Linear Quadratic Tracking Control of Partially-Unknown Continuous-Time Systems Using Reinforcement Learning

Hamidreza Modares and Frank L. Lewis, *Fellow, IEEE*

***Abstract*—In this technical note, an online learning algorithm is developed to solve the linear quadratic tracking (LQT) problem for partially-unknown continuous-time systems. It is shown that the value function is quadratic in terms of the state of the system and the command generator. Based on this quadratic form, an LQT Bellman equation and an LQT algebraic Riccati equation (ARE) are derived to solve the LQT problem. The integral reinforcement learning technique is used to find the solution to the LQT ARE online and without requiring the knowledge of the system drift dynamics or the command generator dynamics. The convergence of the proposed online algorithm to the optimal control solution is verified. To show the efficiency of the proposed approach, a simulation example is provided.**

***Index Terms*—Causal solution, integral reinforcement learning, linear quadratic tracking, policy iteration, reinforcement learning.**

I. Introduction

A primary objective in control system design is often to seek a stabilizing controller to force the output of a system to follow a reference (desired) trajectory. However, stability is only a bare minimum requirement in a system design. To meet other design specifications, the optimal control theory tries to find a control law that not only stabilizes the error dynamics, but also minimizes a pre-defined performance index. For linear systems accompanied by a quadratic performance index, the optimal tracking problem is called the linear quadratic tracking (LQT) which is an important problem in the field of optimal control theory.

Traditional solutions to the LQT problem are composed of two components; a feedback term obtained by solving an algebraic Riccati equation (ARE) and a feedforward term obtained by either solving a differential equation [1] or calculating a desired control input *a priori* using knowledge of the system dynamics [2]. The feedback term tries to stabilize the tracking error dynamics and the feedforward term tries to guarantee perfect tracking. Procedures for computing the feedback and feedforward terms are traditionally based on offline solution methods which must be done in a noncausal manner backwards in time and require complete knowledge of the system dynamics.

Reinforcement learning (RL) [3]–[6] methods, inspired by learning mechanisms observed in mammals, are concerned with how an agent or actor ought to take actions in an environment so as to optimize a cost of its long-term interactions with the environment. Since there is a strong connection between RL and optimal control [7], [8], a recent objective of control system researchers is to introduce and develop RL techniques that result in optimal controllers for uncertain dynamical systems. A class of RL-based feedback controllers, namely adaptive dynamic programming, was first developed by Werbos [6] for solving optimal regulator problems of discrete-time (DT) systems. An extension of the RL-based controllers to continuous-time (CT) systems was first introduced by Doya [9]. RL algorithms have also been extended to solve the optimal H_∞ control problems [10]. Most of the available RL algorithms for solving optimal control problems are based on the policy iteration (PI) technique. For linear systems which are the focus of this technical note, PI has been effectively used to solve the optimal regulator problem for both DT systems [11]–[13] and CT systems [14]–[16].

Although RL algorithms are widely used to solve the optimal regulator problems, few results considered solving the optimal tracking problem using RL techniques for both DT systems [17]–[21] and CT systems [22]. This is mainly because of the additional computational burden created by computing the feedforward control term that is not presented in the optimal regulator problem. The existing RL solutions to the optimal tracking [17]–[22] employs the dynamic inversion concept to obtain the feedforward control term *a priori* and then find the optimal feedback control term using RL techniques. However, the dynamic inversion method requires complete knowledge of the system dynamics *a priori* to obtain the feedforward control term. Moreover, this method can be only used if the input dynamics part of the system is invertible.

In this technical note, we develop an online adaptive controller based on a PI algorithm, namely the integral reinforcement learning (IRL) [14], [23], which converges to the optimal solution of the LQT problem without knowing the system drift dynamics or the command generator dynamics. The algorithm starts with an admissible nonoptimal control policy and learns an optimal control policy using only measured data from the system and the command generator in real time. To achieve this goal, first, it is shown that the value function is quadratic in terms of the system state and the reference trajectory and an augmented system is constructed from the original system and the command generator. Using the quadratic structure of the value function, a novel Bellman equation and an augmented LQT ARE equation are derived for the LQT problem. This formulation allows extending the IRL technique, a promising method for solving optimal regulation problems for CT partially-unknown systems, to learn the solution to the LQT ARE using only partial knowledge about the system dynamics. Convergence of the proposed learning algorithm to an optimal control solution is verified.

II. Continuous-Time Linear Quadratic Tracking Problem and Its Standard Solution

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

Consider the linear CT system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

Manuscript received March 08, 2013; revised August 15, 2013 and January 2, 2014; accepted April 9, 2014. Date of publication April 11, 2014; date of current version October 21, 2014. This work was supported by the National Science Foundation (NSF) grants ECCS-1128050 and NSF IIS-1208623, ONR grant N00014-13-1-0562, AFOSR EOARD Grant 13-3055, China NNSF grant 61120106011, and China Education Ministry Project 111 (B08015). Recommended by Associate Editor H. L. Trentelman.

The authors are with the Arlington Research Institute, University of Texas, Ft. Worth, TX 76118 USA (e-mail: modares@uta.edu; lewis@uta.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2014.2317301

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

Assumption 1: The pair (A, B) is stabilizable and the pair $(A, \sqrt{Q}C)$ is observable.

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

$$J(x,\bar{y}_d)=\frac{1}{2}\int_t^\infty \left[(Cx-y_d)^TQ(Cx-y_d)+u^TRu\right]d\tau \quad (2)$$

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

The standard solution to the LQT problem is given as [1], [24]

$$u=-R^{-1}B^TSx+R^{-1}B^Tv_{SS} \quad (3)$$

where S is obtained by solving the Riccati equation

$$0=A^TS+SA-SBR^{-1}B^TS+C^TQC \quad (4)$$

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

$$-\dot{v}=\left(A-BR^{-1}B^TS\right)^Tv+C^TQy_d,\quad v(T)=0 \quad (5)$$

The first term of the control input (3) is a feedback control part that depends linearly on the system state, and the second term is a feedforward control part that depends on the reference trajectory. The feedforward part of the control input is time varying in general and thus a theoretical difficulty arises in the solution of the infinite-horizon LQT problem. In [24] and [25], methods for real-time computation of v_{SS} are provided.

Remark 1: Note that the performance function (2) is unbounded if the reference trajectory does not approach zero as time goes to infinity. This is because the feedforward part of the control input and consequently the second term under the integral of the performance function (2) depends on the reference trajectory. Therefore, the standard methods presented in [24] and [25] can be only used if the reference trajectory is generated by an asymptotically stable system.

III. Augmented ARE for Causal Solution of the Infinite-Horizon LQT Problem

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

Assumption 2: Assume that the reference trajectory $y_d(t)$ is generated by the command generator system

$$\dot{y}_d=Fy_d \quad (6)$$

where F is a constant matrix of appropriate dimension.

Remark 2: Matrix F is not assumed stable. The command generator dynamics given in (6) can generate a large class of useful command trajectories, including unit step (useful, e.g., in position command), sinusoidal waveforms (useful, e.g., in hard disk drive control), damped sinusoids (useful, e.g., in vibration quenching in flexible beams), the ramp (useful in velocity tracking systems, e.g., satellite antenna pointing), and more.

As was discussed in Section II, the use of the performance function (2) for the LQT problem requires the command generator be asymptotically stable, i.e., F in (6) must be Hurwitz. In order to relax this restrictive assumption, a discounted value function for the LQR problem is introduced as follows:

$$J(x,\bar{y}_d)=\frac{1}{2}\int_t^\infty e^{-\gamma(\tau-t)}\left[(Cx-y_d)^TQ(Cx-y_d)+u^TRu\right]d\tau \quad (7)$$

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

Definition 1. Admissible Control: A control policy $\mu(x)$ is said to be admissible with respect to (2), if $\mu(x)$ is continuous, $\mu(0)=0$, $u(x)=\mu(x)$ stabilizes (1), and $J(x(t),\bar{y}_d)$ is finite $\forall x(t)$ and $\bar{y}_d$.

Lemma 1. Quadratic Form of the LQT Value Function: Consider the LQT problem with the system dynamics and the reference trajectory dynamics given as (1) and (6), respectively. Consider the admissible fixed control policy

$$u=Kx+K'y_d. \quad (8)$$

Then, the value function (7) for control policy (8) can be written as the quadratic form

$$J(x(t),\bar{y}_d)=V(x(t),y_d(t))=\frac{1}{2}\left[x(t)^Ty_d(t)^T\right]P\left[x(t)^Ty_d(t)^T\right]^T \quad (9)$$

for some symmetric $P>0$.

Proof: Putting (8) in the value function (2) and performing some manipulations yields

$$\begin{aligned}V(x(t),y_d(t))=&\frac{1}{2}\int_0^\infty e^{-\gamma\tau}\\ &\times\left[x(\tau+t)^T(C^TQC+K^TRK)\times x(\tau+t)\right.\\ &+2x(\tau+t)^T(-C^TQ+K^TRK')y_d(\tau+t)\\ &\left.+y_d(\tau+t)^T(Q+K'^TRK')y_d(\tau+t)\right]d\tau. \quad (10)\end{aligned}$$

Using (8), the solutions for the linear differential (1) and (6) become

$$\begin{aligned}x(\tau+t)=&e^{(A+BK)\tau}x(t)\\ &+\left(\int_0^\tau e^{(A+BK)\tau'}BK'e^{F\tau'}d\tau'\right)y_d(t)\\ \equiv&L_1(\tau)x(t)+L_2(\tau)y_d(t) \quad (11)\\ y_d(\tau+t)=&e^{F\tau}y_d(t)\equiv L_3(\tau)y_d(t). \quad (12)\end{aligned}$$

Substituting (11) and (12) in (10) results in

$$V\left(x(t),y_d(t)\right)=\frac{1}{2}\left[x(t)^Ty_d(t)^T\right]P\left[x(t)^Ty_d(t)^T\right]^T \quad (13)$$

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

$$P_{11}=\int_0^\infty e^{-\gamma\tau}L_1\left(\tau\right)^T\left(C^TQC+K^TRK\right)L_1(\tau)d\tau \quad (14)$$

$$\begin{aligned}P_{12}=&\int_0^\infty e^{-\gamma\tau}\left[L_1(\tau)^T(C^TQC+K^TRK)L_2(\tau)\right.\\ &\left.+L_1(\tau)^T(-C^TQ+K^TRK')L_3(\tau)\right]d\tau \quad (15)\end{aligned}$$

$$\begin{aligned}P_{21}=&\int_0^\infty e^{-\gamma\tau}\left[L_2(\tau)^T(C^TQC+K^TRK)L_1(\tau)\right.\\ &\left.+L_3(\tau)^T(-QC+K'^TRK)L_1(\tau)\right]d\tau \quad (16)\end{aligned}$$

$$\begin{aligned}P_{22}=&\int_0^\infty e^{-\gamma\tau}\left[L_3(\tau)^T(Q+K'^TRK')L_3(\tau)\right.\\ &+L_2(\tau)^T(C^TQC+K^TRK)L_2(\tau)\\ &\left.+2L_2(\tau)^T(-C^TQ+K^TRK')L_3(\tau)\right]d\tau. \quad (17)\end{aligned}$$

This completes the proof. □

Note that (9) is valid because Assumptions 2 is imposed. Also, note that because the closed-loop system is stable for an admissible policy, L_1 and L_2 in (14)–(17) are bounded. The boundness of L_3 and consequently the existence of a solution to the LQT problem is discussed in the following remark.

Remark 3: If the reference trajectory is bounded (i.e., if F is stable or marginally stable, e.g., tracking a step or sinusoidal waveform), then L_3 and is bounded for every $\gamma > 0$. However, if the command generator dynamics F in (6) is unstable, then the first and last terms of P_{22} in (17) can be unbounded for some values of γ. More specifically, one can conclude form (17) that P_{22} is bounded if $(F - 0.5\gamma I)$ has all its poles in the left-hand side of the complex plane. Therefore, if F is unstable, we need to know an upper bound of the real part of unstable poles of the F to choose γ large enough to make sure P_{22} is bounded and thus a solution to the LQT exists.

Now define the augmented system state as

$$X(t) = \left[x(t)^T \; y_d(t)^T\right]^T. \tag{18}$$

Putting (1) and (6) together construct the augmented system as

$$\dot{X} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} X + \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} u \equiv TX + B_1 u. \tag{19}$$

The value function (9) in terms of the augmented system state becomes

$$V(X(t)) = \frac{1}{2} X(t)^T P X(t). \tag{20}$$

Using value function (20) for the left-hand side of (7) and differentiating (7) along with the trajectories of the augmented system (19) gives the augmented LQT Bellman equation

$$0 = (TX + B_1 u)^T PX + X^T P(TX + B_1 u) - \gamma X^T PX + X^T C_1^T Q C_1 X + u^T R u \tag{21}$$

where

$$C_1 = [C - I] \tag{22}$$

Consider the fixed control input (8) as

$$u = Kx + K' y_d = K_1 X \tag{23}$$

where $K_1 = [K \; K']$. Putting (20) and (23) into (21), the LQT Bellman equation gives the augmented LQT Lyapunov equation

$$(T + B_1 K_1)^T P + P(T + B_1 K_1) - \gamma P + C_1^T Q C_1 + K_1^T R K_1 = 0. \tag{24}$$

Based on (21), define the Hamiltonian

$$H(X, u, P) = (TX + B_1 u)^T PX + X^T P(TX + B_1 u) - \gamma X^T PX + X^T C_1^T Q C_1 X + u^T R u. \tag{25}$$

Theorem 1. Causal Solution for the LQT Problem: The optimal control solution for the infinite-horizon LQT problem is given by

$$u = K_1 X \tag{26}$$

where

$$K_1 = -R^{-1} B_1^T P \tag{27}$$

and P satisfies the augmented LQT algebraic Riccati equation (ARE)

$$0 = T^T P + PT - \gamma P - P B_1 R^{-1} B_1^T P + C_1^T Q C_1. \tag{28}$$

Proof: A necessary condition for optimality [1] is stationarity condition

$$\frac{\partial H}{\partial u} = B_1^T PX + Ru = 0 \tag{29}$$

which results in control input (26). Substituting (20) and (26) in the LQT Bellman equation (21) yields (28). This completes the proof. □

Lemma 2. Existence of the Solution to the LQT ARE: The LQT ARE (28) has a unique positive semi-definite solution if (A, B) is stabilizable and the discount factor $\gamma > 0$ is chosen such that $F - 0.5\gamma I$ is stable.

Proof: Note that the LQT ARE (28) can be written as

$$0 = (T - 0.5\gamma I)^T P + P(T - 0.5\gamma I) - P B_1 R^{-1} B_1^T P + C_1^T Q C_1. \tag{30}$$

This amounts to an ARE without discount factor and with the system dynamics given by $T - 0.5\gamma I$ and B_1. Therefore, a unique solution to the LQT ARE (30) and consequently the LQT ARE (28) exists if $(T - 0.5\gamma I, B_1)$ is stabilizable. This requires that $(A - 0.5\gamma I, B)$ be stabilizable and $F - 0.5\gamma I$ be stable. However, since (A, B) is stabilizable, then $(A - 0.5\gamma I, B)$ is also stabilizable for any $\gamma > 0$. This completes the proof. □

Remark 4: The fact that $F - 0.5\gamma I$ should be stable to have a solution to the LQT ARE supports the conclusion of Remark 3 for the existence of a solution to the LQT problem. In Remark 3, it is further elaborated how to choose the discount factor to make sure the LQT problem has a solution.

Remark 5: The optimal control input (26) can be written in form of $u = Kx + K' y_d$, as in (23). Therefore, similar to the standard solution given in Section II, the proposed control solution (26) has both feedback feedforward control parts. However, in the proposed method, both control parts are obtained simultaneously by solving an LQT ARE in a causal manner. This causal formulation is a consequence of Assumption 2 and the quadratic form (9), (20).

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

Lemma 3 [1]: For any admissible control policy $u(X)$, let P be the corresponding solution to the Bellman equation (21). Define $u^*(X) = -R^{-1} B_1^T PX$. Then

$$H(X, u, P) = H(X, u^*, P) + (u - u^*)^T R (u - u^*) \tag{31}$$

where H is the Hamiltonian function defined in (25).

Theorem 2. Stability of the LQT ARE Solution: Consider the LQT problem for the system (1) with performance function (7). Suppose that P^* is a smooth positive-definite solution to the tracking LQT ARE (28) and define the control input $u^* = -R^{-1} B_1^T P^* X$. Then, u^* makes $\bar{e}_d(t) \equiv e^{-(\gamma/2)t} e_d(t)$ asymptotically stable.

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

$$\frac{dV(X)}{dt} = (TX + B_1 u)^T PX + X^T P(TX + B_1 u) \tag{32}$$

so that

$$H(X, u, P) = \frac{dV(X)}{dt} - \gamma V(X) + X^T C_1^T Q C_1 X + u^T R u. \tag{33}$$

Suppose now that P^* satisfies the LQT ARE (28). Then, using (31) and since $H(X^*, u^*, P^*) = 0$, one has

$$\frac{dV(X)}{dt} - \gamma V(X) + X^T C_1^T Q C_1 X + u^T R u = (u - u^*)^T R (u - u^*). \tag{34}$$

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

$$\frac{dV(X)}{dt} - \gamma V(X) + X^T \left(C_1^T Q C_1 + K_1^T R K_1\right) X = 0 \quad (35)$$

where K_1 is the control gain obtained by solving the LQT ARE and it is given in (27). Multiplying $e^{-\gamma t}$ to the both sides of (35) and using $V(X) = X^T P X$ gives

$$\frac{d}{dt}(e^{-\gamma t} X^T P X) = -e^{-\gamma t} X^T \left(C_1^T Q C_1 + K_1^T R K_1\right) X \leq 0. \quad (36)$$

Now define the new state $\bar{X}(t) = e^{-(\gamma/2)t} X(t)$ and consider the Lyapunov function $V(\bar{X}) = \bar{X}^T P \bar{X}$. Then using (36) one has

$$\dot{V}(\bar{X}) = -\bar{X}^T \left(C_1^T Q C_1 + K_1^T R K_1\right) \bar{X} < 0. \quad (37)$$

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

Remark 6: Note that a discounted performance function is used in [26, Section 3.6] for optimal tracking control of N-player differential games. However, it does not consider developing a value function in terms of both the state and the desired trajectory and consequently obtaining both feedback and feedforward control inputs simultaneously by solving an LQT ARE.

Remark 7: The discount factor γ and the weight matrix Q in (7) are design parameters and they can be chosen appropriately to make the system state goes to a very small region around zero. The larger the Q is, the more negative the Lyapunov function (37) is and consequently the faster the tracking error decreases. Also, the smaller the discount factor is, the faster the tracking error decreases.

IV. Integral Reinforcement Learning for Solving the LQT Online

In this section, first an offline solution to the LQT ARE is presented. Then, a CT Bellman equation is developed based on integral reinforcement learning (IRL). Based on this, a reinforcement learning (RL) technique is employed to solve the LQT problem online in real time and without the need for the knowledge of drift dynamics of the system $\mathrm{A}x$ and command generator dynamics Fy_d.

A. Offline PI Algorithm for Solving the LQT ARE

The LQT Lyapunov equation (24), which can be solved to evaluate a fixed control policy, is linear in P and is easier to solve than the LQT ARE (28). This is the motivation for introducing an iterative technique to solve the LQT problem. An iterative Lyapunov method for solving the LQT problem is given as follows.

Algorithm 1. Offline policy iteration for solving the LQT problem

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

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

$$\left(T - 0.5\gamma I + B_1 K_1^i\right)^T P^i + P^i \left(T - 0.5\gamma I + B_1 K_1^i\right) + C_1^T Q C_1 + \left(K_1^i\right)^T R \left(K_1^i\right) = 0. \quad (38)$$

Policy improvement: update the control gain using

$$K_1^{i+1} = -R^{-1} B_1^T P^i. \quad (39)$$

Algorithm 1 is an offline algorithm which extends Kleinman's algorithm [27] to the LQT problem. It is shown in [27] that if the initial control policy is stabilizing, then all subsequent control policies will also be stabilizing. Convergence of Kleinman's algorithm to the solution of the ARE is also shown in [27].

B. The Proposed Partially-Unknown IRL Algorithm for Solving the LQT Problem

To obviate the need for complete knowledge of the system dynamics, the IRL algorithm [14], [23] can be extended to the LQT problem. The IRL is a PI algorithm which uses an equivalent formulation of the Lyapunov equation that does not involve the system dynamics. Hence, it is central to the development of model-free RL algorithms for CT systems. To obtain the IRL Bellman equation for the LQT problem, note that for time interval $\Delta t > 0$, the value function (7) satisfies

$$V(X(t)) = \frac{1}{2} \int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + u^T R u\right] d\tau + e^{-\gamma \Delta t} V(X(t + \Delta t)) \quad (40)$$

where C_1 is defined in (22). Using (20) in (40) yields the LQT IRL Bellman equation

$$X(t)^T P X(t) = \int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + u^T R u\right] d\tau + e^{-\gamma \Delta t} X(t + \Delta t)^T P X(t + \Delta t). \quad (41)$$

The first term of (41) is known as the integral reinforcement [14].

Lemma 4. Equivalence of the Lyapunov Equation and the IRL Bellman Equation (41): The LQT IRL Bellman equation (41) and the LQT Lyapunov equation (24) have the same positive semi-definite solution for value function.

Proof: Dividing both sides of (41) by Δt and taking limit yields

$$\lim_{\Delta t \to 0} \frac{e^{-\gamma \Delta t} X(t + \Delta t)^T P X(t + \Delta t) - X(t)^T P X(t)}{\Delta t} + \lim_{\Delta t \to 0} \frac{\int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + u^T R u\right] d\tau}{\Delta t} = 0. \quad (42)$$

By L'Hopital's rule, then

$$\lim_{\Delta t \to 0} \frac{\int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + u^T R u\right] d\tau}{\Delta t} = X(t)^T C_1^T Q C_1 X(t) + u^T R u \quad (43)$$

and also

$$\begin{aligned} &\lim_{\Delta t \to 0} \frac{X(t)^T P X(t) - e^{-\gamma \Delta t} X(t + \Delta t)^T P X(t + \Delta t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \left\{-\gamma e^{-\gamma \Delta t} X(t + \Delta t)^T P X(t + \Delta t) + e^{-\gamma \Delta t} \dot{X}(t + \Delta t)^T \times P X(t + \Delta t) + e^{-\gamma \Delta t} X(t + \Delta t)^T P \dot{X}(t + \Delta t)\right\} \\ &= -\gamma X(t)^T P X(t) + \dot{X}(t)^T P X(t) + X(t)^T P \dot{X}(t). \end{aligned} \quad (44)$$

Using the system dynamics (19) in (44) and putting (43) and (44) in (42) gives the Bellman equation (21). On the other hand, the Bellman equation (21) has the same value function solution as the Lyapunov equation (24) and this completes the proof. □

Using (41) instead of (24) in policy evaluation step of Algorithm 1, the following IRL-based algorithm is obtained.

Algorithm 2. Online IRL algorithm for solving the LQT problem

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

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

$$X(t)^T P^i X(t) = \frac{1}{2} \int_t^{t+\Delta t} e^{-\gamma(\tau - t)} \left[X(t)^T C_1^T Q C_1 X(t) + (u^i)^T R (u^i) \right] d\tau + e^{-\gamma \Delta t} X(t+\Delta t)^T P^i X(t+\Delta t). \quad (45)$$

Policy improvement: update the control input using

$$u^{i+1} = -R^{-1} B_1^T P^i X. \quad (46)$$

The policy evaluation and improvement steps (45) and (46) are repeated until the policy improvement step no longer changes the present policy, thus convergence to the optimal controller is achieved. That is, until $\|P^{i+1} - P^i\| \leq \varepsilon$ is satisfied, where ε is a small constant. Algorithm 2 does not require knowledge of A and F. Note that the method of [16] can be used to avoid knowledge of B.

According to Lemma 4, the IRL Bellman equation (45) in Algorithm 2 has the same value function solution as the Lyapunov equation (38) in Algorithm 1. Therefore, iterating between (45) and (46) in Algorithm 2 is equivalent to iterating between (38) and (39) in Algorithm 1. Thus, similar to Algorithm 1, if the initial control policy is stabilizing in Algorithm 2, then all subsequent control policies will be stabilizing and the algorithm converges to the optimal policy, provided that the unique solution to the IRL Bellman equation (45) is obtained at each iteration. This unique solution can be uniquely determined using the least squares technique under some persistence of excitation (PE) condition, as shown in [14].

Remark 8: The PE condition can be satisfied by injecting a probing noise into the control input. This can cause biased results. However, it was shown in [13] that discounting the performance function can significantly reduce the deleterious effects of probing noise. Moreover, since the probing noise is known *a priori*, one can consider its effect into the IRL Bellman equation, as in [28], to avoid affecting the convergence of the learning process.

Remark 9: The proposed IRL Algorithm 2 has the same structure as the IRL algorithm in [14] for solving the LQR problem. However, in the proposed algorithm, the augmented system state involves the reference trajectory in it and also a discount factor is used in the IRL Bellman equation of Algorithm 2. In fact, using Assumption 2 and developing Lemmas 1 and 4 and Theorem 1 allows us to extend the IRL algorithm presented in [14] to the LQT problem.

Remark 10: The solution for P^i in the policy evaluation step (45) is generally carried out in a least squares (LS) sense. In fact (45) is a scalar equation and P is a symmetric $n \times n$ matrix with $n(n+1)/2$ independent elements and therefore at least $n(n+1)/2$ data sets are required before (45) can be solved using LS. Both batch LS and recursive LS methods can be used to perform policy evaluation step (45).

Remark 11: The proposed policy iteration Algorithm 2 requires an initial admissible policy. If one knows that the system to be control is itself stable, which is true for many cases, then the initial policy can be chosen as $u = 0$ and the admissibility of the initial policy is guaranteed without requiring any knowledge of A. Moreover, if the reference trajectory is bounded, which is true for most real-world applications, no knowledge of F is needed. Otherwise, the initial admissible policy can be obtained by using some knowledge of T. Suppose the system (1) has a nominal model T_N satisfying $T = T_N + \Delta T$, where ΔT is unknown part of T. In this case, one can use robust control methods such as H_∞ control with the nominal model T_N to yield an admissible initial policy. Note that the learning process does not require any knowledge of T. Finally, Algorithm 2 is a policy iteration algorithm and IRL value iteration can be used to avoid the need for an initial admissible policy.

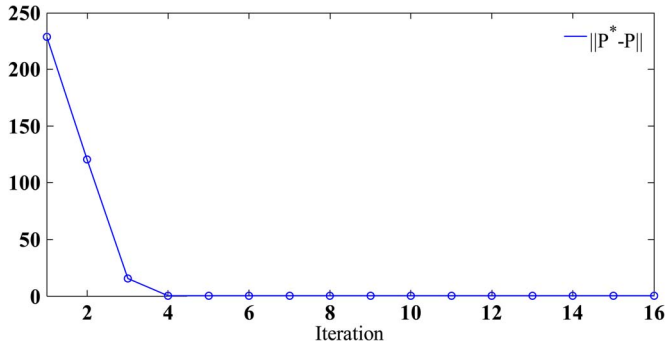

Fig. 1. Convergence of the P matrix parameters to their optimal values.

V. Simulation Results

In this section, an example is provided to verify the correct performance of Algorithm 2 for solving the LQT problem.

Consider the unstable continuous-time linear system

$$\dot{x}(t) = \begin{bmatrix} 0.5 & 1.5 \\ 2.0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 5 \\ 1 \end{bmatrix} u(t), \quad y(t) = [\,1 \quad 0\,]\, x(t) \quad (47)$$

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

$$\dot{y}_d = 0 \quad (48)$$

with the initial value $y_d(0) = 3$. So, the reference trajectory is a step input with amplitude 3. The performance index is given as (2) with $Q = 10$ and $R = 1$ and the discount factor is chosen as $\gamma = 0.1$.

The solution obtained by directly solving the LQT ARE (28) using known dynamics (T, B_1) is given by

$$P^* = \begin{bmatrix} 0.6465 & 0.0524 & -0.6221 \\ 0.0524 & 0.0191 & -0.0244 \\ -0.6221 & -0.0244 & 1.7360 \end{bmatrix} \quad (49)$$

and hence using (27) the optimal control gain becomes

$$K_1^* = [-3.2851 \quad -0.2813 \quad 3.1347]. \quad (50)$$

It is now assumed that the system drift dynamics and the command generator dynamics are unknown and Algorithm 2 is implemented online to solve the LQT problem for the system. The simulation was conducted using data obtained from the augmented system at every 0.05 s. A batch least squares problem is solved after 6 data samples and thus the controller is updated every 0.3 s. The initial control policy is chosen as $K_1 = [-5.0 \ -1.0 \ -0.5]$. Fig. 1 shows how the norm of the difference between the optimal P matrix and the P matrix obtained by the online learning algorithm converges to zero. Also, Fig. 2 depicts the norm of the difference between the optimal control gain and the control gain obtained by the learning algorithm. From Figs. 1 and 2, it is clear that the value function and control gain parameters converge to

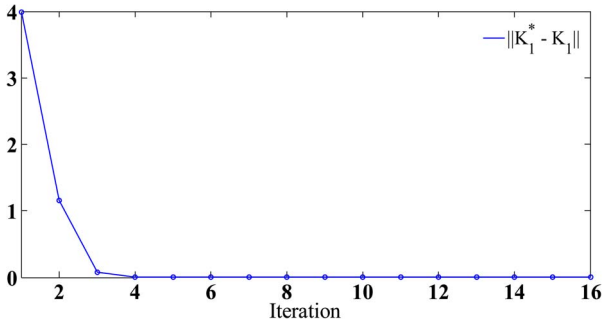

Fig. 2. Convergence of the control gain parameters to their optimal values.

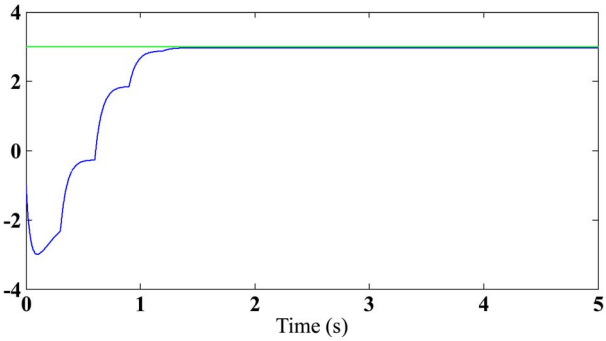

Fig. 3. System output versus reference trajectory.

their optimal values in and after four iterations. Thus, the solution of the LQT ARE is obtained at time $t = 1.2$s. Fig. 3 shows the output and the desired trajectory during simulation. It can be seen that the output tracks the desired trajectory after the optimal control is found.

VI. Conclusion

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

References

[1] F. L. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*, 3rd ed. New York: Wiley, 2012.

[2] A. Mannava, S. N. Balakrishnan, L. Tang, and R. G. Landers, "Optimal tracking control of motion systems," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 6, pp. 1548–1556, Nov. 2012.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning—An Introduction*. Cambridge, MA: MIT Press, 1998.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[5] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley-Interscience, 2007.

[6] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1991, pp. 67–95.

[7] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control," *IEEE Control Syst. Mag.*, pp. 76–105, Dec. 2012.

[8] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Computational Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[9] K. Doya, "Reinforcement learning in continuous-time and space," *Neural Computation*, vol. 12, pp. 219–245, 2000.

[10] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton-Jacobi-Isaacs equation for H_∞ state feedback control with input saturation," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1986–1995, Dec. 2006.

[11] S. J. Bradtke, B. E. Ydestie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. Amer. Control Conf.*, Baltimore, MD, Jun. 1994, pp. 3475–3476.

[12] T. Landelius, "Reinforcement Learning and Distributed Local Model Synthesis," Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden, 1997.

[13] F. L. Lewis and K. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst, Man Cybern. B*, vol. 41, no. 1, pp. 14–23, Feb. 2011.

[14] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, February 2009.

[15] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, Nov. 2012.

[16] Y. Jiang and Z. P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, October 2012.

[17] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man Cybern. B*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[18] T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," in *Proc. Joint 48th IEEE Conf. Decision Control Conf. & 28th Chinese Control Conf.*, Shanghai, China, Dec. 16–18, 2009, pp. 6750–6755.

[19] Q. Wei and D. Liu, "Optimal tracking control scheme for discrete-time nonlinear systems with approximation errors," *Adv. Neural Networks—Lecture Notes Comp. Sci.*, vol. 7952, pp. 1–10, 2013.

[20] R. Song, W. Xiao, and Q. Wei, "Optimal tracking control for a class of nonlinear time-delay systems with actuator saturation," *Adv. Brain Inspired Cognitive Syst.—Lecture Notes Comp. Sci.*, vol. 7888, pp. 208–215, 2013.

[21] Y. Huang and D. Liu, "Neural-network-based optimal tracking control scheme for a class of unknown discrete-time nonlinear systems using iterative ADP algorithm," *Neurocomputing*, vol. 125, pp. 46–56, 2014.

[22] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Networks*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[23] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Network*, vol. 22, pp. 237–246, April 2009.

[24] E. Barbieri and R. Alba-Flores, "On the infinite-horizon LQ tracker," *Syst. Control Lett.*, vol. 40, no. 2, pp. 77–82, Jun. 2000.

[25] E. Barbieri and R. Alba-Flores, "Real-time infinite horizon linear-quadratic tracking controller for vibration quenching in flexible beams," in *Proc. IEEE Conf. Syst., Man, Cybern.*, Taipei, Taiwan, Oct. 8–11, 2006, pp. 38–43.

[26] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. New York: Wiley, 2005.

[27] D. L. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. 18, no. 1, pp. 114–115, Feb. 1968.

[28] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning with explorations for continuous-time nonlinear systems," in *IEEE World Congress on Computational Intelligence*, Brisbane, Australia, Jun. 2012, pp. 10–15.