that

$$A_{ci} = \begin{bmatrix} A_i - B_i K_{xi} & -B_i K_{zi} \\ G_2 C_i & G_1 \end{bmatrix}$$

is a Hurwitz matrix.

The following proposition analyzes the leader-to-formation stability of the closed-loop system with respect to the disturbance $\tilde{w}$.

Proposition 4.1. Under Assumptions 4.7–4.9, for $i = 1, 2, \ldots, N$, the multi-agent system (4.53)–(4.55) in closed-loop with (4.57)–(4.58) is LFS.

In order to ameliorate the transient performance of each subsystem, we develop a robust optimal controller such that the closed-loop system is leader-to-formation stable with respect to the leader disturbance $\tilde{w}$. Moreover, as $v \equiv v^*$, the developed controller is optimal in the sense that it minimizes the following cost

$$J = \int_0^\infty (\tilde{\xi}^T Q \tilde{\xi} + u^T R u) dt \tag{4.59}$$

for the open-loop system

$$\dot{\tilde{\xi}} = \bar{A}\tilde{\xi} + \bar{B}\tilde{u} \tag{4.60}$$

where, for $i = 1, 2, \ldots, N$, $\tilde{u}_i = u_i - U_i v^*$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$. $\tilde{u} = [\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_N]^T$, $\bar{A} = \text{blockdiag}(\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_N)$, $\bar{B} = \text{blockdiag}(\bar{B}_1, \bar{B}_2, \ldots, \bar{B}_N)$, $Q = \text{blockdiag}(Q_1, Q_2, \ldots, Q_N)$ and $R = \text{blockdiag}(R_1, R_2, \ldots, R_N)$.

Based upon optimal control theory, the locally robust optimal control policy is (4.58) with

$$\begin{aligned} u_i^* &= \tilde{u}_i^* + U_i v^* \\ &= -K_{xi}^* \tilde{x}_i - K_{zi}^* \tilde{z}_i + U_i v^* \\ &= -K_{xi}^* x_i - K_{zi}^* z_i, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{4.61}$$

The optimal control gains are

$$[K_{xi}^* \quad K_{zi}^*] = R_i^{-1} \bar{B}_i^T P_i^* := K_i^* \tag{4.62}$$

where P_i^* is the unique solution to the following Riccati equation

$$\bar{A}_i^T P_i^* + P_i^* \bar{A}_i + Q_i - P_i^* \bar{B}_i R_i^{-1} \bar{B}_i^T P_i^* = 0. \tag{4.63}$$

4.3.2 Main Results

We will design a data-driven distributed controller via ADP to achieve LFS under switching network topology. The developed approach is able to approximate the control gains K_i^* for each follower without relying on the knowledge of system matrices A_i, B_i and D_i. To begin with, the internal model (4.58) is modified by

$$\dot{z}_i = G_1 z_i + G_2 \hat{e}_i, \quad i = 1, 2, \ldots, N \tag{4.64}$$

where $\hat{e}_i = y_i + F\zeta_i$. The dynamics of $\zeta_i \in \mathbb{R}^q$ depends on the following equation

$$\dot{\zeta}_i = E\zeta_i + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(\zeta_j - \zeta_i) \quad i = 1, 2, \ldots, N \tag{4.65}$$

where $\zeta_0 = v$.

Then, we rewrite the ith subsystem augmented with the internal model (4.64):

$$\begin{aligned} \dot{\xi}_i &= \bar{A}_i \xi_i + \bar{B}_i u_i + \bar{D}_i \psi_i \\ &= \bar{A}_i^{(k)} \xi_i + \bar{B}_i (K_i^{(k)} \xi_i + u_i) + \bar{D}_i \psi_i \end{aligned}$$

where, for $i = 1, 2, \ldots, N$, $\bar{A}_i^{(k)} = \bar{A}_i - \bar{B}_i K_i^{(k)}$, $\bar{D}_i = \text{blockdiag}(D_i, G_2 F_i)$, $\xi_i = [x_i^T \quad z_i^T]^T \in \mathbb{R}^{m_i}$, $\psi_i = [v_i^T \quad \zeta_i^T]^T \in \mathbb{R}^{2q}$.

By policy evaluation, we have

$$\begin{aligned}
&\xi_i(t+\delta t)^T P_i^{(k)} \xi_i(t+\delta t) - \xi_i(t)^T P_i^{(k)} \xi_i(t) \\
&= \int_t^{t+\delta t} [\xi_i^T ((\bar{A}_i^{(k)})^T P_i^{(k)} + P_i^{(k)} \bar{A}_i^{(k)}) \xi_i + 2\psi_i^T \bar{D}_i^T P_i^{(k)} \xi_i \\
&\quad + 2(u_i + K_i^{(k)} \xi_i)^T \bar{B}_i^T P_i^{(k)} \xi_i] d\tau \\
&= \int_t^{t+\delta t} [-\xi_i^T (Q_i + (K_i^{(k)})^T R_i K_i^{(k)}) \xi_i + 2\psi_i^T \bar{D}_i^T P_i^{(k)} \xi_i. \\
&\quad + 2(u_i + K_i^{(k)} \xi_i)^T R_i K_i^{(k+1)} \xi_i] d\tau. \qquad (4.66)
\end{aligned}$$

For any two vectors a, b and a sufficiently large number $s > 0$, define

$$\delta_a = [\mathrm{vecv}(a(t_1)) - \mathrm{vecv}(a(t_0)), \ldots, \mathrm{vecv}(a(t_s)) - \mathrm{vecv}(a(t_{s-1}))]^T,$$

$$\Gamma_{a,b} = \left[\int_{t_0}^{t_1} a \otimes b d\tau, \int_{t_1}^{t_2} a \otimes b d\tau, \ldots, \int_{t_{s-1}}^{t_s} a \otimes b d\tau \right]^T.$$

(4.66) implies the following linear equation

$$\Psi_i^{(k)} \begin{bmatrix} \mathrm{vecs}(P_i^{(k)}) \\ \mathrm{vec}(K_i^{(k+1)}) \\ \mathrm{vec}(\bar{D}_i^T P_i^{(k)}) \end{bmatrix} = \Phi_i^{(k)} \qquad (4.67)$$

where

$$\begin{aligned}
\Psi_i^{(k)} &= [\delta_{\xi_i}, -2\Gamma_{\xi_i \xi_i} (I \otimes (K_i^{(k)})^T R_i) - 2\Gamma_{\xi_i u_i} (I \otimes R_i), -2\Gamma_{\xi_i \psi_i}], \\
\Phi_i^{(k)} &= -\Gamma_{\xi_i \xi_i} \mathrm{vec}(Q_i + (K_i^{(k)})^T R_i K_i^{(k)}).
\end{aligned}$$

Now, we are ready to present a data-driven ADP algorithm 4.4 which yields approximate solutions to the unknown optimal values K_i^* and P_i^*.

The convergence of Algorithm 4.4 is shown in Theorem 4.11, while the LFS of the closed-loop system is analyzed in Theorem 4.12. The proofs of these theorems are given in Gao *et al.* (2018).

Theorem 4.11. For $i = 1, 2, \ldots, N$, sequences $\{P_i^{(k)}\}_{k=0}^{\infty}$ and $\{K_i^{(k)}\}_{k=1}^{\infty}$ computed by Algorithm 4.4 converge to P_i^* and K_i^*.

Algorithm 4.4 Data-driven ADP Algorithm for Distributed Optimal Controller Design

Initialize: Find a pair (G_1, G_2) such that it incorporates an internal model of E. Select a small $\epsilon > 0$. Apply $u_i = -K_i^{(0)}\xi_i + \nu_i$ on $[t_0, t_s]$ with ν_i an exploration noise. Let $i \leftarrow 1$

repeat

$k \leftarrow -1$

repeat

$k \leftarrow k + 1$

Solve $P_i^{(k)}$ and $K_i^{(k+1)}$ from (4.67)

until $|P_i^{(k)} - P_i^{(k-1)}| < \epsilon$ for $k \geq 1$

$P_i^{\dagger} \leftarrow P_i^{(k)}$

The learned controller is (4.64), (4.65), and

$$u_i = -K_i^{(k+1)}\xi_i := -K_i^{\dagger}\xi_i \tag{4.68}$$

$i \leftarrow i + 1$

until $i = N + 1$

Theorem 4.12. Under Assumptions 4.7–4.10, the multi-agent system (4.53)–(4.55) in closed-loop with the learned controller (4.64), (4.65) and (4.68) is leader-to-formation stable.

The following result compares the cost $J^{\odot}$, for the decentralized controller (4.58), (4.61) with the cost $J^{\dagger}$ for the distributed controller (4.64), (4.65) and (4.68).

Theorem 4.13. There always exist constants $d_1, d_2 > 0$ such that $J^{\dagger} \leq d_1 J^{\odot} + d_2 |\tilde{\zeta}(0)|^2$ if $v \equiv v^*$.

5

Applications

5.1 Model-Free Optimal Biological Motor Control

Humans coordinate movements and interact with the environment through sensory information and motor adaptation in their daily lives. Although extensive research has been made (Bhushan and Shadmehr, 1999; Flash and Hogan, 1985; Harris and Wolpert, 1998; Haruno and Wolpert, 2005; Shadmehr and Mussa-Ivaldi, 1994; Todorov, 2004, 2005; Todorov and Jordan, 2002; Uno *et al.*, 1989; Wolpert and Ghahramani, 2000), the underlying computational mechanism of sensorimotor control and learning is still largely an open problem. Indeed, several recent research findings, including the model-free learning (Haith and Krakauer, 2013; Huang *et al.*, 2011), the active regulation of the motor variability (Cashaback *et al.*, 2015; Lisberger and Medina, 2015; Pekny *et al.*, 2015; Renart and Machens, 2014; Vaswani *et al.*, 2015; Wu *et al.*, 2014), and the presence of suboptimal inference (Acerbi *et al.*, 2014; Bach and Dolan, 2012; Beck *et al.*, 2012; Renart and Machens, 2014), have challenged some of the traditional viewpoints on the sensorimotor learning model. As a result, developing a new computational and system framework is not only necessary but also of great importance.

5.1.1 Single-Joint Human Arm Movement Model

Throughout this section, we focus on the sensorimotor learning model considered by Harris and Wolpert (1998) and Burdet *et al.* (2001). In this experimental setup, the human subject moves a parallel-link direct drive air-magnet floating manipulandum (PFM) in a series of forward point-to-point reaching movements performed on a horizontal tabletop.

The mathematical model of the arm movement is given as follows:

$$dp = vdt, \tag{5.1}$$

$$mdv = (a - bv + f)\, dt, \tag{5.2}$$

$$\tau da = (u - a)dt + G_1 u dw_1 + G_2 u dw_2, \tag{5.3}$$

where $p = [p_{\mathrm{x}}\ p_{\mathrm{y}}]^T$, $v = [v_{\mathrm{x}}\ v_{\mathrm{y}}]^T$, $a = [a_{\mathrm{x}}\ a_{\mathrm{y}}]^T$, and $u = [u_{\mathrm{x}}\ u_{\mathrm{y}}]^T$ denote the two-dimensional hand position, velocity, actuator state, and the control input, respectively; m denotes the mass of the hand, b is the viscosity constant, and τ is the time constant; w_1 and w_2 are standard Brownian motion;

$$G_1 = \begin{bmatrix} c_1 & 0 \\ c_2 & 0 \end{bmatrix} \quad \text{and} \quad G_2 = \begin{bmatrix} 0 & -c_2 \\ 0 & c_1 \end{bmatrix}$$

are gain matrices of the signal dependent noise (Harris and Wolpert, 1998; Liu and Todorov, 2007); and $f = \beta p_{\mathrm{x}}$ with $\beta > 0$ is the divergence force field (DF) generated by the PFM. The values of the system parameters are provided in Table 5.1, which are consistent with the values in Liu and Todorov (2007) and Jiang and Jiang (2014a).

Following Todorov and Jordan (2002) and Liu and Todorov (2007), the optimal control problem is formulated as the one of finding an optimal controller to minimize cost (2.4), where $x = [p^T\ v^T\ a^T]^T$, and Q and R represent the tradeoff between moving accuracy (Q) and the effort exerted by the human subject to accomplish the task (R). Generally, Q and R may have different values in different motor tasks. Even in the same motor task, it is possible that Q and R may slowly change over time.

Table 5.1: Parameters of the hand movement model

Parameters	Description	Value
m	Hand mass	1.3 kg
b	Viscosity constant	10 N $\cdot$ s/m
τ	Time constant	0.05 s
c_1	Noise magnitude	0.075
c_2	Noise magnitude	0.025
β	Force magnitude	150

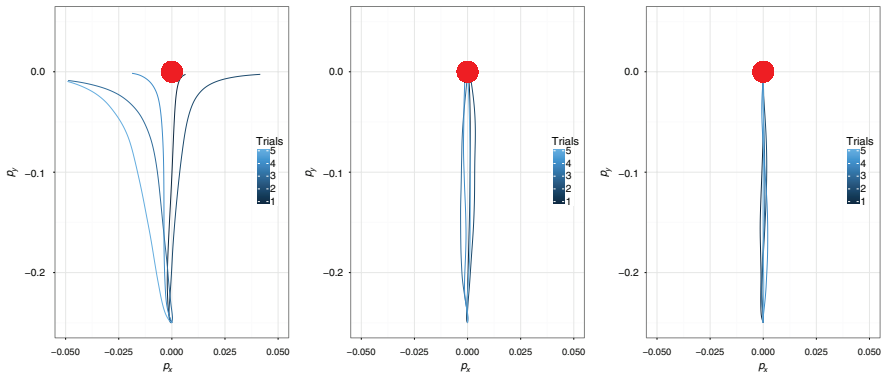
5.1.2 Model-Free Learning in Human Sensorimotor Systems

Now, the optimal motor control problem fits into the standard LQR framework. Hence, ADP methods discussed in Sections 5.1.2 and 5.1.3 can be adapted to validate the learning behavior in human sensorimotor system. For more detailed discussion, see Bian *et al.* (2020).

Since, before conducting the experiment, the human subjects are asked to practice in the null field (NF) for a long period, we assume that the human subject has already adapted to this NF, i.e., an optimal controller with respect to the NF has been obtained. We denote the control gain matrix with respect to this optimal controller in the NF as K_0, and the corresponding performance matrix as P_0.

Once the adaptation to the NF is achieved, i.e., the human subjects have achieved a number of successful trials, the DF will be activated. At this stage, subjects practice in the DF. No information is given to the human subject as when the force field trials will begin. The trajectories in the first five trials in DF are shown in Figure 5.1(a). We can easily see that when the human subject is first exposed to the DF, due to the presence of the force field ($f = \beta p_x$), the variations are amplified by the divergence force, and thus the movement is no longer stable under $u = -K_0 x$.

Starting from K_0 and P_0, a suboptimal control gain matrix is obtained after 50 learning trials. The simulation results of the sensorimotor



(a) First five trajectories of the subject in the DF.

(b) Five independent trajectories of the subject after ADP learning in the DF.

(c) Five independent trajectories of the subject after ADP learning in the NF.

Figure 5.1: Hand movements in X–Y coordinates.

system under this new control policy are given in Figure 5.1(b). Comparing Figures 5.1(b) with 5.1(a), we can see that after learning, the human subject has regained the stability in the DF.

To test the after-effect, we suddenly remove the DF. The after-effect trials are shown in Figure 5.1(c). Obviously, the trajectories are much closer to the y-axis. This is due to the high-gain controller learned in the DF. Here, different from Burdet *et al.* (2001) and Franklin *et al.* (2003), we conjecture that during the (at least, early phase of) learning process, the central nervous system, instead of relying on the internal model completely, simply updates the control strategy through online model-free learning. This is because conducting the model identification is slow and computationally expensive (Shadmehr and Mussa-Ivaldi, 2012, Chapter 9), and thus can only provide limited information to guide the motor adaptation in the early phase of learning. On the other hand, visual and motor sensory feedbacks are extremely active during this phase in the motor learning, which in turn provide a large amount of online data to conduct ADP learning. During the later phase of motor learning, a complete internal model has established, and predictions drawn from the internal model can be incorporated with the visual feedback to provide better estimates of the state.

5.1.3 Two-Joint Human Arm Movement Model

The above learning result can also be extended to the nonlinear two-joint robotic manipulator model (Bhushan and Shadmehr, 1999; Uno *et al.*, 1989):

$$T_s = D_s(\theta)\ddot{\theta} + C_s(\theta, \dot{\theta})\dot{\theta},$$

where $\theta = [\theta_1\ \theta_2]^T$; θ_1 and θ_2 are the joint angles corresponding to the shoulder and the elbow, respectively; and

$$D_s = \begin{bmatrix} k_{s1} & k_{s3}\cos(\theta_2 - \theta_1) \\ k_{s3}\cos(\theta_2 - \theta_1) & k_{s2} \end{bmatrix},$$
$$C_s = \begin{bmatrix} 0 & -k_{s3}\sin(\theta_2 - \theta_1)\dot{\theta}_2 \\ k_{s3}\sin(\theta_2 - \theta_1)\dot{\theta}_1 & 0 \end{bmatrix},$$

and (k_{s1}, k_{s2}, k_{s3}) are the model parameters related on the mass distribution and lengths of the human arm (Bhushan and Shadmehr, 1999). See Figure 5.2 for more details. Note that in practice, the exact values of these parameters may be unknown due to parameter variations. The initial value is given as $\theta(0) = [0\ \pi/2]^T$.

The nonlinear VI-based ADP is used to simulation the arm movement. As we can see form Figure 5.3, the trajectory under the new controller obtained from ADP algorithms is approximately a straight line.

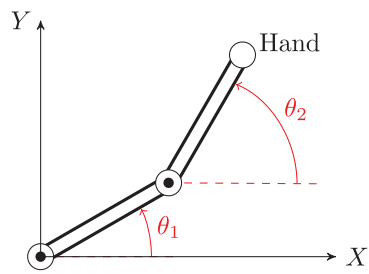


Figure 5.2: A two-joint robotic manipulator. The origin of the (X, Y) coordinates represents the location of the shoulder. X and Y axes represent the side direction and the front direction of the body (Uno *et al.*, 1989).

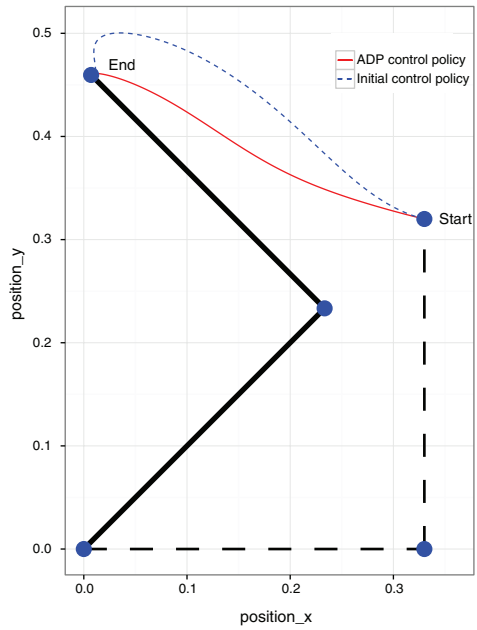


Figure 5.3: Hand movement trajectories. The thick lines represent the initial position (dashed) and the end position (solid) of the human arm. The big dots represent the positions of joints and hands.

Finally, note that our simulation results in above two subsections are consistent with the experimental results provided by different research groups (Burdet *et al.*, 2001; Franklin *et al.*, 2003; Zhou *et al.*, 2016).

5.1.4 Sensorimotor Noise Enhances the Motor Exploration

It has been conjectured (Beers *et al.*, 2002; Harris and Wolpert, 1998; Haruno and Wolpert, 2005) over the past decade that the goal of the motor system is to minimize the endpoint variance caused by the signal-dependent noise. Later, Todorov and Jordan (2002) and Todorov (2004, 2005) further explored this idea by using linear optimal control theory based on the LQR or linear quadratic Gaussian (LQG) methods. However, recently several experimental results (Cashaback *et al.*, 2015; Wu *et al.*, 2014) suggested that the motor variability

facilitates the motor exploration, and as a result increases the learning speed. These surprising results have challenged the optimal control viewpoint in the sense that the motor variability is not purely an unwanted consequence of sensorimotor noise whose effects should by minimized. In this section, we have justified the contribution of motor variability from a robust/adaptive optimal control perspective based on ADP and RADP. Indeed, the motor variability serves a similar role as an exploration noise, which has been proved essential to the ADP and RADP learning. Moreover, our model also shows that the existence of exploration noise does not destabilize the motor system even for learning tasks in an unstable environment.

5.2 Learning-Based Control of Connected and Autonomous Vehicles

The construction of intelligent transportation systems (ITS) attracts considerable attention because of the increasing number of traffic accidents, congestion, and pollution all over the world. Autonomous vehicle technology is the turning-point of development in ITS, which is aimed at reducing driving faults and fuel consumption. By integration of the recent wireless vehicular networking technology in connected vehicles, the connected and autonomous vehicle (CAV) technology is able to further prevent secondary crashes, reducing property damage and injury, congestion and emissions. Among all CAV studies, the controller design of CAV has attracted considerable attention among researchers in the field of control, optimization and communication. For instance, cooperative adaptive cruise controllers (CACC) have been designed for a longitudinal platoon of CAV (see Guo and Yue, 2014; Oncu *et al.*, 2014). The effectiveness of CACC on the safety, traffic flow, and environment has also been tested in different traffic scenarios with mixed human-driven and autonomous vehicles (Arem *et al.*, 2006; Shladover *et al.*, 2012). Cooperative vehicle intersection control (CVIC) is another approach of CAVs (see Lee and Park, 2012). The objective of CVIC is to let vehicles automatically run across the intersection without requiring the traffic signals. Simulation results in Lee and Park (2012) demonstrated that CVIC is able to potentially decrease the traffic pollution

and delay. However, to completely remove the traffic lights may be not easy to realize in the near future. Asadi and Vahidi (2011) proposed the predictive cruise control (PCC). The main idea in PCC is to reduce the idle time of vehicles by using the upcoming traffic signal information. The PCC can help promote a smooth traffic by decreasing the use of breaks, also increase the safety by excluding the red-light violation.

To address the adaptive optimal control problem for CAVs with assured stability, this section will present several intelligent cruise control design strategies under the framework of ADP. Approximate optimal control policies are learned by collected online data from vehicles without relying on the knowledge of either human or vehicle models. They can increase traffic throughput while reducing fuel usage. Most importantly, the stability of the connected vehicle system, which is closely related with safety, is rigorously ensured. We start from the shared robust optimal control design of semi-autonomous vehicles in terms of robust ADP; see Section 5.2.1. In Section 5.2.2, we present an ADP algorithm for CAV platoons with linear dynamics. The presented algorithm is validated through a microscopic traffic simulation in a scenario on Lincoln Tunnel corridor. In Section 5.2.3, a data-driven PCC algorithm is developed in order to regulate the dynamics of CAVs at the vicinity of intersections.

5.2.1 Shared Control of Semi-Autonomous Vehicles

Unlike fully-automated vehicles, semi-autonomous vehicles are operated jointly by a co-pilot controller and a human driver. Here, we will present a novel data-driven cooperative framework that takes into human-vehicle interaction and achieves desired vehicle steering performance. A crucial strategy is to treat the human-vehicle combined system as an interconnected system and to exploit the small-gain theory in the synthesis of vehicle controllers. The proposed shared steering controller is developed without using the (unmeasured) internal states of human driver, but is learned from real-time data collected along the trajectories of the interconnected human-vehicle system.

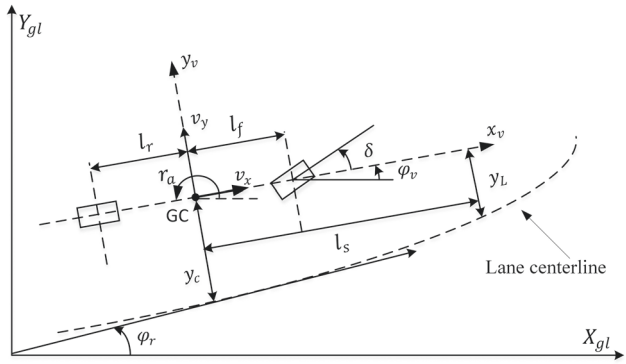


Figure 5.4: Vehicle model illustration.

The vehicle model for steering control consists of the lateral vehicle dynamics, the steering column and the vision-position model for lane-keeping task. Under the assumptions of small angles and constant longitudinal speed, it can be described by

$$\begin{aligned} \dot{x} &= Ax + B(u + T_d) + D\rho_i, \\ y &= Cx, \end{aligned} \tag{5.4}$$

where $x = [v_y \; r_a \; \psi_L \; y_L \; \delta \; \dot{\delta}]^T$, $y = y_c$, and $\psi_L = \psi_v - \psi_r$ as illustrated in Figure 5.4.

We consider the two-point visual driver model developed in Salvucci and Gray (2004):

$$\begin{aligned} \dot{\zeta} &= A_d\zeta + B_d x + D_d\rho_i, \\ T_d &= C_d\zeta, \end{aligned} \tag{5.5}$$

where $\zeta = [\zeta_1 \; \zeta_2]^T$.

Combining (5.4) and (5.5), we obtain an interconnected system model that captures the interaction between the driver and the vehicle. For this system, the learning-based framework in Section 4 can be applied to derive a data-driven learning algorithm for vehicle steering control with stability guarantee. See Huang *et al.* (2019) for the details.

The performance of the proposed learning-based steering assistance system is evaluated on a test road. The lane-keeping error during the whole process is presented in Figure 5.5, where the improved lane-keeping

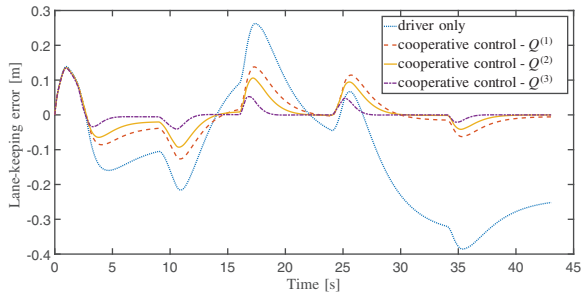


Figure 5.5: Lane-keeping performance comparison between driver and shared control strategies with different Q values.

performance is self-evident with all three configurations. The undershoot and overshoot have been reduced compared to the driver-only scenario.

5.2.2 Adaptive Optimal Cooperative Driving

In this section, we propose a learning-based CACC algorithm that aims to minimize a cost function for CAVs along the exclusive bus lane (XBL) in Lincoln Tunnel corridor. Different from existing model-based CACC algorithms, the proposed approach employs the idea of RL, which does not rely on accurate knowledge of bus dynamics. Considering a time-varying topology where each autonomous vehicle can only receive information from preceding vehicles that are within its communication range, a distributed controller is learned real-time by online headway, velocity, and acceleration data collected from the system trajectories.

Consider a platoon of n autonomous buses. Let h_i be the headway of ith vehicle in the platoon, i.e., the bumper-to-bumper distance between the ith vehicle and the $(i-1)$th vehicle. Time headway (Ioannou and Chien, 1993) and constant spacing (Seiler *et al.*, 2004) are two main spacing policies in CACC. We adopt the former policy since there are different speed limits in each snap of the road. The desired headway is chosen by $h_i^*(t) = \tau v_i(t) + h_0$, where τ is the time headway and h_0 is named the standstill headway. Define $\Delta h_i(t) = h_i(t) - h_i^*(t)$ and $\Delta v_i(t) = v_{i-1}(t) - v_i(t)$. The dynamics of the ith vehicle can be

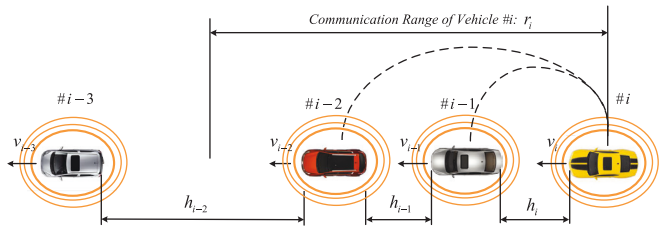


Figure 5.6: Vehicle platoon with time-varying topology.

described by

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + D_i x_{i-1}(t) \tag{5.6}$$

where u_i represents the desired acceleration of vehicle i. For $j = i-1, i$, $x_k = [\Delta h_k, \Delta v_k, a_k]^T$ includes the headway and velocity errors, and the acceleration of vehicle k.

Given the system (5.6), define a time-varying digraph $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$. $\mathcal{V} = \{1, \ldots, n\}$ is the node set. $\mathcal{E}(t) \subset \mathcal{V} \times \mathcal{V}$ refers to the edge set. If the distance between vehicles i and k is smaller than the minimum between the communication range of vehicle k and i at time t, then the edge $(k, i) \in \mathcal{E}(t)$. Denote $\mathcal{N}_i(t)$ the set of all the nodes k such that $(k, i) \in \mathcal{E}(t)$. For instance, the set $\mathcal{N}_i(t) = \{i-1, i-2\}$ at time t since there are only two preceding vehicles staying in the communication range of vehicle i in Figure 5.6.

One of the control objectives is to let the headway and velocity errors and the acceleration of each bus asymptotically converge to zero, i.e., $\lim_{t\to\infty} x_i(t) = 0$ for $i = 1, 2, \ldots, n$. In order to improve the transient response of the system, one can design an optimal controller through minimizing the following cost function

$$J = \int_0^\infty (x^T Q x + u^T R u) d\tau \tag{5.7}$$

where $Q = \text{blockdiag}(Q_1, Q_2, \ldots, Q_n)$, $R = \text{blockdiag}(R_1, R_2, \ldots, R_n)$, $x = [x_1^T, x_2^T, \ldots, x_n^T]^T$, and $u = [u_1, u_2, \ldots, u_n]^T$.

Since it is almost impossible to know the system matrices of all the vehicles considering different types and conditions of vehicles on the road, we propose a data-driven CACC Algorithm in Gao *et al.* (2019a)

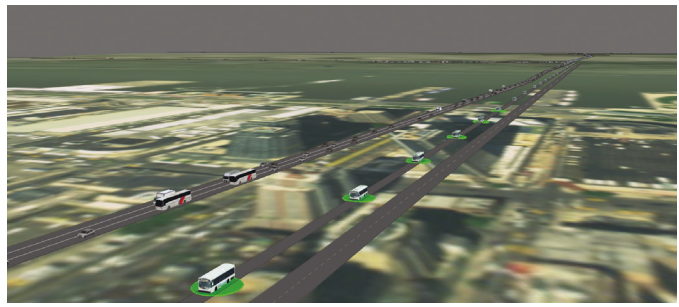


Figure 5.7: Snapshot of buses on XBL in closed-loop with data-driven CACC controller, simulation in paramics.

to learn a distributed controller without knowledge of system matrices A_i and B_i.

For microscopic traffic simulation (see Figure 5.7), we use the real XBL travel time data (field data) collected from the Lincoln Tunnel. We collect the travel times every 15 minutes using the proposed data-driven CACC algorithm. We conclude that the travel time is overall shorter than the present case (field) while the bus volume is increased by 30% from 6:15 AM to 9 AM. Therefore, the proposed data-driven CACC method is able to further improve the traffic conditions for the whole corridor.

5.2.3 Data-Driven Predictive Cruise Control

We have presented an adaptive optimal control approach in Gao *et al.* (2019c) based on ADP to solve the PCC problem of a platoon of CAVs. Similar to CACC, the vehicles in the platoon can exchange its kinetic data through vehicle-to-vehicle (V2V) communication. Moreover, the leader in the platoon can access to the signal phase and time of upcoming traffic intersections through vehicle-to-infrastructure (V2I) communication. In order to develop the presented PCC approach, the reference velocity is first determined for each autonomous vehicle in the platoon. Second, a data-driven adaptive optimal control algorithm was developed to approximate the gains of desired distributed optimal controllers without the exact knowledge of system dynamics. The obtained controller is able to regulate the headway, velocity and acceleration of

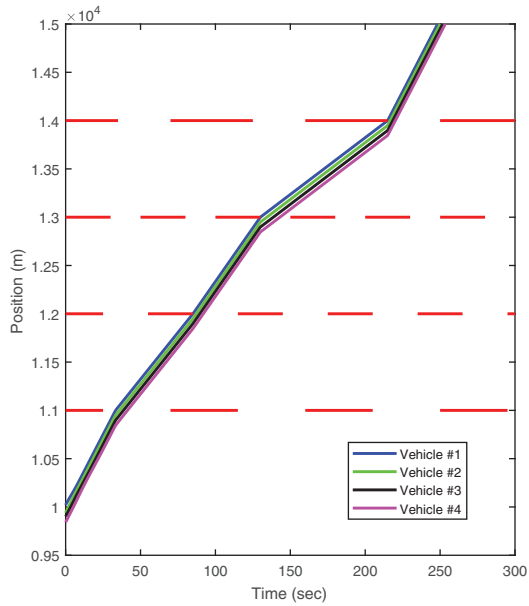


Figure 5.8: Trajectories of vehicles.

each vehicle in a suboptimal sense. The goal of trip time reduction is achieved without compromising vehicle safety and passenger comfort.

Consider a platoon of 4 autonomous vehicles with different dynamics. The allowed maximum and minimum velocity are $V_{\max} = 30$ m/s and $V_{\min} = 0$ m/s. There are four traffic intersection located at 11000 m, 12000 m, 13000 m and 14000 m, respectively. The specific timing information can be referred to Figure 5.8. We first determine the reference velocity for vehicles given the upcoming traffic light information based on the three-phase approach. Then, the online input and state data are collected from $t = 0$ s to $t = 6$ s to approximate the optimal distributed control gain. We update by the learned near-optimal distributed control gains after $t = 6$ s. The trajectories of vehicles using the designed PCC strategy are depicted in Figure 5.8. It can be observed that, without accurate knowledge of system parameters, the designed controller is able to track the desired trajectory timely and reliably, which attests to the safety of the designed control policy. Moreover, by proper determination on reference velocity and design of data-driven controllers, one can see

that the vehicles are able to go across the traffic intersection without unnecessary stopping which increases the traffic mobility.

6

Perspective and Future Work

In this monograph, a novel framework for learning-based optimal control is presented for various important classes of continuous-time linear and nonlinear dynamical systems described by differential equations. We systematically exploit advances in reinforcement learning and adaptive dynamic programming combined with tools in control theory to design learning-based adaptive optimal controllers with guaranteed closed-system stability. The PI and VI methods that have shown great success in solving traditional DP and approximate DP problems (Bellman, 1957; Bertsekas, 2007; Bertsekas and Ioffe, 1996; Howard, 1960; Puterman, 2005) have been generalized to the setting of continuous-time unknown systems evolving in continuous state and input spaces. With the proposed data-driven PI and VI algorithms, the benchmark LQR, LQG, and output regulation problems are revisited from the perspective of learning-based control theory, along with extensions to classes of nonlinear systems, large-scale interconnected systems and multi-agent systems. Rigorous convergence analysis of the adaptive learning algorithms and the stability analysis of the closed-loop systems are studied

in greater details. Robustness of the learning-based controllers is investigated by means of the network/nonlinear small-gain theory as reviewed in Jiang and Liu (2018).

It is our firm belief that the proposed learning-based control approach has wide-ranged applications in the era of robotics and AI. Due to space limitation, in this monograph, we have validated the efficacy of the proposed learning-based design paradigm by means of two representative applications: human motor control and connected and autonomous vehicles. We have used the proposed learning-based control theory to explain how human brain coordinates various challenging hand movement tasks such as reaching and tracking. In particular, the obtained model successfully explains various recent experimental discoveries in the literature, such as the presence of model-free learning (Haith and Krakauer, 2013; Huang *et al.*, 2011), alternative source of motor variability (Acerbi *et al.*, 2014; Bach and Dolan, 2012; Beck *et al.*, 2012; Renart and Machens, 2014), and the fact that actively regulated motor variability promotes sensorimotor learning (Cashaback *et al.*, 2015; Lisberger and Medina, 2015; Pekny *et al.*, 2015; Renart and Machens, 2014; Vaswani *et al.*, 2015; Wu *et al.*, 2014). The experimental phenomena reported in the work of others (Burdet *et al.*, 2001; Franklin *et al.*, 2003; Zhou *et al.*, 2016) are reproduced. Interestingly, our conjecture that learning-based control theory serves as a sound computational principle for human movement is aligned with the vision of Minsky (1954) in the original development of reinforcement learning.

Our future work will be directed at generalizing the proposed learning-based control methods and tools in several directions:

- *Learning-Based Output-Feedback Control.* As most of the learning-based controllers presented in the monograph assume the knowledge of full-state information, the corresponding output-feedback control problem has received little attention. Essentially, we aim to develop tools for learning suboptimal controllers from real-time data that are related to the outputs or partial-states only. Despite its high relevance to practical applications with partially observable states, the conventional model-based observer theory and the separation principle are not directly applicable.

- *Learning-Based Control of Stochastic Nonlinear Systems.* In view of the rich literature of modern nonlinear control, there are many opportunities to generalize the presented results to much broader classes of nonlinear systems, beyond the benchmark class of linear stochastic systems considered in this monograph.

- *Learning-Based Control of Hybrid and Switching Systems.* Hybrid systems (Goebel *et al.*, 2012; Haddad *et al.*, 2014; van der Schaft and Schumacher, 2000) and switching systems (Lee and Jiang, 2008; Liberzon, 2003) are two important classes of dynamical systems that exhibit both continuous- and discrete-time dynamics. How to generalize the learning-based control approach developed here for continuous-time systems to important classes of hybrid systems and switching systems is an interesting topic that deserves deep investigation.

- *Learning-Based Control of Infinite-Dimensional Systems.* Systems represented by functional differential equations (Hale and Lunel, 1993; Karafyllis and Jiang, 2011) and partial differential equations (Christofides, 2001; Karafyllis and Krstić, 2018; Krstić, 2009) are two important classes of infinite-dimensional systems. There has been few research devoted to the learning-based control for this type of infinite-dimensional dynamical systems, in spite of their clear importance in theory and applications. New tools and methods need to be developed that go beyond the results proposed here for the learning-based control of finite-dimensional dynamical systems.

Last but not least, as RL has found many applications in robotics (Kober *et al.*, 2013; Sutton and Barto, 2018), it is a very promising area that demands more learning-based control solutions.

Acknowledgements

First and foremost, this piece of work is dedicated to the memory of Richard Bellman, the inventor of Dynamic Programming, for his vision on the importance of "modern computing machines" in dealing with complex decision making problems. We would like to thank the many people who contributed directly or indirectly to our work. We are very grateful to Drs. Dimitri Bertsekas, John Tsitsiklis, Paul Werbos and Frank Lewis for their seminal works in approximate/adaptive dynamic programming that continue to play an influential role in our research. We would also like to thank Drs. Panos Antsaklis, Alessandro Astolfi, Bob Bitmead, Lei Guo, Miroslav Krstić, and Iven Mareels for various constructive comments on our work at different occasions. The first author wants to express his deep gratitude to his collaborators, especially Frank Lewis and his former students Bahare Kiumarsi, Hamidreza Modares, Kyriakos G. Vamvoudakis, and his other former and current graduate students Yu Jiang, Mengzhe Huang, Eric Mauro, Bo Pang for joint work on this topic of learning-based control. Finally, we sincerely thank the National Science Foundation for its continuous financial support to our research.

References

Acerbi, L., S. Vijayakumar, and D. M. Wolpert (2014). "On the origins of suboptimality in human probabilistic inference". *PLOS Computational Biology*. 10(6): 1–23.

Adams, J. A. (1971). "A closed-loop theory of motor learning". *Journal of Motor Behavior*. 3(2): 111–150.

Anderson, B. D. O. and J. B. Moore (1989). *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice Hall International, Inc.

Antsaklis, P. J., K. M. Passino, and S. J. Wang (1991). "An introduction to autonomous control systems". *IEEE Control Systems Magazine*. 11(4): 5–13.

Antsaklis, P. J. and A. Rahnama (2018). "Control and machine intelligence for system autonomy". *Journal of Intelligent and Robotic Systems, 30th Year Anniversary Special Issue*. 91(1): 23–24.

Arapostathis, A., V. S. Borkar, and M. K. Ghosh (2012). *Ergodic Control of Diffusion Processes*. New York, NY: Cambridge University Press.

Arapostathis, A., V. S. Borkar, and K. Kumar (2014). "Convergence of the relative value iteration for the ergodic control problem of nondegenerate diffusions under near-monotone costs". *SIAM Journal on Control and Optimization*. 52(1): 1–31.

Arem, B. van, C. van Driel, and R. Visser (2006). "The impact of cooperative adaptive cruise control on traffic-flow characteristics". *IEEE Transactions on Intelligent Transportation Systems*. 7(4): 429–436.

Asadi, B. and A. Vahidi (2011). "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time". *IEEE Transactions on Control Systems Technology*. 19(3): 707–714.

Åström, K. J. and B. Wittenmark (1997). *Adaptive Control*. 2nd Ed. Reading, MA: Addison-Wesley.

Bach, D. R. and R. J. Dolan (2012). "Knowing how much you don't know: A neural organization of uncertainty estimates". *Nature Reviews Neuroscience*. 13(8): 572–586.

Baird, III, L. C. (1993). "Advantage updating". *Tech. rep.* No. WL–TR-93-1146. Washington DC: Wright-Patterson Air Force Base Ohio: Wright Laboratory.

Baird, III, L. C. (1994). "Reinforcement learning in continuous time: Advantage updating". In: *Proceedings of the 1999 International Conference on Neural Networks*. Vol. 4. 2448–2453.

Banks, H. and K. Ito (1991). "A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems". *SIAM Journal on Control and Optimization*. 29(3): 499–515.

Barto, A. G., P. S. Tomas, and R. S. Sutton (2017). "Some recent applications of reinforcement learning". In: *Proceedings of the Eighteenth Yale Workshop on Adaptive and Learning Systems*.

Beard, R. W. (1995). "Improving the closed-loop performance of nonlinear systems". *PhD thesis*. Rensselaer Polytechnic Institute.

Beard, R. W., G. N. Saridis, and J. T. Wen (1997). "Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation". *Automatica*. 33(12): 2159–2177.

Beck, J. M., W. J. Ma, X. Pitkow, P. E. Latham, and A. Pouget (2012). "Not noisy, just wrong: The role of suboptimal inference in behavioral variability". *Neuron*. 74(1): 30–39.

Beers, R. J. van, P. Baraduc, and D. M. Wolpert (2002). "Role of uncertainty in sensorimotor control". *Philosophical Transactions of the Royal Society of London B: Biological Sciences*. 357(1424): 1137–1145.

Bellman, R. E. (1954). "Dynamic programming and a new formalism in the calculus of variations". *Proceedings of the National Academy of Sciences of the United States of America*. 40(4): 231–235.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Benner, P. and R. Byers (1998). "An exact line search method for solving generalized continuous-time algebraic Riccati equations". *IEEE Transactions on Automatic Control*. 43(1): 101–107.

Benner, P., J.-R. Li, and T. Penzl (2008). "Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems". *Numerical Linear Algebra with Applications*. 15(9): 755–777.

Bensoussan, A. and J. Frehse (1992). "On Bellman equations of ergodic control in R^n". In: *Applied Stochastic Analysis*. Ed. by I. Karatzas and D. Ocone. Berlin, Heidelberg: Springer. 21–29.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*. 3rd Ed. Vol. 1. Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control*. 3rd Ed. Vol. 2. Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2011). "Approximate policy iteration: A survey and some new methods". *Journal of Control Theory and Applications*. 9(3): 310–335.

Bertsekas, D. P. (2013). *Abstract Dynamic Programming*. Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2017). "Value and policy iterations in optimal control and adaptive dynamic programming". *IEEE Transactions on Neural Networks and Learning Systems*. 28(3): 500–509.

Bertsekas, D. P. (2019). *Reinforcement Learning and Optimal Control*. Belmont, MA: Athena Scientific.

Bertsekas, D. P. and S. Ioffe (1996). "Temporal differences-based policy iteration and applications in neuro-dynamic programming". *Tech. rep.* No. Report LIDS-P-2349. Cambridge, MA: Lab. for Info. and Decision Systems, MIT.

Bhatnagar, S., R. S. Sutton, M. Ghavamzadeh, and M. Lee (2009). "Natural actor-critic algorithms". *Automatica*. 45(11): 2471–2482.

Bhushan, N. and R. Shadmehr (1999). "Computational nature of human adaptive control during learning of reaching movements in force fields". *Biological Cybernetics*. 81(1): 39–60.

Bian, T., Y. Jiang, and Z. P. Jiang (2014). "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems". *Automatica*. 50(10): 2624–2632.

Bian, T., Y. Jiang, and Z. P. Jiang (2015). "Decentralized adaptive optimal control of large-scale systems with application to power systems". *IEEE Transactions on Industrial Electronics*. 62(4): 2439–2447.

Bian, T., Y. Jiang, and Z. P. Jiang (2016). "Adaptive dynamic programming for stochastic systems with state and control dependent noise". *IEEE Transactions on Automatic Control*. 61(12): 4170–4175.

Bian, T. and Z. P. Jiang (2016a). "Stochastic adaptive dynamic programming for robust optimal control design". In: *Control of Complex Systems: Theory and Applications*. Ed. by K. G. Vamvoudakis and S. Jagannathan. Cambridge, MA: Butterworth-Heinemann. Chap. 7. 211–245.

Bian, T. and Z. P. Jiang (2016b). "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design". *Automatica*. 71: 348–360.

Bian, T. and Z. P. Jiang (2016c). "Value iteration, adaptive dynamic programming, and optimal control of nonlinear systems". In: *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*. Las Vegas, USA. 3375–3380.

Bian, T. and Z. P. Jiang (2017). "A tool for the global stabilization of stochastic nonlinear systems". *IEEE Transactions on Automatic Control*. 62(4): 1946–1951.

Bian, T. and Z. P. Jiang (2018). "Stochastic and adaptive optimal control of uncertain interconnected systems: A data-driven approach". *Systems & Control Letters*. 115(5): 48–54.

Bian, T. and Z. P. Jiang (2019a). "Continuous-time robust dynamic programming". *SIAM Journal on Control and Optimization*. 57(6): 4150–4174.

Bian, T. and Z. P. Jiang (2019b). "Reinforcement learning for linear continuous-time systems: An incremental learning approach". *IEEE/CAA Journal of Automatica Sinica*. 6(2): 433–440.

Bian, T., D. Wolpert, and Z. P. Jiang (2020). "Model-free robust optimal feedback mechanisms of biological motor control". *Neural Computation*. 32(3): 562–595.

Borkar, V. S. (2006). "Ergodic control of diffusion processes". In: *Proceedings of the International Congress of Mathematicians*. 1299–1309.

Bradtke, S. J. and M. O. Duff (1994). "Reinforcement learning methods for continuous-time Markov decision problems". In: *Advances in Neural Information Processing Systems 7*. MIT Press. 393–400.

Bryson, J. A. E. (1994). *Control of Spacecraft and Aircraft*. Princeton, NJ: Princeton University Press.

Burdet, E., R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato (2001). "The central nervous system stabilizes unstable dynamics by learning optimal impedance". *Nature*. 414(6862): 446–449.

Cashaback, J. G. A., H. R. McGregor, and P. L. Gribble (2015). "The human motor system alters its reaching movement plan for task-irrelevant, positional forces". *Journal of Neurophysiology*. 113(7): 2137–2149.

Chen, C., H. Modares, K. Xie, F. Lewis, Y. Wan, and S. Xie (2019). "Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics". *IEEE Transactions on Automatic Control*.

Christofides, P. D. (2001). *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*. New York: Springer Science+Business Media, LLC.

Clarke, F. (2013). *Functional Analysis, Calculus of Variations and Optimal Control*. London: Springer.

Dahlquist, G. and Å. Björck (1973). *Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall.

Damm, T. (2004). *Rational Matrix Equations in Stochastic Control*. Berlin, Heidelberg: Springer.

Damm, T. and D. Hinrichsen (2001). "Newton's method for a rational matrix equation occurring in stochastic control". *Linear Algebra and its Applications*. 332–334: 81–109.

Das, T. K., A. Gosavi, S. Mahadevan, and N. Marchalleck (1999). "Solving semi-Markov decision problems using average reward reinforcement learning". *Management Science*. 45(4): 560–574.

Dayan, P. and B. W. Balleine (2002). "Reward, motivation, and reinforcement learning". *Neuron*. 36(2): 285–298.

Ding, Z. (2006). "Output regulation of uncertain nonlinear systems with nonlinear exosystems". *IEEE Transactions on Automatic Control*. 51(3): 498–503.

Ding, Z. (2013). "Consensus output regulation of a class of heterogeneous nonlinear systems". *IEEE Transactions on Automatic Control*. 58(10): 2648–2653.

Doya, K. (2000). "Reinforcement learning in continuous time and space". *Neural Computation*. 12(1): 219–245.

Doya, K. (2002). "Metalearning and neuromodulation". *Neural Networks*. 15(4–6): 495–506.

Doya, K., K. Samejima, K.-I. Katagiri, and M. Kawato (2002). "Multiple model-based reinforcement learning". *Neural Computation*. 14(6): 1347–1369.

Evans, L. C. (2005). "An introduction to mathematical optimal control theory". Lecture Notes, University of California, Department of Mathematics, Berkeley.

Flash, T. and N. Hogan (1985). "The coordination of arm movements: An experimentally confirmed mathematical model". *The Journal of Neuroscience*. 5(7): 1688–1703.

Fleming, W. H. and R. Rishel (1975). *Deterministic and Stochastic Optimal Control*. New York: Springer.

Francis, B. (1977). "The linear multivariable regulator problem". *SIAM Journal on Control and Optimization*. 15(3): 486–505.

Francis, B. A. and W. M. Wonham (1976). "The internal model principle of control theory". *Automatica*. 12(5): 457–465.

Franklin, D. W., E. Burdet, R. Osu, M. Kawato, and T. Milner (2003). "Functional significance of stiffness in adaptation of multijoint arm movements to stable and unstable dynamics". *Experimental Brain Research*. 151(2): 145–157.

Gao, W. and Z. P. Jiang (2016a). "Adaptive dynamic programming and adaptive optimal output regulation of linear systems". *IEEE Transactions on Automatic Control*. 61(12): 4164–4169.

Gao, W. and Z.-P. Jiang (2016b). "Nonlinear and adaptive suboptimal control of connected vehicles: A global adaptive dynamic programming approach". *Journal of Intelligent & Robotic Systems*: 1–15.

Gao, W. and Z. P. Jiang (2018). "Learning-based adaptive optimal tracking control of strict-feedback nonlinear systems". *IEEE Transactions on Neural Networks and Learning Systems*. 29(6): 2614–2624.

Gao, W., Z. P. Jiang, F. L. Lewis, and Y. Wang (2018). "Leader-to-formation stability of multiagent systems: An adaptive optimal control approach". *IEEE Transactions on Automatic Control*. 63(10): 3581–3587.

Gao, W., J. Gao, K. Ozbay, and Z. P. Jiang (2019a). "Reinforcement-learning-based cooperative adaptive cruise control of buses in the Lincoln tunnel corridor with time-varying topology". *IEEE Transactions on Intelligent Transportation Systems*. 20(10): 3796–3805.

Gao, W., Y. Jiang, and M. Davari (2019b). "Data-driven cooperative output regulation of multi-agent systems via robust adaptive dynamic programming". *IEEE Transactions on Circuits and Systems II: Express Briefs*. 66(3): 447–451.

Gao, W., A. Odekunle, Y. Chen, and Z.-P. Jiang (2019c). "Predictive cruise control of connected and autonomous vehicles via reinforcement learning". *IET Control Theory & Applications*. 13(7): 2849–2855.

Glimcher, P. W. (2011). "Understanding dopamine and reinforcement learning: The dopamine reward prediction error hypothesis". *Proceedings of the National Academy of Sciences*. 108(Supplement 3): 15647–15654.

Goebel, R., R. G. Sanfelice, and A. R. Teel (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

Guo, G. and W. Yue (2014). "Sampled-data cooperative adaptive cruise control of vehicles with sensor failures". *IEEE Transactions on Intelligent Transportation Systems*. 15(6): 2404–2418.

Haddad, W. M., V. Chellaboina, and S. G. Nersesov (2014). *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton University Press.

Haith, A. M. and J. W. Krakauer (2013). "Model-based and model-free mechanisms of human motor learning". In: *Progress in Motor Control*. Ed. by M. J. Richardson, M. A. Riley, and K. Shockley. Vol. 782. *Advances in Experimental Medicine and Biology*. New York: Springer. Chap. 1. 1–21.

Hale, J. K. and S. M. V. Lunel (1993). *Introduction to Functional Differential Equations*. New York: Springer-Verlag.

Harris, C. M. and D. M. Wolpert (1998). "Signal-dependent noise determines motor planning". *Nature*. 394(6695): 780–784.

Haruno, M. and D. M. Wolpert (2005). "Optimal control of redundant muscles in step-tracking wrist movements". *Journal of Neurophysiology*. 94(6): 4244–4255.

Haussmann, U. (1971). "Optimal stationary control with state control dependent noise". *SIAM Journal on Control*. 9(2): 184–198.

Haussmann, U. (1973). "Stability of linear systems with control dependent noise". *SIAM Journal on Control*. 11(2): 382–394.

He, H. and X. Zhong (2018). "Learning without external reward". *IEEE Computational Intelligence Magazine*. 13(3): 48–54.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. Cambridge, MA: The MIT Press.

Huang, J. (2004). *Nonlinear Output Regulation: Theory and Applications*. Philadelphia, PA: SIAM.

Huang, J. and Z. Chen (2004). "A general framework for tackling the output regulation problem". *IEEE Transactions on Automatic Control*. 49(12): 2203–2218.

Huang, M., W. Gao, Y. Wang, and Z. P. Jiang (2019). "Data-driven shared steering control of semi-autonomous vehicles". *IEEE Transactions on Human-Machine Systems*. 49(4): 350–361.

Huang, V. S., A. Haith, P. Mazzoni, and J. W. Krakauer (2011). "Rethinking motor learning and savings in adaptation paradigms: Model-free memory for successful actions combines with internal models". *Neuron*. 70(4): 787–801.

Ioannou, P. A. and C. C. Chien (1993). "Autonomous intelligent cruise control". *IEEE Transactions on Vehicular Technology*. 42(4): 657–672.

Isidori, A. and C. I. Byrnes (1990). "Output regulation of nonlinear systems". *IEEE Transactions on Automatic Control*. 35(2): 131–140.

Jiang, Y. and Z. P. Jiang (2011). "Approximate dynamic programming for optimal stationary control with control-dependent noise". *IEEE Transactions on Neural Networks*. 22(12): 2392–2398.

Jiang, Y. and Z. P. Jiang (2012a). "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics". *Automatica*. 48(10): 2699–2704.

Jiang, Y. and Z. P. Jiang (2012b). "Robust adaptive dynamic programming for large-scale systems with an application to multimachine power systems". *IEEE Transactions on Circuits and Systems II: Express Briefs*. 59(10): 693–697.

Jiang, Y. and Z. P. Jiang (2013a). "Robust adaptive dynamic programming with an application to power systems". *IEEE Transactions on Neural Networks and Learning Systems*. 24(7): 1150–1156.

Jiang, Y. and Z. P. Jiang (2014a). "Adaptive dynamic programming as a theory of sensorimotor control". *Biological Cybernetics*. 108(4): 459–473.

Jiang, Y. and Z. P. Jiang (2014b). "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems". *IEEE Trans. Neural Networks and Learning Syst.* 25(5): 882–893.

Jiang, Y. and Z. P. Jiang (2015a). "A robust adaptive dynamic programming principle for sensorimotor control with signal-dependent noise". *Journal of Systems Science and Complexity*. 28(2): 261–288.

Jiang, Y. and Z. P. Jiang (2015b). "Global adaptive dynamic programming for continuous-time nonlinear systems". *IEEE Transactions on Automatic Control*. 60(11): 2917–2929.

Jiang, Y. and Z. P. Jiang (2017). *Robust Adaptive Dynamic Programming*. Hoboken, NJ: Wiley-IEEE Press.

Jiang, Z. P. and I. M. Y. Mareels (1997). "A small-gain control method for nonlinear cascaded systems with dynamic uncertainties". *IEEE Transactions on Automatic Control*. 42(3): 292–308.

Jiang, Z. P. and Y. Jiang (2013b). "Robust adaptive dynamic programming for linear and nonlinear systems: An overview". *European Journal of Control*. 19(5): 417–425.

Jiang, Z. P. and T. Liu (2018). "Small-gain theory for stability and control of dynamical networks: A survey". *Annual Reviews in Control*. 46: 58–79.

Jiang, Z. P., A. R. Teel, and L. Praly (1994). "Small-gain theorem for ISS systems and applications". *Mathematics of Control, Signals and Systems*. 7(2): 95–120.

Jiang, Z. P., I. M. Mareels, and Y. Wang (1996). "A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems". *Automatica*. 32(8): 1211–1215.

Johnson, C. D. (1971). "Accommodation of external disturbances in linear regulator and servomechanism problems". *IEEE Transactions on Automatic Control*. 16: 635–644.

Kalman, R. E. (1960). "Contributions to the theory of optimal control". *Boletin de la Sociedad Matematica Mexicana*. 5: 102–119.

Kamalapurkar, R., H. Dinh, S. Bhasin, and W. E. Dixon (2015). "Approximate optimal trajectory tracking for continuous-time nonlinear systems". *Automatica*. 51: 40–48.

Kamalapurkar, R., P. Walters, J. Rosenfeld, and W. E. Dixon (2018). *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*. Springer.

Karafyllis, I. and Z. P. Jiang (2011). *Stability and Stabilization of Nonlinear Systems*. London: Springer.

Karafyllis, I. and M. Krstić (2018). *Input-to-State Stability for PDEs*. London: Springer.

Khas'minskii, R. (1967). "Necessary and sufficient conditions for the asymptotic stability of linear stochastic systems". *Theory of Probability & Its Applications*. 12(1): 144–147.

Khas'minskii, R. (2012). *Stochastic Stability of Differential Equations*. Berlin, Heidelberg: Springer.

Kiumarsi, B., K. G. Vamvoudakis, H. Modares, and F. L. Lewis (2017). "Optimal and autonomous control using reinforcement learning: A survey". *IEEE Transactions on Neural Networks and Learning Systems*. 29(6): 32–50.

Kleinman, D. L. (1968). "On an iterative technique for Riccati equation computations". *IEEE Transactions on Automatic Control*. 13(1): 114–115.

Kleinman, D. L. (1969). "Optimal stationary control of linear systems with control-dependent noise". *IEEE Transactions on Automatic Control*. 14(6): 673–677.

Kober, J., J. A. Bagnell, and J. Peters (2013). "Reinforcement learning in robotics: A survey". *The Int. J. Robotics Research*. 32(11): 1228–1274.

Kolm, P. N., R. Tütüncü, and F. J. Fabozzi (2014). "60 years of portfolio optimization: Practical challenges and current trends". *European Journal of Operational Research*. 234(2): 356–371.

Krener, A. J. (1992). "The construction of optimal linear and nonlinear regulators". In: *Systems, Models and Feedback: Theory and Applications*. Ed. by A. Isidori and T. J. Tarn. Vol. 12. Boston, Birkhauser. 301–322.

Krstić, M. (2009). *Delay Compensation for Nonlinear, Adaptive, and PDE Systems*. Boston: Birkhäuser.

Krstić, M., I. Kanellakopoulos, and P. V. Kokotovic (1995). *Nonlinear and Adaptive Control Design*. New York, NY: John Wiley & Sons, Inc.

Kučera, V. (1973). "A review of the matrix Riccati equation". *Kybernetika*. 9(1): 42–61.

Kushner, H. J. (1967). "Optimal discounted stochastic control for diffusion processes". *SIAM Journal on Control*. 5(4): 520–531.

Kushner, H. J. and G. G. Yin (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer.

Lancaster, P. and L. Rodman (1995). *Algebraic Riccati Equations*. New York: Oxford University Press.

Lanzon, A., Y. Feng, B. D. O. Anderson, and M. Rotkowitz (2008). "Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method". *IEEE Transactions on Automatic Control*. 53(10): 2280–2291.

Leake, R. and R. Liu (1967). "Construction of suboptimal control sequences". *SIAM Journal on Control*. 5(1): 54–63.

Lee, T. C. and Z. P. Jiang (2008). "Uniform asymptotic stability of nonlinear switched systems with an application to mobile robots". *IEEE Trans. Automatic Control*. 53(5): 1235–1252.

Lee, J. and B. Park (2012). "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment". *IEEE Transactions on Intelligent Transportation Systems*. 13(1): 81–90.

Lewis, F. L., D. M. Dawson, and C. T. Abdallah (2004). *Robot Manipulator Control: Theory and Practice*. 2nd Ed. New York, NY: Marcel Dekker, Inc.

Lewis, F. L. and D. Vrabie (2009). "Reinforcement learning and adaptive dynamic programming for feedback control". *IEEE Circuits and Systems Magazine*. 9(3): 32–50.

Lewis, F. L., D. Vrabie, and V. L. Syrmos (2012a). *Optimal Control*. 3rd Ed. John Wiley & Sons, Inc.

Lewis, F. L., D. Vrabie, and K. G. Vamvoudakis (2012b). "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers". *IEEE Control Systems*. 32(6): 76–105.

Liberzon, D. (2003). *Switching in Systems and Control*. Birkhauser.

Liberzon, D. (2012). *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ: Princeton University Press.

Lisberger, S. G. and J. F. Medina (2015). "How and why neural and motor variation are related". *Current Opinion in Neurobiology*. 33: 110–116.

Liu, D. and E. Todorov (2007). "Evidence for the flexible sensorimotor strategies predicted by optimal feedback control". *The Journal of Neuroscience*. 27(35): 9354–9368.

Liu, L., Z. Chen, and J. Huang (2009). "Parameter convergence and minimal internal model with an adaptive output regulation problem". *Automatica*. 45(5): 1306–1311.

Liu, T., Z. P. Jiang, and D. J. Hill (2014). *Nonlinear Control of Dynamic Networks*. New York, NY: CRC Press.

Liu, D., Q. Wei, D. Wang, X. Yang, and H. Li (2017). *Adaptive Dynamic Programming with Applications in Optimal Control*. Springer International Publishing.

Lo, C.-C. and X.-J. Wang (2006). "Cortico-basal ganglia circuit mechanism for a decision threshold in reaction time tasks". *Nature Neuroscience*. 9(7): 956–963.

Lorica, B. (2017). *Practical Applications of Reinforcement Learning in Industry: An Overview of Commercial and Industrial Applications of Reinforcement Learning*.

Luo, B., H.-N. Wu, T. Huang, and D. Liu (2014). "Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design". *Automatica*. 50(12): 3281–3290.

Luo, B., D. Liu, T. Huang, and D. Wang (2016). "Model-free optimal tracking control via critic-only Q-learning". *IEEE Transactions on Neural Networks and Learning Systems*. 27(10): 2134–2144.

Mahadevan, S. (1996). "Average reward reinforcement learning: Foundations, algorithms, and empirical results". *Machine Learning*. 22(1–3): 159–195.

Marino, R. and P. Tomei (2003). "Output regulation for linear systems via adaptive internal model". *IEEE Transactions on Automatic Control*. 48(12): 2199–2202.

Meyn, S. P. and R. L. Tweedie (1993a). "Stability of Markovian processes II: Continuous-time processes and sampled chains". *Advances in Applied Probability*. 25(3): 487–517.

Meyn, S. P. and R. L. Tweedie (1993b). "Stability of Markovian processes III: Foster-Lyapunov criteria for continuous-time processes". *Advances in Applied Probability*. 25(3): 518–548.

Minsky, M. L. (1954). "Theory of neural-analog reinforcement systems and its application to the brain model problem". *PhD thesis*. Princeton University.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). "Human-level control through deep reinforcement learning". *Nature*. 518(7540): 529–533.

Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu (2016). "Asynchronous methods for deep reinforcement learning". In: *Proceedings of the 33rd International Conference on Machine Learning*. New York, New York, USA. 1928–1937.

Modares, H. and F. L. Lewis (2014). "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning". *IEEE Transactions on Automatic Control*. 59(11): 3051–3056.

Munos, R. (2000). "A study of reinforcement learning in the continuous case by the means of viscosity solutions". *Machine Learning*. 40(3): 265–299.

Murray, J. J., C. J. Cox, G. G. Lendaris, and R. Saeks (2002). "Adaptive dynamic programming". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 32(2): 140–153.

Nedić, A. and D. P. Bertsekas (2003). "Least squares policy evaluation algorithms with linear function approximation". *Discrete Event Dynamic Systems*. 13(1–2): 79–110.

Ni, Z., H. He, and J. Wen (2013). "Adaptive learning in tracking control based on the dual critic network design". *IEEE Transactions on Neural Networks and Learning Systems*. 24(6): 913–928.

Odekunle, A., W. Gao, M. Davari, and Z. P. Jiang (2020). "Reinforcement learning and non-zero-sum game output regulation for multi-player linear uncertain systems". *Automatica*. 112(2): 108672.

Oncu, S., J. Ploeg, N. van de Wouw, and H. Nijmeijer (2014). "Cooperative adaptive cruise control: Network-aware analysis of string stability". *IEEE Transactions on Intelligent Transportation Systems*. 15(4): 1527–1537.

Pang, B. and Z. P. Jiang (2020). "Adaptive optimal control of linear periodic systems: An off-policy value iteration approach". *IEEE Trans. Automatic Control*. DOI: 10.1109/TAC.2020.2987313.

Pang, B., T. Bian, and Z. P. Jiang (2019). "Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete-time systems". *Control Theory and Technology*. 17(1): 73–84.

Pang, B., Z. P. Jiang, and I. Mareels (2020). "Reinforcement learning for adaptive optimal control of continuous-time linear periodic systems". *Automatica*. 118: 109035.

Park, J. and I. W. Sandberg (1991). "Universal approximation using radial-basis-function networks". *Neural Computation*. 3(2): 246–257.

Pekny, S. E., J. Izawa, and R. Shadmehr (2015). "Reward-dependent modulation of movement variability". *The Journal of Neuroscience*. 35(9): 4015–4024.

Pliska, S. R. (1986). "A stochastic calculus model of continuous trading: Optimal portfolios". *Mathematics of Operations Research*. 11(2): 371–382.

Praly, L. and Y. Wang (1996). "Stabilization in spite of matched unmodeled dynamics and an equivalent definition of input-to-state stability". *Mathematics of Control, Signals and Systems*. 9(1): 1–33.

Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ: John Wiley & Sons, Inc.

Rahnama, A. and P. J. Antsaklis (2019). "Learning-based event-triggered control for synchronization of passive multi-agent systems under attack". *IEEE Transactions on Automatic Control*. 65(10): 4170–4185.

Recht, B. (2019). "A tour of reinforcement learning: The view from continuous control". *Annual Review of Control, Robotics, and Autonomous Systems*. 2(1): 253–279.

Renart, A. and C. K. Machens (2014). "Variability in neural activity and behavior". *Current Opinion in Neurobiology*. 25: 211–220.

Rizvi, S. A. A. and Z. Lin (2019). "Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback". *IEEE Transactions on Cybernetics*. Accepted.

Robbins, H. and S. Monro (1951). "A stochastic approximation method". *The Annals of Mathematical Statistics*. 22(3): 400–407.

Ross, S. A. (1976). "The arbitrage theory of capital asset pricing". *Journal of Economic Theory*. 13(3): 341–360.

Russell, S. J. and P. Norvig (2010). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education.

Salvucci, D. D. and R. Gray (2004). "A two-point visual control model of steering". *Perception*. 33(10): 1233–1248.

Sandell, N. R. (1974). "On Newton's method for Riccati equation solution". *IEEE Transactions on Automatic Control*. 19(3): 254–255.

Saridis, G. N. and C.-S. G. Lee (1979). "An approximation theory of optimal control for trainable manipulators". *IEEE Transactions on Systems, Man and Cybernetics*. 9(3): 152–159.

Schmidhuber, J. (2015). "Deep learning in neural networks: An overview". *Neural Networks*. 61(Jan.): 85–117.

Schwartz, A. (1993). "A reinforcement learning method for maximizing undiscounted rewards". In: *Proceedings of the 10th International Conference on Machine Learning*. Amherst, MA. 298–305.

Seiler, P., A. Pant, and K. Hedrick (2004). "Disturbance propagation in vehicle strings". *IEEE Transactions on Automatic Control*. 49(10): 1835–1842.

Serrani, A., A. Isidori, and L. Marconi (2001). "Semiglobal nonlinear output regulation with adaptive internal model". *IEEE Transactions on Automatic Control*. 46(8): 1178–1194.

Shadmehr, R. and F. A. Mussa-Ivaldi (1994). "Adaptive representation of dynamics during learning of a motor task". *The Journal of Neuroscience*. 14(5): 3208–3224.

Shadmehr, R. and S. Mussa-Ivaldi (2012). *Biological Learning and Control: How the Brain Builds Representations, Predicts Events, and Makes Decisions*. Cambridge, MA: The MIT Press.

Shayman, M. (1986). "Phase portrait of the matrix Riccati equation". *SIAM Journal on Control and Optimization*. 24(1): 1–65.

Shladover, S., D. Su, and X.-Y. Lu (2012). "Impacts of cooperative adaptive cruise control on freeway traffic flow". *Transportation Research Record*. 2324: 63–70.

Silver, D. (2015). "Reinforcement Learning (COMPM050/COMPGI13)". Lecture Notes, University College London, Computer Science Department, London.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis (2016). "Mastering the game of Go with deep neural networks and tree search". *Nature*. 529(7587): 484–489.

Silver, D., J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis (2017). "Mastering the game of Go without human knowledge". *Nature*. 550(7676): 354–359.

Song, R., F. L. Lewis, Q. Wei, H. Zhang, Z. P. Jiang, and D. Levine (2015). "Multiple actor-critic structures for continuous-time optimal control using input–output data". *IEEE Transactions on Neural Networks and Learning Systems*. 26(4): 851–865.

Sontag, E. D. (2008). "Input to state stability: Basic concepts and results". In: *Nonlinear and Optimal Control Theory*. Ed. by P. Nistri and G. Stefani. Berlin, Heidelberg: Springer. 163–220.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken, NJ: John Wiley & Sons, Inc.

Su, Y. and J. Huang (2012). "Cooperative output regulation of linear multi-agent systems". *IEEE Transactions on Automatic Control*. 57(4): 1062–1066.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd Ed. Cambridge, MA: The MIT Press.

Sutton, R. S., A. G. Barto, and R. J. Williams (1992). "Reinforcement learning is direct adaptive optimal control". *IEEE Control Systems*. 12(2): 19–22.

Sutton, R. S., D. Precup, and S. P. Singh (1999). "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". *Artificial Intelligence*. 112(1–2): 181–211.

Sutton, R. S., D. A. McAllester, S. P. Singh, and Y. Mansour (2000). "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in Neural Information Processing Systems 12*. Vol. 12. MIT Press. 1057–1063.

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers.

Tanner, H. G., G. J. Pappas, and V. Kumar (2004). "Leader-to-formation stability". *IEEE Transactions on Robotics and Automation*. 20(3): 443–455.

Tao, G. (2003). *Adaptive Control Design and Analysis*. Hoboken, NJ: John Wiley & Sons, Inc.

Theodorou, E. A., J. Buchli, and S. Schaal (2010). "A generalized path integral control approach to reinforcement learning". *Journal of Machine Learning Research*. 11(Dec.): 3137–3181.

Thrun, S. B. (1992). "Efficient exploration in reinforcement learning". *Tech. rep.* No. CMU-CS-92-102. Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

Todorov, E. (2004). "Optimality principles in sensorimotor control". *Nature Neuroscience*. 7(9): 907–915.

Todorov, E. (2005). "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system". *Neural Computation*. 17(5): 1084–1108.

Todorov, E. and M. I. Jordan (2002). "Optimal feedback control as a theory of motor coordination". *Nature Neuroscience*. 5(11): 1226–1235.

Tsitsiklis, J. N. (1994). "Asynchronous stochastic approximation and Q-learning". *Machine Learning*. 16(3): 185–202.

Tsitsiklis, J. N. and B. Van Roy (1997). "An analysis of temporal-difference learning with function approximation". *IEEE Transactions on Automatic Control*. 42(5): 674–690.

Tsitsiklis, J. N. and B. Van Roy (1999). "Average cost temporal-difference learning". *Automatica*. 35(11): 1799–1808.

Tsitsiklis, J. N. and B. Van Roy (2002). "On average versus discounted reward temporal-difference learning". *Machine Learning*. 49(2–3): 179–191.

Uno, Y., M. Kawato, and R. Suzuki (1989). "Formation and control of optimal trajectory in human multijoint arm movement". *Biological Cybernetics*. 61(2): 89–101.

Vamvoudakis, K. G. (2017). "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach". *Systems & Control Letters*. 100: 14–20.

Vamvoudakis, K. G. and H. Ferraz (2018). "Model-free event-triggered control algorithms for continuous-time linear systems with optimal performance". *Systems & Control Letters*. 87: 412–420.

van Hasselt, H. (2012). "Reinforcement learning in continuous state and action spaces". In: *Reinforcement Learning: State-of-the-Art*. Ed. by M. Wiering and M. Otterlo. Vol. 12. *Adaptation, Learning, and Optimization*. Berlin, Heidelberg: Springer. Chap. 7. 207–251.

van Hasselt, H. and M. A. Wiering (2007). "Reinforcement learning in continuous action spaces". In: *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. 272–279.

van der Schaft, A. J. (2017). *L_2-Gain and Passivity Techniques in Nonlinear Control*. 3rd Ed. Springer International Publishing.

van der Schaft, A. J. and H. Schumacher (2000). *An Introduction to Hybrid Dynamical Systems*. Springer.

Vaswani, P. A., L. Shmuelof, A. M. Haith, R. J. Delnicki, V. S. Huang, P. Mazzoni, R. Shadmehr, and J. W. Krakauer (2015). "Persistent residual errors in motor adaptation tasks: Reversion to baseline and exploratory escape". *The Journal of Neuroscience*. 35(17): 6969–6977.

Vrabie, D., O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis (2009). "Adaptive optimal control for continuous-time linear systems based on policy iteration". *Automatica*. 45(2): 477–484.

Vrabie, D., K. G. Vamvoudakis, and F. L. Lewis (2013). *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. London, UK: Institution of Engineering and Technology.

Wang, D. and C. Mu (2019). *Adaptive Critic Control with Robust Stabilization for Uncertain Nonlinear Systems*. Singapore: Springer.

Wang, D., H. He, and D. Liu (2017). "Adaptive critic nonlinear robust control: A survey". *IEEE Transactions on Cybernetics*. 47(10): 3429–3451.

Wang, F.-Y., H. Zhang, and D. Liu (2009). "Adaptive dynamic programming: An introduction". *IEEE Computational Intelligence Magazine*. 4(2): 39–47.

Wang, J. X., Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, and M. Botvinick (2018). "Prefrontal cortex as a meta-reinforcement learning system". *Nature Neuroscience*. 21(6): 860–868.

Wang, X., Y. Hong, J. Huang, and Z. P. Jiang (2010). "A distributed control approach to a robust output regulation problem for multi-agent linear systems". *IEEE Transactions on Automatic Control*. 55(12): 2891–2895.

Werbos, P. (1968). "The elements of intelligence". *Cybernetica (Namur)*. 11(3): 131.

Werbos, P. (2013). "Reinforcement learning and approximate dynamic programming (RLADP) – Foundations, common misconceptions and the challenges ahead". In: *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Ed. by F. L. Lewis and D. Liu. Hoboken, NJ: Wiley. 3–30.

Werbos, P. J. (2014). "From ADP to the brain: Foundations, roadmap, challenges and research priorities". In: *Proceedings of the 2014 International Joint Conference on Neural Networks*. Beijing, China. 107–111.

Werbos, P. J. (2018). "AI intelligence for the grid 16 years later: Progress, challenges and lessons for other sectors". In: *Proceedings of the 2018 International Joint Conference on Neural Networks*. 1–8.

Willems, J. L. (1971). "Least squares stationary optimal control and the algebraic Riccati equation". *IEEE Transactions on Automatic Control*. 16(6): 621–634.

Wise, R. A. (2004). "Dopamine, learning and motivation". *Nature Reviews Neuroscience*. 5(June): 483–494.

Wolpert, D. M. and Z. Ghahramani (2000). "Computational principles of movement neuroscience". *Nature Neuroscience*. 3: 1212–1217.

Wolpert, D. M., J. Diedrichsen, and J. R. Flanagan (2011). "Principles of sensorimotor learning". *Nature Reviews Neuroscience*. 12(12): 739–751.

Wonham, W. (1967). "Optimal stationary control of a linear system with state-dependent noise". *SIAM Journal on Control*. 5(3): 486–500.

Wu, H. G., Y. R. Miyamoto, L. N. G. Castro, B. P. Olveczky, and M. A. Smith (2014). "Temporal structure of motor variability is dynamically regulated and predicts motor learning ability". *Nature Neuroscience*. 17(2): 312–321.

Yang, Y., L. Wang, H. Modares, D. Ding, Y. Yin, and D. Wunsch (2019). "Data-driven integral reinforcement learning for continuous-time non-zero-sum games". *IEEE Access*. 7(1): 82901–82912.

Yu, H. and D. P. Bertsekas (2009). "Convergence results for some temporal difference methods based on least squares". *IEEE Transactions on Automatic Control*. 54(7): 1515–1531.

Zakai, M. (1969). "A Lyapunov criterion for the existence of stationary probability distributions for systems perturbed by noise". *SIAM Journal on Control*. 7(3): 390–397.

Zhou, S.-H., J. Fong, V. Crocher, Y. Tan, D. Oetomo, and I. M. Y. Mareels (2016). "Learning control in robot-assisted rehabilitation of motor skills—A review". *Journal of Control and Decision*. 3(1): 19–43.