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 paper studies the adaptive optimal control problem for a class of linear time-delay systems described by delay differential equations (DDEs). A crucial strategy is to take advantage of recent developments in reinforcement learning (RL) and adaptive dynamic programming (ADP) and develop novel methods to learn adaptive optimal controllers from finite samples of input and state data. In this paper, the data-driven policy iteration (PI) is proposed to solve the infinite-dimensional algebraic Riccati equation (ARE) 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, optimal control, linear time-delay systems, policy iteration.**

I. Introduction

Time-delay systems are ubiquitous in many branches of science and engineering; see the books [1]–[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_∞ 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 multi-agent 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]–[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 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 and quadrupedal robots, the state and action spaces are infinite and the stability of the controller generated by an RL algorithm is innegligible. 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 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 sub-optimal controller with improved performance is obtained. Besides, the sequence of these sub-optimal 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]–[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]–[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], [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. Firstly, for an infinite-dimensional system, optimality properties are hard to analyze, because the corresponding ARE are complex partial differential equations (PDEs). Secondly, stability analysis and controller design for a time-delay system are much more challenging than finite-dimensional systems.

*This work is supported partly by the National Science Foundation under Grants EPCN-1903781 and CNS-2148309.

L. Cui, B. Pang, and Z. P. Jiang 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-mail: l.cui@nyu.edu; bo.pang@nyu.edu; zjiang@nyu.edu).

Therefore, the model-free optimal control for a continuous-time time-delay system remains an open problem.

In this paper, in the absence of the precise knowledge of system dynamics, a novel data-driven PI approach for continuous-time linear time-delay systems is proposed based on ADP. The contributions of this paper are as follows. Firstly, 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 stability of the updated sub-optimal 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. Secondly, based on the model-based PI, this paper 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 paper 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 paper. 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, some concluding remarks are drawn in Section VI.

Notations: In this paper, $\mathbb{R}$ ($\mathbb{R}_+$) denotes the set of (nonnegative) real numbers and $\mathbb{N}_+$ denotes the set of positive integers. $|\cdot|$ denotes the Euclidean norm of a vector or Frobenius norm of a matrix. $\|\cdot\|_\infty$ denotes the supremum norm of a function. $\mathcal{C}^0(X,Y)$ denotes the class of continuous functions from the linear space X to the linear space Y. $\mathcal{AC}([-\tau,0],\mathbb{R}^n)$ denotes the class of absolutely continuous functions. $\frac{\mathrm{d}f}{\mathrm{d}\theta}(\cdot)$ denotes the function which is the derivative of the function f. $\oplus$ denotes the direct sum. $L_i([-\tau,0],\mathbb{R}^n)$ denotes the space of measurable functions for which the ith power of the Euclidean norm is Lebesgue integrable, $\mathcal{M}_2 = \mathbb{R}^n \oplus L_2([-\tau,0],\mathbb{R}^n)$, and $\mathcal{D} = \left\{ \begin{bmatrix} r \\ f(\cdot) \end{bmatrix} \in \mathcal{M}_2 : f \in \mathcal{AC}, \frac{\mathrm{d}f}{\mathrm{d}\theta}(\cdot) \in L_2, \text{ and } f(0) = r \right\}$. $\langle \cdot,\cdot \rangle$ denotes the inner product in $\mathcal{M}_2$, i.e. $\langle z_1, z_2 \rangle = r_1^\top r_2 + \int_{-\tau}^0 f_1^\top(\theta) f_2(\theta)\mathrm{d}\theta$, where $z_i = [r_i, f_i(\cdot)]^\top$ for $i = 1, 2$. $\mathcal{L}(X)$ and $\mathcal{L}(X,Y)$ denote the class of continuous bounded linear operators from X to X and from X to Y, respectively. $\otimes$ denotes the Kronecker product. $\mathrm{vec}(A) = \left[a_1^\top, a_2^\top, \cdots, a_n^\top\right]^\top$, where $A \in \mathbb{R}^{n\times n}$ and a_i is the ith column of A. For a symmetric matrix $P \in \mathbb{R}^{n\times n}$ with the entry p_{ij}, $\mathrm{vecs}(P) = [p_{11}, 2p_{12}, \cdots, 2p_{1n}, p_{22}, 2p_{23}, \cdots, 2p_{(n-1)n}, p_{nn}]^\top$, $\mathrm{vecu}(P) = 2[p_{12}, \cdots, p_{1n}, p_{23}, \cdots, p_{(n-1)n}]^\top$, and $\mathrm{diag}(P) = [p_{11}, p_{22}, \cdots, p_{nn}]^\top$. For two arbitrary vectors $\nu, \mu \in \mathbb{R}^n$, $\mathrm{vecd}(\nu,\mu) = [\nu_1\mu_1, \cdots, \nu_n\mu_n]^\top$, $\mathrm{vecv}(\nu) = [\nu_1^2, \nu_1\nu_2, \cdots, \nu_1\nu_n, \nu_2^2, \cdots, \nu_{n-1}\nu_n, \nu_n^2]^\top$, $\mathrm{vecp}(\nu,\mu) = [\nu_1\mu_2, \cdots, \nu_1\mu_n, \nu_2\mu_3, ..., \nu_{n-1}\mu_n]^\top$. $[a]_{i,j}$ denotes the sub-vector of the vector a comprised of the entries between the ith and jth entries. $A^\dagger$ denotes the Moore-Penrose inverse of matrix A.

II. Problem Formulation and Preliminaries

A. Problem Formulation

Consider a linear time-delay system

$$\dot{x}(t) = Ax(t) + A_d x(t-\tau) + Bu(t), \tag{1}$$

where $\tau \in \mathbb{R}_+$ denotes the constant delay of the system and is assumed to be known, $x(t) \in \mathbb{R}^n$, and $u(t) \in \mathbb{R}^m$. A, $A_d \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown constant matrices. Let $x_t(\theta) = x(t+\theta), \forall\theta \in [-\tau, 0]$, denote a segment of the state trajectory in the interval $[t-\tau, t]$. Due to the infinite dimensionality of system (1), the state of the system is $z(t) = [x^\top(t), x_t^\top(\cdot)]^\top \in \mathcal{M}_2$. Define the linear operators $\mathbf{A} \in \mathcal{L}(\mathcal{M}_2)$ and $\mathbf{B} \in \mathcal{L}(\mathbb{R}^m, \mathcal{M}_2)$ as $\mathbf{A}z(t) = \begin{bmatrix} Ax(t) + A_d x_t(-\tau) \\ \frac{\mathrm{d}x_t}{\mathrm{d}\theta}(\cdot) \end{bmatrix}$ and $\mathbf{B}u(t) = \begin{bmatrix} Bu(t) \\ 0 \end{bmatrix}$. Then, according to [36, Theorem 2.4.6], (1) is rewritten as

$$\dot{z}(t) = \mathbf{A}z(t) + \mathbf{B}u(t), \tag{2}$$

with the domain of $\mathbf{A}$ given by $\mathcal{D}$. Let $z_0 = [x^\top(0), x_0^\top(\cdot)]^\top \in \mathcal{D}$ denote the initial state of the system (2). The quadratic performance index is defined as

$$J(x_0, u) = \int_0^\infty x(t)^\top Q x(t) + u(t)^\top R u(t)\mathrm{d}t, \tag{3}$$

where $Q = Q^\top \geq 0$ and $R = R^\top > 0$.

Definition 1 : *A control policy $u_c(x_t)$ is admissible for system (1) with respect to (3), if system (1) with $u_c(x_t)$ is globally asymptotically stable (GAS) at the origin [37, Definition 1.1], and the performance index (3) is finite for all $z_0 \in \mathcal{D}$.*

Assumption 1. *System (1) with output $y(t) = Q^{\frac{1}{2}}x(t)$ is exponentially stabilizable and detectable, which are defined in [36, Definition 5.2.1] and checked by [36, Theorem 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 paper can be formulated as follows.

Problem 1. *Given an initial admissible controller $u_1(x_t) = -K_{0,1}x(t) - \int_{-\tau}^0 K_{1,1}(\theta)x_t(\theta)\mathrm{d}\theta$, and without knowing the dynamics of system (1), 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. $A_d = 0$ 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 1 ([14], [40]). *For system (1) with Assumption 1,*

$$u^*(x_t) = -\underbrace{R^{-1}B^\top P_0^*}_{K_0^*} x(t) - \int_{-\tau}^0 \underbrace{R^{-1}B^\top P_1^*(\theta)}_{K_1^*(\theta)} x_t(\theta)\mathrm{d}\theta \tag{4}$$

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

$$V^*(x_0) = x^\top(0)P_0^*x(0) + 2x^\top(0)\int_{-\tau}^{0} P_1^*(\theta)x_0(\theta)\mathrm{d}\theta + \int_{-\tau}^{0}\int_{-\tau}^{0} x_0^\top(\xi)P_2^*(\xi,\theta)x_0(\theta)\mathrm{d}\xi\mathrm{d}\theta, \tag{5}$$

where $P_0^* = P_0^{*\top}$, $P_1^*(\theta)$, and $P_2^{*\top}(\theta,\xi) = P_2^*(\xi,\theta)$ for $\theta,\xi \in [-\tau,0]$ are the unique stabilizing solution to the following PDEs

$$\begin{aligned}
&A^\top P_0^* + P_0^*A - P_0^*BR^{-1}B^\top P_0^* + P_1^*(0) + P_1^{*\top}(0) + Q = 0,\\
&\frac{\mathrm{d}P_1^*(\theta)}{\mathrm{d}\theta} = (A^\top - P_0^*BR^{-1}B^\top)P_1^*(\theta) + P_2^*(0,\theta),\\
&\partial_\xi P_2^*(\xi,\theta) + \partial_\theta P_2^*(\xi,\theta) = -P_1^{*\top}(\xi)BR^{-1}B^\top P_1^*(\theta),\\
&P_1^*(-\tau) = P_0^*A_d, \qquad P_2^*(-\tau,\theta) = A_d^\top P_1^*(\theta).
\end{aligned} \tag{6}$$

By [36, Theorem 6.2.7], the time-delay system (1) in closed-loop with u^* is exponentially stable at the origin.

III. Model-Based Policy Iteration

According to Lemma 1, the optimal controller is obtained by solving (6) . Since (6) is nonlinear with respect to P_0^*, P_1^* and P_2^*, 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 $u_1(x_t) = -K_{0,1}x(t) - \int_{-\tau}^{0} K_{1,1}(\theta)x_t(\theta)\mathrm{d}\theta$, 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 $P_{0,i} = P_{0,i}^\top$, $P_{1,i}(\theta)$, and $P_{2,i}^\top(\theta,\xi) = P_{2,i}(\xi,\theta)$ by solving the following PDEs,

$$\begin{aligned}
&A_i^\top P_{0,i} + P_{0,i}A_i + Q_i + P_{1,i}(0) + P_{1,i}^\top(0) = 0,\\
&\frac{\mathrm{d}P_{1,i}(\theta)}{\mathrm{d}\theta} = A_i^\top P_{1,i}(\theta) - P_{0,i}BK_{1,i}(\theta) + K_{0,i}^\top RK_{1,i}(\theta) + P_{2,i}(0,\theta),\\
&\partial_\xi P_{2,i}(\xi,\theta) + \partial_\theta P_{2,i}(\xi,\theta) = K_{1,i}^\top(\xi)RK_{1,i}(\theta) - 2K_{1,i}^\top(\xi)B^\top P_{1,i}(\theta),\\
&P_{1,i}(-\tau) = P_{0,i}A_d, \qquad P_{2,i}(-\tau,\theta) = A_d^\top P_{1,i}(\theta),
\end{aligned} \tag{7}$$

where $A_i = (A - BK_{0,i})$ and $Q_i = Q + K_{0,i}^\top RK_{0,i}$.

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

$$u_{i+1}(x_t) = -\underbrace{R^{-1}B^\top P_{0,i}}_{K_{0,i+1}}x(t) - \int_{-\tau}^{0}\underbrace{R^{-1}B^\top P_{1,i}(\theta)}_{K_{1,i+1}(\theta)}x_t(\theta)\mathrm{d}\theta. \tag{8}$$

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

$$V_i(x_0) = x^\top(0)P_{0,i}x(0) + 2x^\top(0)\int_{-\tau}^{0} P_{1,i}(\theta)x_0(\theta)\mathrm{d}\theta + \int_{-\tau}^{0}\int_{-\tau}^{0} x_0^\top(\xi)P_{2,i}(\xi,\theta)x_0(\theta)\mathrm{d}\xi d\theta. \tag{9}$$

By policy improvement, the value functional is monotonically decreasing ($V_{i+1}(x_0) \le V_i(x_0)$), and converges to the optimal value functional $V^*(x_0)$. 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 $u_L(x_t) = -\mathbf{K}z(t)$ satisfies $J(x_0,u_L) < \infty$ for any $z_0 \in \mathcal{D}$, where $\mathbf{K} \in \mathcal{L}(\mathcal{M}_2,\mathbb{R}^m)$, 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 $u_1(x_t)$, for $P_{0,i}$, $P_{1,i}(\theta)$, $P_{2,i}(\xi,\theta)$, and $u_{i+1}(x_t)$ obtained by solving (7) and (8), and for all $i \in \mathbb{N}_+$, the following properties hold*

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

Proof. Along the trajectories of (1) driven by u, where u without subscript stands for an arbitrary input, $\dot{V}_i(x_t)$ is

$$\dot{V}_i(x_t) = -x^\top Qx - u_i^\top Ru_i + 2u_{i+1}^\top Ru_i - 2u^\top Ru_{i+1}. \tag{10}$$

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

Property 1) is proved by induction. When $i = 1$, the admissibility of $u_1(x_t)$ is given. For $i > 1$, assume that u_i is admissible. When system (1) is driven by u_i, by (10), the expression of $\dot{V}_i(x_t)$ is

$$\dot{V}_i(x_t) = -x^\top Qx - u_i^\top Ru_i. \tag{11}$$

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

$$\begin{aligned}
V_i(x_0) &= \int_0^\infty x^\top(t)Qx(t) + u_i^\top(t)Ru_i(t)\mathrm{d}t\\
&= J(x_0,u_i) < \infty.
\end{aligned} \tag{12}$$

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

$$\begin{aligned}
\dot{V}_i(x_t) = &-x^\top Qx - u_{i+1}^\top Ru_{i+1}\\
&- (u_{i+1} - u_i)^\top R(u_{i+1} - u_i).
\end{aligned} \tag{13}$$

Integrating both sides of (13) from 0 to ∞ yields

$$\begin{aligned}
J(x_0,u_{i+1}) &= V_i(x_0) - V_i(x_\infty)\\
&- \int_0^\infty (u_{i+1} - u_i)^\top R(u_{i+1} - u_i)\mathrm{d}t \le V_i(x_0) < \infty.
\end{aligned} \tag{14}$$

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),

$$\dot{V}_{i+1}(x_t) = -x^\top Qx - u_{i+1}^\top Ru_{i+1}. \tag{15}$$

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

Define $\mathbf{P}_i \in \mathcal{L}(\mathcal{M}_2)$, such that for any z_0, $\mathbf{P}_iz_0$ is

$$\mathbf{P}_iz_0 = \begin{bmatrix} P_{0,i}x(0) + \int_{-\tau}^{0} P_{1,i}(\theta)x_0(\theta)\mathrm{d}\theta \\ \int_{-\tau}^{0} P_{2,i}(\cdot,\theta)x_0(\theta)\mathrm{d}\theta + P_{1,i}^\top(\cdot)x(0) \end{bmatrix}. \tag{16}$$

It is easy to check that $\mathbf{P}_i$ is symmetric [42, Chapter 6], and non-negative [42, Definition 6.3.1], and $V_i(x_0) = \langle z_0, \mathbf{P}_i z_0 \rangle$. Furthermore, according to statement 2), for any $i \in \mathbb{N}_+$, $\mathbf{P}^* \leq \mathbf{P}_{i+1} \leq \mathbf{P}_i$. According to [42, Theorem 6.3.2], there exists $\mathbf{P}_p = \mathbf{P}_p^\top \geq 0$, such that for all $z_0 \in \mathcal{M}_2$, we have

$$\lim_{i\to\infty} \mathbf{P}_i z_0 = \mathbf{P}_p z_0. \tag{17}$$

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 $\mathbf{P}_i$ converges, $P_{0,p}$, $P_{1,p}(\theta)$ and $P_{2,p}(\xi,\theta)$ satisfy (7) with i replaced by p. $K_{0,i}$ and $K_{1,i}$ converge to $K_{0,p}$ and $K_{1,p}$. By the policy improvement step (8), $K_{0,p}$ and $K_{1,p}$ satisfy

$$K_{0,p} = R^{-1}B^\top P_{0,p}, \qquad K_{1,p}(\theta) = R^{-1}B^\top P_{1,p}(\theta). \tag{18}$$

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^*, $P_1^*(\theta)$ and $P_2^*(\xi,\theta)$. 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, Chapter 4, Theorem 16]. Hence, 3) is proved. □

Notice that although (7) is linear with respect to $P_{0,i}$, $P_{1,i}$, and $P_{2,i}$, since (7) are PDEs, obtaining the analytical solution to (7) is still non-trivial. Besides, the accurate knowledge of the system matrices A, A_d, and B is required to implement the model-based PI, and in practice due to the complex structure of the system, it is often hard to derive such an accurate model. Therefore, in the next section, a data-driven PI algorithm is proposed to approximate the optimal solution.

Remark 2. *When $A_d = 0$, (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 paper, the concrete expression of PI is constructed in (7) and (8), which is one of the major contributions in this paper. Besides, it can be checked that at each iteration, $\mathbf{P}_i$ 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 Policy Iteration

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 $[t_1, t_{L+1}]$ is applied to train the control policy. From the RL perspective, u is named behaviour/exploratory policy.

Define $v_i(t) = u(t) - u_i(x_t)$, where $u_i(x_t)$ 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 behaviour/exploratory policy u,

$$\dot{V}_i(x_t) = -x^\top Q x - u_i^\top R u_i - 2u_{i+1}^\top R v_i. \tag{19}$$

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

$$V_i(x_{t_{k+1}}) - V_i(x_{t_k}) = \int_{t_k}^{t_{k+1}} -x^\top Q x - u_i^\top R u_i - 2u_{i+1}^\top R v_i \mathrm{d}t. \tag{20}$$

Plugging the expressions of u_{i+1} in (8) and V_i in (9) into (20) yields

$$\begin{aligned}
&\Bigg[x^\top(t)P_{0,i}x(t) + 2x^\top(t)\int_{-\tau}^{0} P_{1,i}(\theta)x_t(\theta)\mathrm{d}\theta \\
&\quad + \int_{-\tau}^{0}\int_{-\tau}^{0} x_t^\top(\xi)P_{2,i}(\xi,\theta)x_t(\theta)\mathrm{d}\xi\mathrm{d}\theta\Bigg]_{t=t_k}^{t_{k+1}} \\
&\quad - 2\int_{t_k}^{t_{k+1}} \left(x^\top(t)K_{0,i+1}^\top + \int_{-\tau}^{0} x_t^\top(\theta)K_{1,i+1}^\top(\theta)\mathrm{d}\theta\right) R v_i(t)\mathrm{d}t \\
&= -\int_{t_k}^{t_{k+1}} x(t)^\top Q x(t) + u_i(t)^\top R u_i(t)\mathrm{d}t.
\end{aligned} \tag{21}$$

As seen in (7) and (8), $K_{1,i}(\theta)$ and $P_{1,i}(\theta)$ ($P_{2,i}(\xi,\theta)$) 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 Φ, Λ and Ψ. According to the approximation theory [44], the following equations hold

$$\begin{aligned}
&\mathrm{vecs}(P_{0,i}) = W_{0,i},\ \mathrm{vec}(P_{1,i}(\theta)) = W_{1,i}^N\Phi(\theta) + e_{\Phi,i}^N(\theta), \\
&\mathrm{diag}(P_{2,i}(\xi,\theta)) = W_{2,i}^N\Psi(\xi,\theta) + e_{\Psi,i}^N(\xi,\theta), \\
&\mathrm{vecu}(P_{2,i}(\xi,\theta)) = W_{3,i}^N\Lambda(\xi,\theta) + e_{\Lambda,i}^N(\xi,\theta), \\
&\mathrm{vec}(K_{0,i}) = U_{0,i},\ \mathrm{vec}(K_{1,i}(\theta)) = U_{1,i}^N\Phi(\theta) + e_{K,i}^N(\theta),
\end{aligned} \tag{22}$$

where $W_{0,i} \in \mathbb{R}^{n_1}$, $n_1 = \frac{n(n+1)}{2}$, $W_{1,i}^N \in \mathbb{R}^{n^2\times N}$, $W_{2,i}^N \in \mathbb{R}^{n\times N}$, $W_{3,i}^N \in \mathbb{R}^{n_2\times N}$, $n_2 = \frac{n(n-1)}{2}$, $U_{0,i} \in \mathbb{R}^{nm}$, and $U_{1,i}^N \in \mathbb{R}^{nm\times N}$ are weighting matrices of the basis functions. $e_{\Phi,i}^N(\theta) \in \mathcal{C}^0([-\tau,0],\mathbb{R}^{n^2})$, $e_{\Psi,i}^N(\xi,\theta) \in \mathcal{C}^0([-\tau,0]^2,\mathbb{R}^n)$, $e_{\Lambda,i}^N(\xi,\theta) \in \mathcal{C}^0([-\tau,0]^2,\mathbb{R}^{n_2})$, and $e_{K,i}^N(\theta) \in \mathcal{C}^0([-\tau,0],\mathbb{R}^{mn})$ 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 $N > N^*$,

$$\begin{aligned}
&\|e_{\Phi,i}^N(\theta)\|_\infty \leq \eta, \qquad \|e_{K,i}^N(\theta)\|_\infty \leq \eta, \\
&\|e_{\Psi,i}^N(\xi,\theta)\|_\infty \leq \eta, \qquad \|e_{\Lambda,i}^N(\xi,\theta)\|_\infty \leq \eta.
\end{aligned} \tag{23}$$

Therefore, the key idea of the data-driven PI is that $W_{j,i}(j=0,\cdots,3)$ and $U_{j,i}(j=0,1)$ are directly approximated by the data collected from system (1). Define Υ_i^N as the composite vector of the weighting matrices, i.e.

$$\begin{aligned}\Upsilon_i^N = \big[&W_{0,i}^\top, \mathrm{vec}^\top(W_{1,i}^N), \mathrm{vec}^\top(W_{2,i}^N), \mathrm{vec}^\top(W_{3,i}^N)\\ &U_{0,i+1}^\top, \mathrm{vec}^\top(U_{1,i+1}^N)\big]^\top.\end{aligned} \tag{24}$$

Let $\hat{\Upsilon}_i^N$ be the approximation of Υ_i^N, and then, the approximations of $P_{j,i}(j=0,1,2)$ can be reconstructed by

$$\begin{aligned}&\hat{P}_{0,i} = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{1,n_1}),\ \hat{W}_{1,i}^N = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{n_1+1,n_3}),\\ &\hat{W}_{2,i}^N = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{n_3+1,n_4}),\ \hat{W}_{3,i}^N = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{n_4+1,n_5}),\\ &\hat{P}_{1,i}(\theta) = \mathrm{vec}^{-1}(\hat{W}_{1,i}^N\Phi(\theta)),\ \mathrm{diag}(\hat{P}_{2,i}(\xi,\theta)) = \hat{W}_{2,i}^N\Psi(\xi,\theta),\\ &\mathrm{vecu}(\hat{P}_{2,i}(\xi,\theta)) = \hat{W}_{3,i}^N\Lambda(\xi,\theta),\end{aligned} \tag{25}$$

where $n_3 = n_1 + n^2N$, $n_4 = n_3 + nN$, $n_5 = n_4 + n_2N$. Furthermore, $\hat{K}_{0,i+1}$ 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

$$\begin{aligned}&\hat{K}_{0,i+1} = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{n_5+1,n_6}),\ \hat{U}_{1,i+1} = \mathrm{vec}^{-1}([\hat{\Upsilon}_i^N]_{n_6+1,n_7}),\\ &\hat{K}_{1,i+1}(\theta) = \mathrm{vec}^{-1}(\hat{U}_{1,i+1}\Phi(\theta)),\end{aligned} \tag{26}$$

where $n_6 = n_5+nm$, and $n_7 = n_6+nmN$. As a consequence, $\hat{u}_i(x_t)$, the approximation of $u_i(x_t)$, can be expressed as $\hat{u}_i(x_t) = -\hat{K}_{0,i}x(t) - \int_{-\tau}^0 \hat{K}_{1,i}(\theta)x_t(\theta)\mathrm{d}\theta$.

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

$$\begin{aligned}\Gamma_{\Phi xx}(t) &= \int_{-\tau}^0 \Phi^\top(\theta)\otimes x_t^\top(\theta)\otimes x^\top(t)\mathrm{d}\theta,\\ \Gamma_{\Psi xx}(t) &= \int_{-\tau}^0\int_{-\tau}^0 \Psi^\top(\xi,\theta)\otimes \mathrm{vecd}^\top(x_t(\xi),x_t(\theta))\mathrm{d}\xi\mathrm{d}\theta,\\ \Gamma_{\Lambda xx}(t) &= \int_{-\tau}^0\int_{-\tau}^0 \Lambda^\top(\xi,\theta)\otimes \mathrm{vecp}^\top(x_t(\xi),x_t(\theta))\mathrm{d}\xi\mathrm{d}\theta,\\ G_{x\hat{v}_i,k} &= \int_{t_k}^{t_{k+1}} (x^\top(t)\otimes\hat{v}_i^\top(t))(I_n\otimes R)\mathrm{d}t,\\ G_{\Phi x\hat{v}_i,k} &= \int_{t_k}^{t_{k+1}}\int_{-\tau}^0 \Phi^\top(\theta)\otimes((x_t^\top(\theta)\otimes\hat{v}_i^\top(t))(I_n\otimes R))\mathrm{d}\theta\mathrm{d}t.\end{aligned} \tag{27}$$

With the help of (22) and (27), each term in (21) is expressed linearly with respect to the weighting matrices

$$\begin{aligned}&x^\top(t)P_{0,i}x(t) = \mathrm{vecv}^\top(x(t))W_{0,i},\\ &x^\top(t)\int_{-\tau}^0 P_{1,i}(\theta)x_t(\theta)d\theta = \Gamma_{\Phi xx}(t)\mathrm{vec}(W_{1,i}) + \epsilon_{1,i}(t),\\ &\int_{-\tau}^0\int_{-\tau}^0 x_t^\top(\xi)P_{2,i}(\xi,\theta)x_t(\theta)d\xi d\theta = \Gamma_{\Psi xx}(t)\mathrm{vec}(W_{2,i})\\ &\quad + \Gamma_{\Lambda xx}(t)\mathrm{vec}(W_{3,i}) + \epsilon_{2,i}(t) + \epsilon_{3,i}(t),\\ &\int_{t_k}^{t_{k+1}} x^\top(t)K_{0,i+1}^\top Rv_i(t)dt = G_{x\hat{v}_i,k}U_{0,i+1} + \rho_{i,k}^0,\\ &\int_{t_k}^{t_{k+1}}\int_{-\tau}^0 x_t^\top(\theta)K_{1,i+1}^\top(\theta)Rv_i(t)d\theta dt = G_{\Phi x\hat{v}_i,k}\mathrm{vec}(U_{1,i+1})\\ &\quad + \psi_{i,k} + \rho_{i,k}^1,\end{aligned} \tag{28}$$

where $\epsilon_{1,i}(t)$ $\epsilon_{2,i}(t)$, $\epsilon_{3,i}(t)$, and $\psi_{i,k}$ are induced by the approximation truncation errors in (23), and $\rho_{i,k}^0$ and $\rho_{i,k}^1$ are induced by $\tilde{u}_i$ (their expressions are in [41, Equation (38)]). With the collected input-state trajectories, define

$$\begin{aligned}M_{i,k} &= \Big[\mathrm{vecv}^\top(x(t))|_{t_k}^{t_{k+1}}, 2\Gamma_{\Phi xx}|_{t_k}^{t_{k+1}}, \Gamma_{\Psi xx}(t)|_{t_k}^{t_{k+1}},\\ &\qquad \Gamma_{\Lambda xx}(t)|_{t_k}^{t_{k+1}}, -2G_{x\hat{v}_i,k}, -2G_{\Phi x\hat{v}_i,k}\Big],\\ Y_{i,k} &= -\int_{t_k}^{t_{k+1}} x^\top Qx + \hat{u}_i^\top R\hat{u}_i\mathrm{d}t,\\ E_{i,k} &= [2\epsilon_{1,i}(t) + \epsilon_{2,i}(t) + \epsilon_{3,i}(t)]_{t=t_k}^{t_{k+1}} - 2\psi_{i,k} - 2\rho_{i,k}^0\\ &\quad - 2\rho_{i,k}^1 - \rho_{i,k}^2,\\ M_i &= \big[M_{i,1}^\top,\cdots,M_{i,k}^\top,\cdots,M_{i,L}^\top\big]^\top,\\ Y_i &= [Y_{i,1},\cdots,Y_{i,k},\cdots,Y_{i,L}]^\top,\\ E_i &= [E_{i,1},\cdots,E_{i,k},\cdots,E_{i,L}]^\top,\end{aligned} \tag{29}$$

where $\rho_{i,k}^2 = \int_{t_k}^{t_{k+1}} \tilde{u}_i^\top R(\hat{u}_i + u_i)\mathrm{d}t$.

By (28) and the definitions of $M_{i,k}$, $Y_{i,k}$ and $E_{i,k}$ in (29), (21) is finally transferred to a linear equation

$$M_{i,k}\Upsilon_i^N + E_{i,k} = Y_{i,k}. \tag{30}$$

Combining (30) from $k=1$ to $k=L$ yields

$$M_i\Upsilon_i^N + E_i = Y_i. \tag{31}$$

Let $\hat{E}_i$ be the linear regression error defined as

$$\hat{E}_i = Y_i - M_i\hat{\Upsilon}_i^N. \tag{32}$$

Assumption 2. *Given $N\in\mathbb{N}_+$, there exist $L^*\in\mathbb{N}_+$ and $\alpha>0$, such that for all $L>L^*$ and $i\in\mathbb{N}_+$,*

$$\frac{1}{L}M_i^\top M_i \geq \alpha I. \tag{33}$$

Remark 5. *Assumption 2 is reminiscent of the persistent excitation (PE) condition [45], [46]. It is needed to guarantee the uniqueness of the least-square solution to (31), and prove the convergence of the proposed data-driven PI algorithm. As in the literature of ADP-based data-driven control [19], [20], one can fulfill it by means of added exploration noise, such as sinusoidal signals and random noise.*

Under Assumption 2, the method of least squares is applied to minimize $\hat{E}_i^\top\hat{E}_i$, i.e. $\hat{E}_i^\top\hat{E}_i$ is minimized by

$$\hat{\Upsilon}_i^N = M_i^\dagger Y_i. \tag{34}$$

With the result of $\hat{\Upsilon}_i^N$ in (34), $\hat{P}_{j,i}(j=0\cdots2)$ and $\hat{K}_{j,i}(j=0,1)$ can be reconstructed by (25) and (26) respectively.

The proposed algorithm is shown in Algorithm 1. From (29), M_i and Y_i are constructed by the input-state trajectory data of system (1). Hence, the system matrices are not involved in the computation of $\hat{\Upsilon}_i^N$. Furthermore, since the behaviour/exploratory policy u is different from the updated policy u_i, Algorithm 1 is called off-policy.

Remark 6. *Due to the property that $P_{2,i}^\top(\xi,\theta) = P_{2,i}(\theta,\xi)$, the diagonal elements of $P_{2,i}$ satisfy $diag(P_{2,i}(\xi,\theta)) = diag(P_{2,i}(\theta,\xi))$. Hence, the vector of basis functions Ψ should satisfy $\Psi(\xi,\theta) = \Psi(\theta,\xi)$ to approximate such functions.*

Remark 7. *In practice, the integrals in (27) are calculated by Riemann sum, like midpoint, trapezoid, and Simpson's rules.*

Lemma 3. *Under Assumption 2, and given an admissible controller* $u_1(x_t) = -K_{0,1}x(t) - \int_{-\tau}^{0} K_{1,1}(\theta)x_t(\theta)\mathrm{d}\theta$*, for each* $i \in \mathbb{N}_+$ *and any* $\eta > 0$*, there exists a positive integer* $N^* > 0$*, such that if* $N > N^*$*,*

$$|\hat{P}_{0,i} - P_{0,i}| \leq \eta,\ \|\hat{P}_{1,i} - P_{1,i}\|_\infty \leq \eta,\ \|\hat{P}_{2,i} - P_{2,i}\|_\infty \leq \eta$$
$$|\hat{K}_{0,i+1} - K_{0,i+1}| \leq \eta,\ \|\hat{K}_{1,i+1} - K_{1,i+1}\|_\infty \leq \eta. \quad (35)$$

Proof. Define the approximation error as $\tilde{\Upsilon}_i^N = \Upsilon_i^N - \hat{\Upsilon}_i^N$. Subtracting (32) from (31) yields

$$\hat{E}_i = M_i\tilde{\Upsilon}_i^N + E_i. \quad (36)$$

Since $\hat{E}_i^\top \hat{E}_i$ is minimized by the method of least squares,

$$\frac{1}{L}\hat{E}_i^\top \hat{E}_i \leq \frac{1}{L}E_i^\top E_i. \quad (37)$$

Furthermore, combining (36) and (37), we have

$$\frac{1}{L}\tilde{\Upsilon}_i^{N\top} M_i^\top M_i \tilde{\Upsilon}_i^N = \frac{1}{L}(\hat{E}_i - E_i)^\top(\hat{E}_i - E_i) \leq \frac{4}{L}E_i^\top E_i. \quad (38)$$

Therefore, via Assumption 2, the following inequality holds

$$\tilde{\Upsilon}_i^{N\top}\tilde{\Upsilon}_i^N \leq \frac{4}{\alpha L}E_i^\top E_i \leq \frac{4}{\alpha}\max_{1\leq k\leq L} E_{i,k}^2. \quad (39)$$

Then, the lemma is proved by induction. When $i = 1$, $\hat{u}_1 = u_1$, so $\tilde{u}_1 = 0$, and $\rho_{1,k}^0 = \rho_{1,k}^1 = \rho_{1,k}^2 = 0$. Furthermore, since $\epsilon_{1,i}(t)$ $\epsilon_{2,i}(t)$, $\epsilon_{3,i}(t)$, and $\psi_{i,k}$ are induced by the approximation truncation errors in (22), they converge to zero as $N \to \infty$. Therefore, by the expression of $E_{i,k}$ in (29), for any $1 \leq k \leq L$, $E_{i,k}$ converges to zero as $N \to \infty$. Consequently, by (39), for $i = 1$,

$$\lim_{N\to\infty} \tilde{\Upsilon}_i^{N\top}\tilde{\Upsilon}_i^N = 0. \quad (40)$$

As a result, the estimation of the weighting matrices in (22) converge to the true values as $N \to \infty$. By the boundedness of the functions $\Phi(\theta)$, $\Psi(\xi,\theta)$, $\Lambda(\xi,\theta)$ on the compact interval $\theta, \xi \in [-\tau, 0]$ and the uniform convergence of the approximation truncation errors in (23), (35) holds for $i = 1$.

Suppose (35) holds for some $i - 1 > 1$. Then, from the second line of (35), it is seen that $\tilde{u}_i$ converges to zero as $N \to \infty$. Since $\rho_{i,k}^0$, $\rho_{i,k}^1$, and $\rho_{i,k}^2$ are induced by $\tilde{u}_i$, they converge to zero as $N \to \infty$. Furthermore, since $\epsilon_{1,i}(t)$ $\epsilon_{2,i}(t)$, $\epsilon_{3,i}(t)$, 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 $1 \leq k \leq L$, $E_{i,k}$ 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* $N > N^{**}$

$$|\hat{P}_{0,i^*} - P_0^*| \leq \eta,\ \|\hat{P}_{1,i^*} - P_1^*\|_\infty \leq \eta,\ \|\hat{P}_{2,i^*} - P_2^*\|_\infty \leq \eta,$$
$$|\hat{K}_{0,i^*+1} - K_0^*| \leq \eta,\ \|\hat{K}_{1,i^*+1} - K_1^*\|_\infty \leq \eta. \quad (41)$$

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

By Theorem 2, we see that $\hat{K}_{j,i+1}(j = 0, 1)$ obtained by Algorithm 1 converges to $K_j^*(j = 0, 1)$ 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.

Algorithm 1 Data-driven Policy Iteration

1: Choose the vectors of the basis functions Φ, Ψ, and Λ.
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: Calculate $\hat{u}_i(t) = \hat{u}_i(x_t)$ 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 $\hat{\Upsilon}_i^N$ by solving (34).
11: Get $\hat{K}_{0,i+1}$ and $\hat{K}_{1,i+1}$ by (26).
12: $\hat{u}_{i+1}(x_t) = -\hat{K}_{0,i+1}x(t) - \int_{-\tau}^{0} \hat{K}_{1,i+1}(\theta)x_t(\theta)\mathrm{d}\theta$
13: $i \leftarrow i + 1$
14: **until** $|\hat{\Upsilon}_i^N - \hat{\Upsilon}_{i-1}^N| < \delta$.
15: Use $\hat{u}_i(x_t)$ as the control input.

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 $([x(t)]_1 - [x(t-\tau)]_1)$, leading to the time-delay effect. The model is described by (1) with $A \in \mathbb{R}^{2\times 2}$, $A_d \in \mathbb{R}^{2\times 2}$ and $B \in \mathbb{R}^{2\times 1}$ expressed in [41, Section V-A], and $\tau = 1.3s$. The initial admissible controller is $\hat{u}_1(x_t) = -K_{0,1}x(t)$, with $K_{0,1} = [1.74, 3.92]$. The exploration noise is $e(t) = 20\sum_{i=1}^{50}\sin\omega_i t$, where ω_i is randomly sampled from an independent uniform distribution over $[-10, 10]$. $Q = \mathrm{diag}([100, 100])$ and $R = 1$. $\delta = 10^{-3}$. For the basis functions, $\Phi(\theta) = [1, \theta, \theta^2, \theta^3]^\top$, $\Psi(\xi,\theta) = [1, \xi+\theta, \xi^2+\theta^2, \xi\theta, \xi^3+\theta^3, \xi^2\theta+\xi\theta^2, \xi^3\theta+\xi\theta^3, \xi^2\theta^2, \xi^3\theta^2+\xi^2\theta^3, \xi^3\theta^3]^\top$, and $\Lambda(\xi,\theta) = [1, \theta, \theta^2, \theta^3]^\top \otimes [1, \xi, \xi^2, \xi^3]^\top$.

As shown in Fig. 1, the weights of the basis functions $\hat{\Upsilon}$ 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 $J(x_0, \hat{u}_1) = 5.89 \times 10^4$ and $J(x_0, \hat{u}_8) = 3.03 \times 10^4$.

Semi-discretization [48] is applied to discretize (1) into a delay-free system with sampling period $\Delta t = 0.1s$. Then, Algorithm 1 is compared with the model-based discrete-time linear quadratic 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.

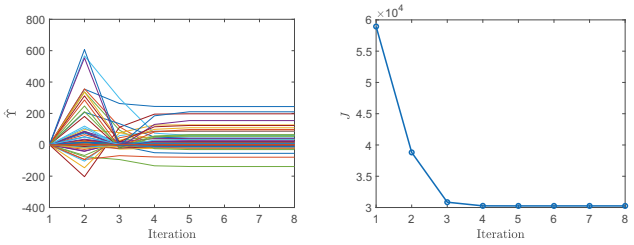

Fig. 1: Evolution of $\hat{\Upsilon}$ and J with respect to iterations.

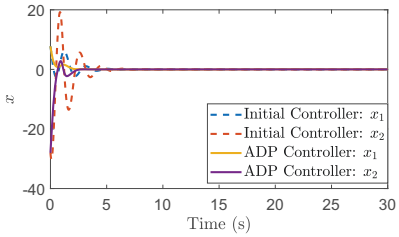

Fig. 2: Compare the initial and ADP controllers for RCMC.

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 semi-discretization, 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 $\varphi(t) \sim \mathcal{N}(0, 0.2)$. 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 $A, A_d \in \mathbb{R}^{4\times4}$ and $B \in \mathbb{R}^{4\times1}$ depicted in [41, Section V-B], and $\tau = 1.2s$. The weighting matrices of the performance index are $Q = \text{diag}([1, 1, 10, 10])$, and $R = 1$. The initial admissible controller is $\hat{u}_1(x_t) = -K_{0,1}x(t)$, with $K_{0,1} = [-0.09, -0.28, -0.30, 0.52]$. The exploration noise is set as $e(t) = \sum_{i=1}^{200} \sin \omega_i t$, where ω_i is sampled from an independent uniform distribution over $[-100, 100]$. The basis functions are the same as in the previous example. $\delta = 10^{-3}$. The analytical expressions of the optimal values K_0^* and K_1^* are derived by [8], where the precise model is assumed known.

The convergence of $\hat{K}_0$ and $\hat{K}_{1,i}$ are shown in Fig. 5. At the last iteration of PI, $\frac{|\hat{K}_{0,10} - K_0^*|}{|K_0^*|} = 0.0008$ and $\frac{||\hat{K}_{1,10} - K_1^*||_\infty}{||K_1^*||_\infty} = 0.0137$. Therefore, the proposed data-driven PI algorithm well approximates the optimal controller. The performance comparisons of the initial controller and the ADP controller are shown in Fig. 6. Since x_1 and x_2 are the states of HDV2, which cannot be influenced by the controller for the AV, they are not plotted in the figure. For the performance indices,

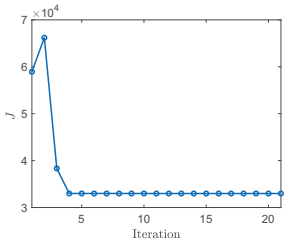

Fig. 3: Evolution of the performance index using noisy data.

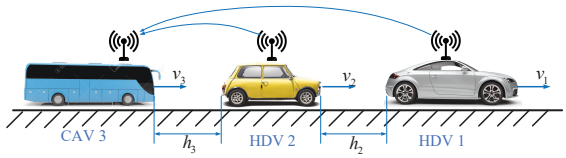

Fig. 4: A platoon of two HDVs and an AV.

$J(x_0, \hat{u}_1) = 1.46 \times 10^5$ and $J(x_0, \hat{u}_{10}) = 4.73 \times 10^4$. Hence, the proposed algorithm minimizes the performance index.

VI. Conclusions

This paper has proposed for the first time a novel data-driven PI algorithm for a class of linear time-delay systems described by DDEs. The first major contribution of this paper 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 paper 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 multi-agent systems.

References

[1] V. Kolmanovskii and A. Myshkis, *Introduction to the Theory and Applications of Functional Differential Equations*. New York, NY: Kluwer Academic Publishers, 1999.
[2] J. K. Hale and S. M. V. Lunel, *Introduction to functional differential equations*. New York, NY: Springer-Verlag, 1993.
[3] I. Karafyllis and Z. P. Jiang, *Stability and Stabilization of Nonlinear Systems*. London, UK: Springer-Verlag, 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_∞ control of distributed delay systems with application to combustion control," *IEEE Trans. Autom. Control*, vol. 46, no. 12, pp. 1930–1935, 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, 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, 2007.
[8] J. I. 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, 2017.

TABLE I: Comparison of Alg. 1 with semi-discretization.

Alg. 1	Model-based DLQR	Discrete-time ADP
$J = 3.0 \times 10^4$	$J = 3.3 \times 10^4$	$J = 4.8 \times 10^4$

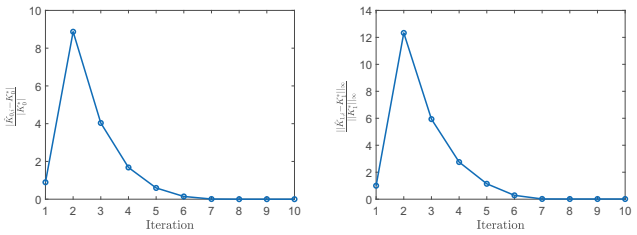

Fig. 5: Convergence of $\hat{K}_{0,i}$ and $\hat{K}_{1,i}(\theta)$ to K_0^* and $K_1^*(\theta)$.

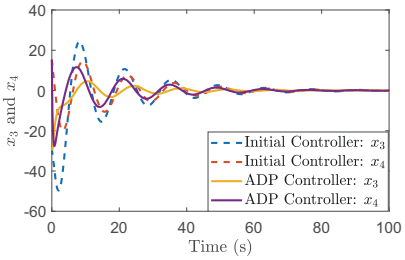

Fig. 6: Compare the initial and ADP controllers for CAVs.

[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, 2022.

[10] G. Stépán, "Delay effects in the human sensory system during balancing," *Phil. Trans. R. Soc. A.*, vol. 367, pp. 1195–212, 04 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. Tamg, H. Gao, W. Zhang, and J. Jurths, "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. 14, no. 6, pp. 678–687, 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 lag systems via a quadratic criterion," *IEEE Trans. Autom. Control*, vol. 16, no. 6, pp. 664–672, 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*. NJ, USA: Wiley-IEEE Press, 2017.

[20] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. NJ, USA: Wiley-IEEE Press, 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, 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, 2021.

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

[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 *60th IEEE Conference on Decision and Control (CDC)*, pp. 77–82, 2021.

[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. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, eds.), pp. 62–77, New York, NY, USA: Springer, 2021.

[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.*, pp. 6536–6541, 2019.

[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," *Journal of the Franklin Institute*, vol. 353, no. 9, pp. 2117–2136, 2016.

[30] H. Zhang, R. 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, 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*, pp. 1–1, 2021.

[33] R. Moghadam, S. Jagannathan, V. Narayanan, and K. Raghavan, *Optimal Adaptive Control of Partially Uncertain Linear Continuous-Time Systems with State Delay*, pp. 243–272. Cham: Springer International Publishing, 2021.

[34] R. Moghadam and S. Jagannathan, "Optimal adaptive control of uncertain nonlinear continuous-time systems with input and state delays," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–10, 2021.

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

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

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

[38] E. Fridman, *Introduction to Time-Delay Systems Analysis and Control*. 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," *arXiv preprint arXiv:2210.00204*, 2022.

[42] Y. Eidelman, V. Milman, and A. Tsolomitis, *Functional Analysis, An Introduction*. Rhode Island, USA: American Mathematical Society, 2004.

[43] C. C. Pugh, *Real Mathematical Analysis, 2nd Edition*. AG Switzerland: Springer International Publishing, 2015.

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

[45] Z. P. Jiang, C. Prieur, and A. Astolfi (Editors), *Trends in Nonlinear and Adaptive Control: A Tribute to Laurent Praly for His 65th Birthday,*. NY, USA: Springer Nature, 2021.

[46] K. J. Åström and B. Wittenmark, *Adaptive control, 2nd Edition*. 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, 06 2005.

[48] T. Insperger and G. Stépán, "Semi-discretization method for delayed systems," *Int. J. Numer. Methods Eng.*, vol. 55, no. 5, pp. 503–518, 2002.