Learning-Based Adaptive Optimal Control of Linear Time-Delay Systems: A Policy Iteration Approach

Leilei Cui, Bo Pang, and Zhong-Ping Jiang, Fellow, IEEE

Abstract—This article studies the adaptive optimal control problem for a class of linear time-delay systems described by delay differential equations. A crucial strategy is to take advantage of recent developments in reinforcement learning and adaptive dynamic programming and develop novel methods to learn adaptive optimal controllers from finite samples of input and state data. In this article, the data-driven policy iteration (PI) is proposed to solve the infinite-dimensional algebraic Riccati equation iteratively in the absence of exact model knowledge. Interestingly, the proposed recursive PI algorithm is new in the present context of continuous-time time-delay systems, even when the model knowledge is assumed known. The efficacy of the proposed learning-based control methods is validated by means of practical applications arising from metal cutting and autonomous driving.

Index Terms—Adaptive dynamic programming (ADP), linear time-delay systems, optimal control, policy iteration (PI).

I. INTRODUCTION

Time-delay systems are ubiquitous in many branches of science and engineering; see the books [1], [2], [3] for many references and examples. Recently, many theoretical results are developed for time-delay systems, such as input-to-state stability [4], robust $H_\infty$ control [5], and stability analysis of systems with time-varying delay [6], [7]. Examples of time-delay systems are in transportation [8], [9], biological motor control [10], and multiagent systems [11], [12]. It is, thus, not surprising that the optimal control problem of time-delay systems has been a fundamentally important, yet challenging, research topic in control theory for several decades. For instance, Eller et al. [13] and Ross et al. [14], [15] proposed solutions to the finite-horizon and infinite-horizon linear quadratic (LQ) optimal control problems of linear time-delay systems, respectively. In these papers, the certain infinite-dimensional Riccati equations have to be solved. For this problem, many numerical algorithms have been developed [16], [17], [18]. However, an accurate model of the system is required for these algorithms, and in reality, it is difficult to derive an exact model due to the complexity of the system and inevitable system uncertainties. Therefore, developing a model-free optimal control approach for time-delay systems is a timely research topic of both theoretical importance and practical relevance. Recent progresses and successes in reinforcement learning (RL) provide an opportunity to advance the state of the art in the area of adaptive optimal control of time-delay systems.

RL is an important branch of machine learning and is aimed at maximizing (or minimizing) the cumulative reward (or cost) through agent-environment interactions. Traditional RL has some fundamental limitations. For example, it often assumes that the environment is depicted by Markov decision processes or discrete-time systems with finite state-action space. Often, the stability aspect of the learned controller by RL is not guaranteed. For many systems described by differential equations, such as autonomous vehicles (AVs) and quadrupedal robots, the state and action spaces are infinite and the stability of the controller generated by an RL algorithm is negligible. Therefore, for these safety-critical engineering systems, conventional RL is not directly applicable to learning stable optimal controllers from data, which has motivated the development of adaptive dynamic programming (ADP) [19], [20]. In contrast with conventional RL, the purpose of continuous-time ADP is addressing decision-making problems for dynamical systems described by differential equations, of which both the state and action spaces are continuous. It is theoretically shown that at each iteration of ADP, a stable suboptimal controller with improved performance is obtained. Besides, the sequence of these suboptimal controllers converges to the optimal one [19].

Recently, ADP techniques are developed for various important classes of linear/nonlinear/periodic dynamical systems and for optimal stabilization, tracking and output regulation problems [19], [21], [22], [23], [24], [25]. However, a systematic ADP approach to adaptive optimal control of continuous-time time-delay systems is lacking, due to the infinite-dimensional nature of these systems. In [26], although the model-free data-driven control for continuous-time time-delay systems is studied, discretization and/or linearization techniques are applied to transfer the time-delay system to a finite-dimensional delay-free system with augmented states, which leads to an approximate model. In [9], [27], [28], [29], [30], [31] and, [32], ADP for discrete-time systems with time delays is studied. Due to the finite dimensionality of discrete-time systems with time delays, these proposed ADP methods are not applicable to continuous-time-delay systems. In [33] and [34], ADP technique is applied for both linear and nonlinear systems with time delays, but the resulting controller cannot achieve optimality [33, Remark 9.1]. Technically, there are several obstacles in the generalization of ADP to time-delay systems. First, for an infinite-dimensional system, optimality properties are hard to analyze, because the corresponding algebraic Riccati equation (ARE) is complex partial differential equations (PDEs). Second, stability analysis and controller design for a time-delay system are much more challenging than finite-dimensional systems. Therefore, the model-free optimal control for a continuous-time time-delay system remains an open problem.

In this article, in the absence of the precise knowledge of system dynamics, a novel data-driven policy iteration (PI) approach for continuous-time linear time-delay systems is proposed based on ADP. The contributions of this article are as follows. First, inspired by Kleinman’s model-based PI algorithm for delay-free linear systems [35], a new model-based PI algorithm is proposed for a class of linear time-delay systems. Given an initial admissible controller, both the

Manuscript received 16 October 2022; revised 11 April 2023; accepted 22 April 2023. Date of publication 8 May 2023; date of current version 29 December 2023. This work was supported in part by the National Science Foundation under Grant EPCN-1903781 and Grant CNS-2148309. Recommended by Associate Editor M. E. Broucke. (Corresponding author: Leilei Cui.)

The authors are with the Control and Networks Lab, Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mails: l.cui@nyu.edu; b.pang@nyu.edu; zjiang@nyu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2023.3273786.

Digital Object Identifier 10.1109/TAC.2023.3273786
stability of the updated suboptimal controller and the convergence of the algorithm to the (unknown) optimal controller are proved theoretically. It is worth pointing out that due to the infinite dimensionality, both the value function and the control law are functional of the system’s state, which in consequence increases the difficulty to design the PI algorithm. Second, based on the model-based PI, this article contributes a data-driven PI approach to adaptive optimal controller design using only the data measured along the trajectories of the system.

The rest of this article is organized as follows. Section II introduces the class of linear time-delay systems and formulates the adaptive optimal control problem to be addressed in the article. Section III proposes a model-based PI approach to iteratively solve the LQ optimal control problem for linear time-delay systems. In Section IV, a data-driven PI approach is proposed, and the convergence property of the algorithm is analyzed. Section V illustrates the proposed data-driven PI approach by means of two practical examples. Finally, Section VI concludes this article.

Notations: In this article,  is the set of (nonnegative) real numbers and  denotes the set of positive integers.  denotes the Euclidean norm of a vector or Frobenius norm of a matrix.  denotes the supremum norm of a function.  denotes the class of absolutely continuous functions.  denotes the derivative of the function with respect to the direct sum.  denote the space of measurable functions for which the th power of the Euclidean norm is Lebesgue integrable. 

Problem Formulation and Preliminaries

A. Problem Formulation

Consider a linear time-delay system

\[ \dot{x}(t) = Ax(t) + A_d x(t - \tau) + Bu(t) \]  

(1)

where  denotes the constant delay of the system and is assumed to be known,  and  are known matrices.  and  are unknown constant matrices. Let  denote a segment of the state trajectory in the interval . Due to the infinite dimensionality of system (1), the state of the system is

\[ z(t) = [x^T(t), x^T(\cdot)]^T \in M_2. \]

Define the linear operators  and  as

\[ A(x(t)) = \begin{bmatrix} Ax(t) + A_d x(t - \tau) \\ \frac{d}{dt} u(t) \end{bmatrix}, \]

and

\[ B(u(t)) = \begin{bmatrix} Bu(t) \\ 0 \end{bmatrix}. \]

Then, according to [36, Th. 2.4.6], (1) is rewritten as

\[ \dot{z}(t) = A z(t) + B u(t) \]

(2)

with the domain of  given by . Let  denote the initial state of the system (2). The quadratic performance index is defined as

\[ J(x_0, u) = \int_0^\infty x^T(t) Q x(t) + u(t)^T R u(t) \, dt \]

(3)

where  and  are admissible for system (1) with respect to (3), if system (1) with  is globally asymptotically stable at the origin [37, Definition 1.1], and the performance index (3) is finite for all  in (2).

Definition 1: A control policy  is admissible for system (1) with output  with the entry for system (1) to be known, and we have  and  are unknown constant matrices. Let  denote the th power of the Euclidean norm. The class of continuous bounded linear operators from  to  is denoted by  and is defined in [14, Definition 5.2.1] and checked by [36, Th. 5.2.12].

Remark 1: Assumption 1 is a standard prerequisite for LQ optimal control of system (1) to ensure the existence of a unique stabilizing solution [36, 38].

Given the aforementioned assumption, the problem to be studied in this article can be formulated as follows.

Problem 1: Given an initial admissible controller  with , design a PI-based ADP algorithm to approximate the optimal controller in (4) using only the input-state data measured along the trajectories of the system.

B. Optimality and Stability

For a linear system without time delay, i.e.,  and  in (1), one can find the optimal solution by solving the ARE as discovered by Kalman [39]. Correspondingly, for system (1), the optimal solution is stated as follows.

Lemma I (see [14, 40]):

For system (1) with Assumption 1

\[ u^*(x_t) = -R^{-1}B^T P_0 x_t - \int_{\tau}^0 R^{-1}B^T P_1(\xi) x(\xi) \, d\xi \]

(4)

is the optimal controller minimizing (3), and the corresponding minimal value functional is

\[ V^*(x_0) = x^T(0) P_0^* x(0) + 2 x^T(0) \int_{\tau}^0 P_1^*(\xi) x(\xi) \, d\xi \]

(5)

where  are the unique stabilizing solution to the following PDEs:

\[ \frac{d P_0^*}{d\theta} = (A^T - P_0^* B R^{-1} B^T) P_0^* + P_1^* (0) + Q \]

\[ \frac{d P_1^*}{d\theta} = (A^T - P_0^* B R^{-1} B^T) P_1^* + P_2^* (0, 0) \]

\[ \frac{d P_2^*}{d\theta} = -P_1^* (\xi) B R^{-1} B^T P_1^* (0) - P_1^* (\xi) \]

\[ P_1^*(\tau) = P_0^* A_d, \quad P_2^*(\tau, \theta) = A_d^T P_1^* (\theta). \]

(6)

By [36, Th. 6.2.7], the time-delay system (1) in closed-loop with  is exponentially stable at the origin.
According to Lemma 1, the optimal controller is obtained by solving (6). Since (6) is nonlinear with respect to $P_0^i$, $P_1^i$, and $P_2^i$, it is difficult to solve it directly. In this section, a model-based PI algorithm is proposed to simplify the process of solving (6).

Given an admissible controller $u_i(x_i) = -K_{0,1}x_i(t) - \int_{0}^{t} K_{1,1}(\theta)x_i(\theta)d\theta$, the model-based PI algorithm for system (1) is proposed as follows.

1) Policy Evaluation: For $i \in \mathbb{N}_+$, and $\xi, \theta \in [-\tau, 0]$, calculate $P_{0,i} = P_{0}^i$, $P_{1,i}(\theta)$, and $P_{2,i}(\theta, \xi) = P_{2,i}(\theta, \xi, t)$ by solving the following PDEs:

$$ dP_{1,i}(\theta) = A_i^T P_{1,i}(\theta) - P_{1,i}B K_{1,1}(\theta) + K_{2,i} R K_{1,1}(\theta) + P_{2,i}(0, 0) $$

$$ \partial_\xi P_{2,i}(\xi, \theta) + \partial_\theta P_{2,i}(\xi, \theta) = K_{1,1}^T(\xi) K_{1,1}(\theta) - 2 K_{1,1}(\xi) B^T P_{1,i}(\theta) $$

$$ P_{1,i}(-\tau) = P_{0,i}, \quad A_i = (A - B K_{0,1}) $$

(7)

where $A_i = (A - B K_{0,1})$ and $Q_i = Q + K_{0,1}^T R K_{0,1}$.

2) Policy Improvement: Update the policy $u_{i+1}$ by

$$ u_{i+1}(x_i) = -R^{-1}B^T P_{0,i} x(t) - \int_{0}^{\tau} R^{-1}B^T P_{1,i}(\theta) x_i(\theta)d\theta $$

(8)

The policy evaluation step calculates the value function $V_i(x_0)$ for the controller $u_i$, which is expressed as

$$ V_i(x_0) = x^T(0) P_{0,i} x(0) + 2x^T(0) \int_{0}^{\tau} P_{1,i}(\theta) x_0(\theta)d\theta $$

$$ + \int_{0}^{\tau} \int_{0}^{-\tau} x_i(\xi) P_{2,i}(\xi, \theta) x_0(\theta)d\xi d\theta $$

(9)

By policy improvement, the value function is monotonically decreasing ($V_i(x_0) \leq V_{i+1}(x_0)$), and converges to the optimal value function $V^*(x_0)$. Correspondingly, $P_{0,i}$, $P_{1,i}(\theta)$, and $P_{2,i}(\xi, \theta)$ converge to the optimal solutions $P_0^*, P_1^*(\theta)$, and $P_2^*(\xi, \theta)$, respectively. The convergence of the model-based PI algorithm is rigorously demonstrated in Theorem 1. Before stating Theorem 1, we first introduce Lemma 2, which is instrumental for the proof of Theorem 1. By Lemma 2, for a linear controller $u_L$, if $J(x_0, u_L)$ is finite, the closed-loop system consisting of (1) and $u_L$ is globally exponentially stable.

**Lemma 2: Consider system (1) with Assumption 1. If a linear controller $u_L(x_i) = -K x_i(t)$ satisfies $J(x_0, u_L) < \infty$ for any $x_0 \in D$, where $K \in L(M_2, \mathbb{R}^{n})$, then the closed-loop system with $u_L$ is globally exponentially stable at the origin.**

**Proof:** The details of the proof are in [41, Lemma 2].

**Theorem 1:** Given an admissible control $u_i(x_i)$, for $P_{0,i}$, $P_{1,i}(\theta)$, $P_{2,i}(\xi, \theta)$, and $u_{i+1}(x_i)$ obtained by solving (7) and (8), and for all $i \in \mathbb{N}_+$, the following properties hold:

1) $u_{i+1}(x_i)$ is admissible;
2) $V^*(x_0) \leq V_{i+1}(x_0) \leq V_i(x_0)$;
3) $V_i(x_0)$ and $x_i(t)$ converge to $V^*(x_0)$ and $u^*(x_i)$.

**Proof:** Along the trajectories of (1) driven by $u_i$, where $u$ without subscript stands for an arbitrary input, $V_i(x_i)$ is

$$ V_i(x_i) = -x^T Q x - u_i^T R u_i + 2 u_i^T R u_i - 2 u_i^T R u_{i+1} $$

(10)

The detailed derivation of (10) is in [41, eq. (10)].

Property 1) is proved by induction. When $i = 1$, the admissibility of $u_1(x_1)$ is given. For $i > 1$, assume that $u_i$ is admissible. When system (1) is driven by $u_i$, by (10), the expression of $V_i(x_i)$ is

$$ V_i(x_i) = -x^T Q x - u_i^T R u_i $$

Following the fact that $u_i$ is admissible and integrating (11) from 0 to $\infty$, we have

$$ V_i(x_0) = \int_{0}^{\infty} x^T(t)Q x(t) + u_i^T(t) R u_i(t) dt $$

$$ = J(x_0, u_i) < \infty $$

(12)

By (10), along the trajectories of (1) driven by $u_{i+1}$,

$$ V_i(x_i) = -x^T Q x - u_{i+1}^T R u_{i+1} - (u_{i+1} - u_i)^T R (u_{i+1} - u_i) $$

(13)

Integrating both sides of (13) from 0 to $\infty$ yields

$$ J(x_0, u_{i+1}) = V_i(x_0) - V_i(x_\infty) $$

$$ - \int_{0}^{\infty} (u_{i+1} - u_i)^T R (u_{i+1} - u_i) dt \leq V_i(x_0) < \infty $$

(14)

It follows from (14) and Lemma 2 that $u_{i+1}$ is a globally and exponentially stabilizing controller. By Definition 1, $u_{i+1}$ is admissible. Via induction, $u_i$ is admissible for any $i \in \mathbb{N}_+$.

Along the trajectories of system (1) driven by $u_{i+1}$, by (10)

$$ V_i(x_i) = -x^T Q x - u_{i+1}^T R u_{i+1} $$

(15)

Since $u_{i+1}$ is admissible, integrating (15) from 0 to $\infty$ yields $V_{i+1}(x_0) = J(x_0, u_{i+1})$. Hence, $V_{i+1}(x_0) \leq V_i(x_0)$ is obtained by (14). Furthermore, since $V^*(x_0) = J(x_0, u^*)$ is the minimal value functional by Lemma 1, for any $i \in \mathbb{N}_+$, $V^*(x_0) \leq V_i(x_0)$. Therefore, the proof of (2) is completed.

Define $P_i \in L(M_2)$, such that for any $z_0$, $P_i z_0$ is

$$ P_i z_0 = \begin{bmatrix} P_{0,i} x(0) + \int_{0}^{\tau} P_{1,i}(\theta) x(\theta)d\theta \\ \int_{0}^{\tau} P_{2,i}(\theta) x(\theta)d\theta + P_{1,i}^T(x_0) \end{bmatrix} $$

(16)

It is easy to check that $P_i$ is symmetric [42, Ch. 6], and nonnegative [42, Definition 6.3.1], and $V_i(x_0) = (z_0, P_i z_0)$. Furthermore, according to statement 2), for any $i \in \mathbb{N}_+$, $P_i \leq P_{i+1} \leq P_i$. According to [42, Th. 6.3.2], there exists $P_P = P^T \geq 0$, such that for all $z_0 \in M_2$, we have

$$ \lim_{i \to \infty} P_i z_0 = P_P z_0 $$

(17)

Therefore, $P_{0,i}$, $P_{1,i}(\theta)$, and $P_{2,i}(\xi, \theta)$ pointwisely converge to $P_{0,p}$, $P_{1,p}(\theta)$, and $P_{2,p}(\xi, \theta)$, respectively. When $P_i$ converges, $P_{0,p}$, $P_{1,p}(\theta)$, and $P_{2,p}(\xi, \theta)$ satisfy (7) with $i$ replaced by $p$. $K_{0,p}$ and $K_{1,p}$ converge to $K_{0,p}$ and $K_{1,p}$ by the policy improvement step (8), $K_{0,p}$ and $K_{1,p}$ satisfy

$$ K_{0,p} = R^{-1} B^T P_{0,p}, \quad K_{1,p}(\theta) = R^{-1} B^T P_{1,p}(\theta) $$

(18)

Substituting (18) into (7) with $i$ replaced by $p$, it is seen that $P_{0,p}$, $P_{1,p}(\theta)$, and $P_{2,p}(\xi, \theta)$ solve the PDEs (6). Due to the uniqueness of the solution to (6), $P_{0,i}$, $P_{1,i}(\theta)$, and $P_{2,i}(\xi, \theta)$ pointwisely converge to $P_{0,p}$, $P_{1,p}(\theta)$, and $P_{2,p}(\xi, \theta)$. Since both $P_{1,i}(\theta)$ and $P_{2,i}(\xi, \theta)$ are continuously differentiable, $P_{1,i}(\theta) : i \in \mathbb{N}_+$ and $P_{2,i}(\xi, \theta) : i \in \mathbb{N}_+$ are equicontinuous, which leads to the uniform convergence by [43, Ch. 4, Th. 16]. Hence, 3) is proved.
an accurate model. Therefore, in the following section, a data-driven PI algorithm is proposed to approximate the optimal solution.

**Remark 2**: When $A_2 = 0$, (1) is degraded to the normal delay-free systems. According to (7) and (8), we can see that $P_{1,i}(\theta) = 0$, $P_{2,i}(\xi, \theta) = 0$, and $K_{1,i}(\theta) = 0$. As a consequence, (7) and (8) are the same as the model-based PI method in [35]. Therefore, the proposed model-based PI algorithm is a generalization of celebrated Kleinman's PI to linear time-delay systems.

**Remark 3**: In [18], the model-based PI is developed for infinite-dimensional linear systems in the Hilbert space. Although system (1) is one of the infinite-dimensional systems, the concrete expression of PI for linear time-delay systems is not given in [18], and as a consequence, the PI developed in [18] cannot be directly applied to solve the PDEs (6). In this article, the concrete expression of PI is constructed in (7) and (8), which is one of the major contributions in this article. Besides, it can be checked that at each iteration, $P_i$ defined in (16) satisfies the PI update equations in [18], which is another way to prove the validity of the proposed PI theoretically.

**Remark 4**: As shown in [18], the convergence rate of PI algorithm in the Hilbert space is quadratic, and therefore, the proposed model-based PI for system (1) has the same quadratic convergence rate.

### IV. DATA-DRIVEN PI

The purpose of this section is to propose a corresponding data-driven PI algorithm that does not require the accurate knowledge of system (1) to solve Problem 1. The input-state trajectories of system (1) is required for the data-driven PI. In other words, the continuous-time trajectories of $x(t)$ and $u(t)$ sampled from system (1) within the interval $[t_k, t_{L+1}]$ is applied to train the control policy. From the RL perspective, $u$ is named behavior/exploratory policy.

Define $v_i(t) = u(t) - u_i(x_i)$, where $u_i(x_i)$ is the value of the control policy $u_i$ calculated along the sampled trajectory. By (10), along the trajectories of system (1) driven by the behavior/exploratory policy $u_i$,

$$
\dot{V}_i(x_i) = -x^\top Q x - u_i^\top R u_i - 2u_i^\top R u_i \ \ \ \ (19)
$$

Let $[t_k, t_{k+1}]$ denote the $k$th segment of the sampling interval $[t_1, t_{L+1}]$. Integrating both sides of (19) from $t_k$ to $t_{k+1}$ yields

$$
V_i(x_{k+1}) - V_i(x_k) = \int_{t_k}^{t_{k+1}} -x^\top Q x - u_i^\top R u_i - 2u_i^\top R u_i \ dt. \ \ \ \ (20)
$$

Plugging the expressions of $u_i$ in (8) and $V_i$ in (9) into (20) yields

$$
\begin{align*}
& \left[ x^\top (t) P_{0,i} x(t) + 2x^\top (t) \int_0^t P_{1,i}(\theta)x(\theta) d\theta \right]_{t=k+1}^{t=k} \\
& + \int_{t_k}^{t_{k+1}} \int_0^t x^\top (\xi) P_{2,i}(\xi, \theta)x(\theta) d\xi d\theta \\
& - 2 \int_{t_k}^{t_{k+1}} \left( x^\top (t) K_{0,i+1}(\theta) + \int_{-\tau}^0 x^\top (\xi) K_{1,i+1}(\theta)(\xi) d\xi \right) R u_i(t) d\theta \\
& = - \int_{t_k}^{t_{k+1}} x^\top (t) Q x(t) + u_i^\top (t) R u_i(t) \ dt.
\end{align*}
$$

As seen in (7) and (8), $K_{1,i}(\theta)$ and $P_{2,i}(\xi, \theta)$ are continuous functions defined over the set $[-\tau, 0]$ $([-\tau, 0]^2)$. Next, we use the linear combinations of basis functions to approximate these continuous functions, such that only the weighting matrices of the basis functions should be determined for the function approximation. Let $\Phi(\theta)$, $\Lambda(\xi, \theta)$, and $\Psi(\xi, \theta)$ denote the $N$-dimensional vectors of linearly independent basis functions. To simplify the notation, we choose the same number of basis functions for $\Phi$, $\Lambda$, and $\Psi$. According to the approximation theory [44], the following equations hold:

$$
\begin{align*}
\text{vec}(P_{0,i}) &= W_{0,i}, \quad \text{vec}(P_{1,i}(\theta)) = W_{1,i}^N \Phi(\theta) + e_{\Phi,i}(\theta) \\
\text{diag}(P_{2,i}(\xi, \theta)) &= \Psi(\xi, \theta) + e_{\Psi,i}(\xi, \theta) \\
\text{vec}(P_{2,i}(\xi, \theta)) &= W_{2,i}^N \Lambda(\xi, \theta) + e_{\Lambda,i}(\xi, \theta)
\end{align*}
$$

where $W_{0,i} \in \mathbb{R}^{n_1}$, $n_1 = \frac{n(n+1)}{2}$, $W_{1,i} \in \mathbb{R}^{n_2 \times N}$, $W_{2,i} \in \mathbb{R}^{n \times N}$, $n_2 = \frac{n(n-1)}{2}$, $U_{0,i} \in \mathbb{R}^{m \times m}$, and $U_{1,i} \in \mathbb{R}^{m \times m \times N}$ are weighting matrices of the basis functions. $e_{\Phi,i}(\theta) \in \mathbb{C}^{n \times [\tau, 0], \mathbb{R}^{n_2}}$, $e_{\Psi,i}(\xi, \theta) \in \mathbb{C}^{n \times [\tau, 0]^2, \mathbb{R}^{n_2}}$, and $e_{\Lambda,i}(\xi, \theta) \in \mathbb{C}^{n \times [\tau, 0], \mathbb{R}^{n_2}}$ are approximation truncation errors. Therefore, by the uniform approximation theory, as $N \to \infty$, the truncation errors converge uniformly to zero, i.e., for any $\eta > 0$, there exists $N^* \in \mathbb{N}_+$, such that if $N > N^*$

$$
\begin{align*}
\|e_{\Phi,i}(\theta)\|_\infty &\leq \eta, \quad \|e_{\Psi,i}(\xi, \theta)\|_\infty \leq \eta, \\
\|e_{\Lambda,i}(\xi, \theta)\|_\infty &\leq \eta.
\end{align*}
$$

Therefore, the key idea of the data-driven PI is that $P_{1,i}(j = 0, \ldots, 3)$ and $U_{0,i}(j = 0, 1)$ are directly approximated by the data collected from system (1). Define $\bar{\mathcal{Y}}_i$ as the composite vector of the weighting matrices, i.e.,

$$
\bar{\mathcal{Y}}_i = \left[ W_{0,i}^\top, \text{vec}^\top(W_{1,i}), \text{vec}^\top(W_{2,i}), \vec{v}^\top(W_{3,i}) \right].
$$

Let $\tilde{\mathcal{Y}}_i$ be the approximation of $\mathcal{Y}_i$, and then the approximations of $P_{1,i}(j = 0, 1, 2)$ can be reconstructed by

$$
\begin{align*}
\hat{P}_{0,i} &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{0,i+1}), \\
\hat{W}_{1,i} &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{1,i+1}), \\
\hat{W}_{2,i} &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{2,i+1}), \\
\hat{K}_{1,i+1}(\theta) &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{3,i+1}(\theta)).
\end{align*}
$$

Further, $\hat{K}_{0,i+1}$ and $\hat{K}_{1,i+1}(\theta)$, the approximations of $K_{0,i+1}$ and $K_{1,i+1}(\theta)$, respectively, can be reconstructed by

$$
\begin{align*}
\hat{K}_{0,i+1}(\theta) &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{0,i+1}, \hat{\Phi}(\theta)) \\
\hat{K}_{1,i+1}(\theta) &= \text{vec}^{-1}(\tilde{\mathcal{Y}}_{1,i+1}, \hat{\Phi}(\theta)).
\end{align*}
$$

Based on the approximations in (22), we will transfer (21) to a linear equation with respect to $\bar{\mathcal{Y}}_i$. Then, the unknown vector $\bar{\mathcal{Y}}_i$ will be approximated by linear regression, and consequently, $P_{1,i}(j = 0, 1, 2)$ and $K_{1,i+1}(j = 1, 2)$ can be approximated by (25) and (26). In detail, let $\tilde{u}_i = u_i - \hat{u}_i$, define the data-constructed matrices

$$
\begin{align*}
\Gamma_{\Phi,i}(t) &= \int_{-\tau}^0 \Phi(\theta) \otimes x_i^\top(\theta) \otimes x^\top(t) d\theta, \\
\Gamma_{\Psi,i}(t) &= \int_{-\tau}^0 \Psi(\xi, \theta) \otimes \text{vec}^\top(x_i(\xi), x_i(\theta)) d\xi d\theta, \\
\Gamma_{\Lambda,i}(t) &= \int_{-\tau}^0 \Lambda(\xi, \theta) \otimes \text{vec}^\top(x_i(\xi), x_i(\theta)) d\xi d\theta.
\end{align*}
$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
With the help of (22) and (27), each term in (21) is expressed linearly with respect to the weighting matrices
\[
x^T(t)P_{0:i}(x(t)) = \text{vec}^\top(x(t))W_{0:i}
\]
\[
x^T(t)\int_{-\tau}^{0} P_{1:i}(\theta)x(t) d\theta = \Gamma_\theta x(t)\text{vec}(W_{1:i}) + \epsilon_1,1(t)
\]
\[
\int_{-\tau}^{0} \int_{-\tau}^{0} \hat{x}_i(t)\hat{x}_j(t) d\theta d\tau = \Gamma_\theta x(t)\text{vec}(W_{2:i}) + \Gamma_\theta x(t)\epsilon_1,1(t)
\]
\[
\hat{x}_i(t)K_{0:i+1}R_\theta(t) d\tau = G_{\theta}x_{i,k}U_{0:i+1} + e_{i,k}(0)
\]
\[
\hat{x}_i(t)K_{0:i+1}R_\theta(t) d\tau = G_{\theta}x_{i,k}\text{vec}(U_{1:i+1}) + \psi_{i,k} + \tilde{P}_{i,k}
\]
where \(\epsilon_1,1(t), \epsilon_2,1(t), \epsilon_3,1(t)\), and \(\psi_{i,k}\) are induced by the approximation truncation errors in (23), and \(\tilde{P}_{i,k}\) and \(\psi_{i,k}\) are induced by \(\tilde{u}_i\) (their expressions are in [41, eq. (38)]). With the collected input-state trajectories, define
\[
M_{i,k} = \left[\text{vec}^\top(x(t))\right]_{i+1}^{T} - 2G_{\theta}x_{i,k} - 2G_{\theta}x_{i,k}
\]
\[
Y_{i,k} = \int_{-\tau}^{0} x^T Q x + \tilde{u}_i^T R \tilde{u}_i dt
\]
\[
E_{i,k} = \left[2\epsilon_1,1(t) + \epsilon_2,1(t) + \epsilon_3,1(t)\right]_{i+1}^{T} - 2\psi_{i,k} - 2\tilde{P}_{i,k}
\]
\[
M_i = \left[M_{i,1}, \ldots, M_{i,k}, \ldots, M_{i,L}\right]^\top
\]
\[
Y_i = \left[Y_{i,1}, \ldots, Y_{i,k}, \ldots, Y_{i,L}\right]^\top
\]
\[
E_i = \left[E_{i,1}, \ldots, E_{i,k}, \ldots, E_{i,L}\right]^\top
\]
where \(\tilde{P}_{i,k} = \int_{i}^{T} \tilde{u}_i^T R \tilde{u}_i + u_i^T dt\).
By (28) and the definitions of \(M_i, Y_i, k\) and \(E_{i,k}\) in (29), (21) is finally transferred to a linear equation
\[
M_{i,k} \hat{Y}_{i}^{N} + E_{i,k} = Y_{i,k}
\]
Combining (30) from \(k = 1 to L\) yields
\[
M_i \hat{Y}_i^{N} + E_i = Y_i
\]
Let \(\hat{E}_i\) be the linear regression error defined as
\[
\hat{E}_i = Y_i - M_i \hat{Y}_i^{N}
\]

**Assumption 2:** Given \(N \in \mathbb{N}_+\), there exist \(L' \in \mathbb{N}_+\) and \(\alpha > 0\), such that for all \(L > L'\) and \(i \in \mathbb{N}_+\)
\[
\frac{1}{L} M_i \geq \alpha I
\]
Algorithm 1: Data-Driven Policy Iteration.

1: Choose the vectors of the basis functions \( \Phi, \Psi, \) and \( \Lambda. \)
2: Choose \( t_1, t_{L+1}, \) and \( t_k \in [t_1, t_{L+1}] . \)
3: Choose input \( u = u_1 + e, \) with \( e \) an exploration signal, to explore system (1) and collect the data of \( u(t), x(t), t \in [t_1, t_{L+1}] . \) Set the threshold \( \delta > 0 \) and \( i = 1 . \)
4: repeat
5: \( \hat{u}_i(t) = \hat{u}_i(x_i) \) along the trajectory of \( x . \)
6: Construct \( M_i \) and \( Y_i \) by (29).
7: while Assumption 2 is not satisfied
8: Collect more data and insert it into \( M_i \) and \( Y_i . \)
9: end while
10: Get \( \hat{T}_i \) by solving (34).
11: Get \( \hat{K}_j \) and \( \hat{K}_j^{i+1} \) by (26).
12: \( \hat{u}_{i+1}(x_i) = - \hat{K}_{i+1} x_i - \int_{\theta_1}^{\theta_2} \hat{K}_j x_i (\theta) d\theta \)
13: \( i \leftarrow i + 1 \)
14: until \( |\hat{T}_i - \hat{T}_{i-1}| < \delta . \)
15: Use \( \hat{u}_i(x_i) \) as the control input.

Furthermore, since \( \epsilon_1(t), \epsilon_2(t), \epsilon_3(t), \) and \( \psi_{i,k} \) are induced by the approximation truncation errors in (22), they converge to zero as \( N \to \infty. \) Consequently, by the expression of \( E_{i,k} \) in (29), for any \( l \leq L, \) \( E_{i,k} \) converges to zero as \( N \to \infty. \) Therefore, by (39), (40) holds for \( i . \) Following the logic of the content below (40), we obtain that (35) holds for \( i \). The proof is completed by induction.

Theorem 2: Given an admissible controller \( u_1, \) for any \( \eta > 0, \) there exist integers \( i^* > 0 \) and \( N^* > 0, \) such that if \( N > N^* \)
\[
|P_{i,i^*} - P_{i,i^*}^0| \leq \eta, \quad \|P_{1,i} - P_{1,i}^0\|_\infty \leq \eta, \quad \|P_{2,i} - P_{2,i}^0\|_\infty \leq \eta
\]
\[
|K_{0,i^*} - K_{0,i^*}^0| \leq \eta, \quad \|K_{1,i} - K_{1,i}^0\|_\infty \leq \eta.
\]

Proof: The theorem is proven by Theorem 1, Lemma 3, and triangle inequality. See [41, Th. 2] for details.

By Theorem 2, we see that \( \hat{K}_{j+1} \) obtained by Algorithm 1 converges to \( K_j \) as the iteration step of the algorithm and the number of basis functions tend to infinity. Hence, the proposed data-driven PI solves Problem 1.

V. PRACTICAL APPLICATIONS

The proposed data-driven PI algorithm is demonstrated by two practical examples, with regards to regenerative chatter in metal cutting (RCMC) and connected and autonomous vehicles (CAVs) in mixed traffic consisting of autonomous vehicles (AVs) and human-driven vehicles (HDVs).

A. Regenerative Chatter in Metal Cutting

Consider the example of metal cutting [37, Example 1.1], [47]. The thrust force is proportional to the instantaneous chip thickness \( (x(t+1) - x(t - \tau(t))) \), leading to the time-delay effect. The model is described by (1) with \( A \in \mathbb{R}^{2 \times 2}, \) \( A_2 \in \mathbb{R}^{2 \times 2}, \) and \( B \in \mathbb{R}^{2 \times 1} \) expressed in [41, Sec. V-A], and \( \tau = 1.3 s. \) The initial admissible controller is \( \hat{u}_1(x_1) = -K_0 x_1, \) with \( K_0 = [1.74, 3.92] . \) The exploration noise is \( e(t) = 20 \sum_{i=1}^{50} \sin \omega_i t, \) \( \omega_i \) is randomly sampled from an independent uniform distribution over \([-10, 10] . \) \( \Phi = \text{diag}([0.1, 0.1]) \) and \( \Omega = \text{diag}([1.0, \theta^2, \theta^3]) . \) For the basis functions, \( \Phi(\theta) = [1, \theta, \theta^2, \theta^3]^T, \) \( \Psi(\theta) = [1, \xi + \theta, \xi^2 + \theta^2, \xi \theta, \xi \theta^2, \xi \theta^3, \xi \theta^4 + \theta^3, \xi \theta^5 + \theta^4, \xi \theta^6 + \theta^5, \xi \theta^7 + \theta^6]^T, \) and \( \Lambda(\xi, \theta) = [1, \theta, \theta^2, \theta^3]^T \otimes [1, \xi, \xi^2, \xi^3]^T . \)

As shown in Fig. 1, the weights of the basis functions \( \hat{T} \) converge after the eighth iteration. In order to inspect the evolution of the performance index, we compare the controllers updated at each iteration for the same initial state \( x_0. \) In Fig. 1, it is seen that the performance index decreases. The responses of the state with the initial controller and the learned ADP controller are compared in Fig. 2. The performance indices are \( J(x_0, u_1) = 5.89 \times 10^4 \) and \( J(x_0, u_0) = 3.03 \times 10^4. \)

Semidiscretization [48] is applied to discretize (1) into a delay-free system with sampling period \( \Delta t = 0.1 s . \) Then, Algorithm 1 is compared with the model-based discrete-time linear quadrilateral regulator (DLQR) and the discrete-time ADP algorithm in [9] (with the same length of trajectory data). For the same initial state, the performance indices are shown in Table I. The performance index is minimal under Algorithm 1, showing that discretization sacrifices the system performance. Ideally, the performance of the discrete-time ADP is similar as the model-based DLQR. The large deviation between them is induced by the fact that the PE condition for the discrete-time ADP is not satisfied. This further illustrates that by semidiscretization, the dramatically increased dimension of the augmented state (26-dimensional) makes the requirements on the sampled data more demanding.

The robustness of Algorithm 1 to measurement noise is evaluated. The measurement of \( x(t) \) is disturbed by an independent Gaussian noise \( \varphi(t) \sim N(0, 0.2) . \) The result is shown in Fig. 3. Using the noisy data, for the same initial state \( x_0, \) the performance index converges to \( J = 3.35 \times 10^4. \) Comparing Fig. 3 with the second figure in Fig. 1, we see that Algorithm 1 can still find a near-optimal solution in the presence of noise.

B. CAVs in Mixed Traffic

Consider the platoon in Fig. 4, where the human reaction time results in the time delay. The system can be described as system (1) with \( A, A_d \in \mathbb{R}^{4 \times 4}, \) and \( B \in \mathbb{R}^{3 \times 1} \) depicted in [41, Sec. V-B],

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Evolution of the performance index using noisy data.

Fig. 3. Evolution of the performance index using noisy data.

A platoon of two HDVs and an AV.

Fig. 4. A platoon of two HDVs and an AV.

Convergence of $K_0,1$ and $K_1(\theta)$ to $K_0^*$ and $K_1^*(\theta)$.

Fig. 5. Convergence of $K_0,1$ and $K_1,1(\theta)$ to $K_0^*$ and $K_1^*(\theta)$.

Compare the initial and ADP controllers for CAVs.

Fig. 6. Compare the initial and ADP controllers for CAVs.

VI. CONCLUSION

This article has proposed for the first time a novel data-driven PI algorithm for a class of linear time-delay systems described by delay differential equations. The first major contribution of this article is to generalize the well-known Kleinman algorithm [35]—a model-based PI algorithm—from linear time-invariant systems to linear time-delay systems. The second major contribution of this article is that we have combined the proposed model-based PI algorithm and RL techniques to develop a data-driven PI algorithm for solving the direct adaptive optimal control problem for linear time-delay systems with unknown dynamics. The efficacy of the proposed learning-based adaptive optimal control design methods has been validated by two real-world applications arising from metal cutting and connected vehicles. Our future work will be directed at extending the proposed learning-based control methodology to other practically important classes of time-delay systems, such as nonlinear systems and multiantigen systems.

REFERENCES

[1] V. Kolmanovskii and A. Myshkis, Introduction to the Theory and Applications of Functional Differential Equations. New York, NY, USA: Kluwer, 1999.
[2] J. R. Hale and S. M. V. Lunel, Introduction to Functional Differential Equations. New York, NY, USA: Springer-Verlag, 1993.
[3] I. Karafyllis and Z. P. Jiang, Stability and Stabilization of Nonlinear Systems. London, U.K.: Springer, 2011.
[4] P. Pepe and Z. P. Jiang, “A Lyapunov–Krasovskii methodology for ISS and iISS of time-delay systems,” Syst Control Lett, vol. 55, no. 12, pp. 1006–1014, 2006.
[5] L. Xie, E. Fridman, and U. Shaked, “Robust $H_\infty$ control of distributed delay systems with application to combustion control,” IEEE Trans. Autom. Control, vol. 46, no. 12, pp. 1930–1935, Dec. 2001.
[6] Y. He, Q.-G. Wang, L. Xie, and C. Lin, “Further improvement of free-weighting matrices technique for systems with time-varying delay,” IEEE Trans. Autom. Control, vol. 52, no. 2, pp. 293–299, Feb. 2007.
[7] H. Gao and T. Chen, “New results on stability of discrete-time systems with time-varying state delay,” IEEE Trans. Autom. Control, vol. 52, no. 2, pp. 328–334, Feb. 2007.
[8] J. L. Ge and G. Orosz, “Optimal control of connected vehicle systems with communication delay and driver reaction time,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 8, pp. 2056–2070, Aug. 2017.
[9] M. Huang, Z. P. Jiang, and K. Ozbay, “Learning-based adaptive optimal control for connected vehicles in mixed traffic: Robustness to driver reaction time,” IEEE Trans. Cybern., vol. 52, no. 6, pp. 5267–5277, Jun. 2022.
[10] G. Stépán, “Delay effects in the human sensory system during balancing,” Phil. Trans. R. Soc. A: Math., Phys. Eng. Sci., vol. 367, pp. 1195–212, 2009.
[11] S. Liu, L. Xie, and H. Zhang, “Distributed consensus for multi-agent systems with delays and noises in transmission channels,” Automatica, vol. 47, no. 5, pp. 920–934, 2011.
[12] Y. Tang, H. Gao, W. Zhong, and J. Jiehun, “Leader-following consensus of a class of stochastic delayed multi-agent systems with partial mixed impulses,” Automatica, vol. 53, no. 1, pp. 346–354, 2015.
[13] D. Eller, J. Aggarwal, and H. Banks, “Optimal control of linear time-delay systems,” IEEE Trans. Autom. Control, vol. AC-14, no. 6, pp. 678–687, Dec. 1969.
[14] D. Ross and I. Flügge-Lotz, “An optimal control problem for systems with differential-difference equation dynamics,” SIAM J. Control Optim., vol. 7, no. 4, pp. 609–623, 1969.
[15] D. Ross, “Controller design for time delay systems via a quadratic criterion,” IEEE Trans. Autom. Control, vol. AC-16, no. 6, pp. 664–672, Dec. 1971.
[16] J. S. Gibson, “Linear-quadratic optimal control of hereditary differential systems: Infinite dimensional Riccati equations and numerical approximations,” SIAM J. Control Optim., vol. 21, no. 1, pp. 95–139, 1983.
[17] H. T. Banks, I. G. Rosen, and K. Ito, “A spline based technique for computing Riccati operators and feedback controls in regulator problems for delay equations,” SIAM J. Sci. Comput., vol. 5, no. 4, pp. 830–855, 1984.
[18] J. A. Burns, E. W. Sachs, and L. Zietsman, “Mesh independence of Kleinman–Newton iterations for Riccati equations in Hilbert space,” SIAM J. Control Optim., vol. 47, no. 5, pp. 2663–2692, 2008.

[19] Y. Jiang and Z. P. Jiang, Robust Adaptive Dynamic Programming, Hoboken, NJ, USA: Wiley, 2017.

[20] F. L. Lewis and D. Liu, Reinforcement Learning and Approximate Dynamic Programming for Feedback Control. Hoboken, New Jersey, USA: Wiley, 2013.

[21] Z. P. Jiang, T. Bian, and W. Gao, “Learning-based control: A tutorial and some recent results,” Found. Trends Syst. Control, vol. 8, no. 3, pp. 176–284, 2020.

[22] W. Gao and Z. Jiang, “Adaptive dynamic programming and adaptive optimal output regulation of linear systems,” IEEE Trans. Autom. Control, vol. 61, no. 12, pp. 4164–4169, Dec. 2016.

[23] B. Pang and Z. P. Jiang, “Adaptive optimal control of linear periodic systems: An off-policy value iteration approach,” IEEE Trans. Autom. Control, vol. 66, no. 2, pp. 888–894, Feb. 2021.

[24] L. Cui and Z. P. Jiang, “A reinforcement learning look at risk-sensitive linear quadratic Gaussian control,” 2022, arXiv:2212.02072.

[25] T. Liu, L. Cui, B. Pang, and Z. P. Jiang, “Data-driven adaptive optimal control of mixed-traffic connected vehicles in a ring road,” in Proc. IEEE 69th Conf. Decis. Control, 2021, pp. 77–82.

[26] M. Huang, Z. P. Jiang, M. Malisoff, and L. Cui, “Robust Autonomous Driving With Human in the Loop,” in Handbook of Reinforcement Learning and Control, K. G. Y. Yamvoudakis, F. L. Wan Lewis, and D. Cansever, Eds. New York, NY, USA: Springer, 2021, pp. 62–77.

[27] S. A. Asad Rizvi, Y. Wei, and Z. Lin, “Model-free optimal stabilization of unknown time delay systems using adaptive dynamic programming,” in Proc. IEEE Conf. Decis. Control., 2019, pp. 6536–6541.

[28] Q.-L. Wei, H.-G. Zhang, D.-R. Liu, and Y. Zhao, “An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming,” Acta Automatica Sinica, vol. 36, no. 1, pp. 121–129, 2010.

[29] Y. Liu, H. Zhang, Y. Luo, and J. Han, “ADP based optimal tracking control for a class of linear discrete-time system with multiple delays,” J. Franklin Inst., vol. 355, no. 9, pp. 2117–2136, 2016.

[30] H. Zhang, B. Song, Q. Wei, and T. Zhang, “Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming,” IEEE Trans. Neural Netw., vol. 22, no. 12, pp. 1851–1862, Dec. 2011.

[31] B. Wang, D. Zhao, C. Alippi, and D. Liu, “Dual heuristic dynamic programming for nonlinear discrete-time uncertain systems with state delay,” Neurocomputing, vol. 134, pp. 222–229, 2014.

[32] J. G. Rueda-Escobedo, E. Fridman, and J. Schiffer, “Data-driven control for linear discrete-time delay systems,” IEEE Trans. Autom. Control, vol. 67, no. 7, pp. 3321–3336, Jul. 2022.

[33] R. Moghadam, S. Jaganathan, V. Narayanan, and K. Raghavan, Optimal Adaptive Control of Partially Uncertain Linear Continuous-Time Systems With State Delay. Cham, Switzerland: Springer, 2021.

[34] R. Moghadam and S. Jaganathan, “Optimal adaptive control of uncertain nonlinear continuous-time systems with input and state delays,” IEEE Trans. Neural Netw. Learn. Syst., to be published, doi: 10.1109/TNNLS.2021.3112566.

[35] D. Kleinman, “On an iterative technique for Riccati equation computations,” IEEE Trans. Autom. Control, vol. AC-13, no. 1, pp. 114–115, Feb. 1968.

[36] R. F. Curtain, An Introduction to Infinite-Dimensional Linear Systems Theory. New York, NY, USA: Springer, 1995.

[37] K. Gu, V. L. Kharitonov, and J. Chen, Stability of Time-Delay Systems. Boston, MA, USA: Birkhäuser, 2003.

[38] E. Fridman, Introduction to Time-Delay Systems Analysis and Control. Cham, Switzerland: Springer, 2014.

[39] R. E. Kalman, “Contributions to the theory of optimal control,” Boletin Sociedad Matematica Mexicana, vol. 5, no. 2, pp. 102–119, 1960.

[40] K. Uchida and E. Shimemura, “Closed-loop properties of the infinite-time linear-quadratic optimal regulator for systems with delays,” Int. J. Control, vol. 43, no. 3, pp. 773–779, 1986.

[41] L. Cui, B. Pang, and Z. P. Jiang, “Learning-based adaptive optimal control of linear time-delay systems: A policy iteration approach,” 2022, arXiv:2210.00204.

[42] Y. Eidelman, V. Milman, and A. Tsolomitis, Functional Analysis, an Introduction. Providence, RI, USA: Amer. Math. Soc., 2004.

[43] C. C. Pugh, Real Mathematical Analysis, 2nd ed. Cham, Switzerland: Springer International Publishing, 2015.

[44] M. J. D. Powell, Approximation Theory and Methods. New York, NY, USA: Cambridge Univ. Press, 1981.

[45] Z. P. Jiang, C. Prieur, and A. Astolfi, Eds., Trends in Nonlinear and Adaptive Control: A Tribute to Laurent Praly for His 65th Birthday, Berlin, Germany: Springer, 2021.

[46] K. J. Aström and B. Wittenmark, Adaptive Control, 2nd ed. Boston, MA, USA: Addison-Wesley, 1997.

[47] C. Mei, J. G. Cherng, and Y. Wang, “Active control of regenerative chatter during metal cutting process,” J. Manuf. Sci. Eng., vol. 128, pp. 346–349, 2005.

[48] T. Insperger and G. Stépán, “Semi-discretization method for delayed certain nonlinear continuous-time systems with input and state delays,” IEEE Trans. Autom. Control, vol. 61, no. 12, pp. 4164–4169, Dec. 2016.

[49] M. Gu, V. L. Kharitonov, and J. Chen, Stability of Time-Delay Systems. Boston, MA, USA: Birkhäuser, 2003.

[50] K. Uchida and E. Shimemura, “Closed-loop properties of the infinite-time linear-quadratic optimal regulator for systems with delays,” Int. J. Control, vol. 43, no. 3, pp. 773–779, 1986.

[51] L. Cui, B. Pang, and Z. P. Jiang, “Learning-based adaptive optimal control of linear time-delay systems: A policy iteration approach,” 2022, arXiv:2210.00204.

[52] Y. Eidelman, V. Milman, and A. Tsolomitis, Functional Analysis, an Introduction. Providence, RI, USA: Amer. Math. Soc., 2004.

[53] C. C. Pugh, Real Mathematical Analysis, 2nd ed. Cham, Switzerland: Springer International Publishing, 2015.

[54] M. J. D. Powell, Approximation Theory and Methods. New York, NY, USA: Cambridge Univ. Press, 1981.

[55] Z. P. Jiang, C. Prieur, and A. Astolfi, Eds., Trends in Nonlinear and Adaptive Control: A Tribute to Laurent Praly for His 65th Birthday, Berlin, Germany: Springer, 2021.

[56] K. J. Aström and B. Wittenmark, Adaptive Control, 2nd ed. Boston, MA, USA: Addison-Wesley, 1997.