IET Control Theory & Applications

Research Article

# Finite horizon optimal tracking control of partially unknown linear continuous-time systems using policy iteration

ISSN 1751-8644
Received on 15th December 2014
Revised on 24th February 2015
Accepted on 23rd March 2015
doi: 10.1049/iet-cta.2014.1325
www.ietdl.org

*Chao Li, Derong Liu ✉, Hongliang Li*

*The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190 People's Republic of China*

✉ *E-mail: derongliu@gmail.com*

**Abstract:** In this study, a neural-network-based online learning algorithm is established to solve the finite horizon linear quadratic tracking (FHLQT) problem for partially unknown continuous-time systems. An augmented problem is constructed with an augmented state which consists of the system state and the reference trajectory. The authors obtain a solution for the augmented problem which is equivalent to the standard solution of the FHLQT problem. To solve the augmented problem with partially unknown system dynamics, they develop a time-varying Riccati equation. A critic neural network is used to approximate the value function and an online learning algorithm is established using the policy iteration technique to solve the time-varying Riccati equation. An integral policy iteration method and an online tuning law are used when the algorithm is implemented without the knowledge of the system drift dynamics and the command generator dynamics. A simulation example is given to show the effectiveness of the established algorithm.

## 1 Introduction

The purpose of optimal tracking control is to obtain an optimal control law that minimises the performance index function and forces the plant to track a desired trajectory. The objective in finite horizon controller design is to seek a control law which satisfies such demands over a specified time interval. In the field of optimal control theory [1, 2], the finite horizon linear quadratic tracking (FHLQT) is an important problem. The FHLQT problem tries to find a control law that not only minimises a predefined performance index function, but also tracks a desired reference trajectory and satisfies a final constraint condition over a specified time interval. The standard solution of the optimal control law to the FHLQT problem can be obtained by solving two differential equations backward using the exact system dynamics and boundary conditions. This procedure is a backward-in-time scheme which is not practical for real-time control and is generally an offline method which requires the complete system dynamics. An ideal FHLQT optimal control law using forward-in-time control design and partial knowledge of the system dynamics can overcome this weakness.

Dynamic programming (DP) [3] provides a principled method for determining optimal control policies for dynamic systems. Owing to the nature of exhaustive search, DP is often computationally untenable and it also requires the accurate system representation. Among the methods for solving the optimal control problems, adaptive DP (ADP) has received increasing attention owing to its learning and optimal capabilities [4–16, 44]. Reinforcement learning (RL) is another computational method and it can interactively find an optimal policy [17–20]. The ADP and RL schemes relax the need for a complete and accurate model of the process to be controlled in DP by using compact parameterised function representations whose parameters are adjusted through adaption. In the existing literature of ADP-based optimal control, either policy iteration (PI) or value iteration is utilised to solve the Bellman equation or the Hamilton–Jcaobi–Bellman equation. Liu *et al.* [21] extended the PI algorithm to non-linear optimal control problem with unknown dynamics and discounted cost function. Wang *et al.* [22] investigated a neural-network-based robust optimal control design for a class of uncertain non-linear systems via ADP approach. Wang *et al.* [23] established a novel strategy to design a robust controller for a class of continuous-time non-linear systems with uncertainties. Yang *et al.* [24] developed an adaptive RL-based solution for the infinite-horizon optimal control problem of constrained-input continuous-time non-linear systems in the presence of non-linearities with unknown structures. Vrabie and Lewis [25] derived an integral RL method to obtain direct adaptive optimal control for non-linear input-affine continuous-time systems with partially unknown dynamics. Jiang and Jiang [26] presented a novel PI approach for continuous-time linear systems with completely unknown dynamics. Lee *et al.* [27, 28] presented an integral $Q$-learning algorithm for continuous-time systems without the exact knowledge of the system dynamics. Liu *et al.* [29] developed an online synchronous approximate optimal learning algorithm based on policy iteration to solve a multiplayer non-zero-sum game with unknown dynamics. Li *et al.* [30] established an integral RL (IRL) algorithm to solve two-player zero-sum differential games with completely unknown linear continuous-time dynamics. Liu *et al.* [31] developed a novel online learning optimal control approach to deal with the decentralised stabilisation problem for a class of continuous-time non-linear interconnected systems. The existing RL solutions to the optimal tracking [32–34] employ the dynamic inversion concept to obtain the feedforward control term a priori and find the optimal feedback control term using RL techniques. Near optimal control schemes were developed in [35–37] for linear and non-linear systems over finite horizon by iterative methodology with partial knowledge of the system dynamics. Modares and Lewis [38, 39] developed an online learning algorithm to solve the linear quadratic tracking problem for partially-unknown continuous-time systems. Modares and Lewis [40] extended the IRL technique to solve the solution of the optimal tracking control problem with non-linear partially-unknown constrained-input systems. Kiumarsi and Lewis [41, 42] developed optimal control for linear and non-linear discrete-time systems with unknown dynamics using reinforcement $Q$-learning. Song *et al.* [43] proposed a new optimal tracking control method for a class of complex-valued non-linear systems based on ADP.

Although ADP-based and RL-based algorithms are widely used to solve the optimal regulator problem and the infinite horizon optimal tracking problem, there are few results for the FHLQT problem. Compared with the infinite horizon optimal tracking problem which has a time-invariant solution in [39], the FHLQT problem is more challenging since the solution is time varying and a terminal constraint has to be satisfied. The novelty of this paper is that we establish an online learning algorithm to solve the FHLQT problem with partially unknown system dynamics. We formulate the FHLQT problem into an augmented problem by defining the augmented state which consists of the system state and the reference trajectory. To obtain the augmented solution with partially unknown system dynamics, we develop a time-varying Riccati equation. Using the PI technique, we establish an online learning algorithm to solve the time-varying Riccati equation. To implement this algorithm, a critic neural network (NN) is used to approximate the value function. An integral PI method and a tuning law are implemented to obtain the optimal control policy. The effectiveness of the optimal tracking control law is demonstrated by a simulation example.

The rest of this paper is organised as follows. In Section 2, we present the FHLQT problem and its standard solution. In Section 3, we formulate the FHLQT problem into a related augmented problem and obtain an augmented solution. In Section 4, we establish an online learning algorithm using PI to obtain the augmented solution with partially unknown system dynamics. In Section 5, a simulation example is provided to illustrate the effectiveness of the derived optimal tracking control law. In Section 6, we conclude the paper with a few remarks.

## 2 Problem formulation

Consider the linear time-invariant continuous-time system

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

where $x(t) \in \mathbb{R}^n$ is the measurable system state vector, $u(t) \in \mathbb{R}^r$ is the control input vector, $y(t) \in \mathbb{R}^m$ is the output vector and the matrices $A$, $B$ and $C$ have appropriate dimensionalities. Let $z(t) \in \mathbb{R}^m$ be the desired output.

The objective of the FHLQT problem is to find the optimal control policy $u^*$ to control system (1) in such a way that the system output $y(t)$ tracks the desired output $z(t)$ as close as possible during the interval $[t_0, t_f]$ with minimum expenditure of control effort. For this, we define the error vector as

$$e(t) = z(t) - y(t)$$

and choose the value function as

$$\begin{aligned}V(t) &= \frac{1}{2}e^{\mathsf{T}}(t_f)Fe(t_f)\\ &\quad + \frac{1}{2}\int_t^{t_f}\left[e^{\mathsf{T}}(\tau)Qe(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau\end{aligned} \tag{2}$$

where $t \in [t_0, t_f]$ and $t_f$ is the fixed final time. In (2), $F \in \mathbb{R}^{m\times m}$, $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{r\times r}$ are all positive definite symmetric matrices. We call $e^{\mathsf{T}}(t)Qe(t) + u^{\mathsf{T}}(t)Ru(t)$ is the utility function.

The standard solution of optimal control $u^*$ to the FHLQT problem is given as [1]

$$u^*(t) = -R^{-1}B^{\mathsf{T}}P(t)x^*(t) + R^{-1}B^{\mathsf{T}}g(t) \tag{3}$$

where $x^*(t)$ is the optimal system state, $P(t)$ and $g(t)$ can be obtained by solving the matrix differential Riccati equation (DRE) and the non-homogeneous vector differential equation, respectively

$$\begin{aligned}\dot{P}(t) &= -P(t)A - A^{\mathsf{T}}P(t) + P(t)BR^{-1}B^{\mathsf{T}}P(t) - C^{\mathsf{T}}QC\\ \dot{g}(t) &= -\left[A - BR^{-1}B^{\mathsf{T}}P(t)\right]^{\mathsf{T}}g(t) - C^{\mathsf{T}}Qz(t)\end{aligned} \tag{4}$$

with the terminal conditions $P(t_f) = C^{\mathsf{T}}FC$ and $g(t_f) = C^{\mathsf{T}}Fz(t_f)$.

Substituting the optimal control policy (3) into the system dynamics (1), we can obtain that the optimal state $x^*(t)$ satisfies

$$\dot{x}^*(t) = \left[A - BR^{-1}B^{\mathsf{T}}P(t)\right]x^*(t) + BR^{-1}B^{\mathsf{T}}g(t)$$

with initial condition $x(t_0) = x_0$. Using the optimal state, the optimal value function $V^*(t)$ can be represented as

$$V^*(t) = \frac{1}{2}x^{*\mathsf{T}}(t)P(t)x^*(t) - x^{*\mathsf{T}}(t)g(t) + \frac{1}{2}h(t)$$

where $h(t)$ is the solution of

$$\dot{h}(t) = g^{\mathsf{T}}(t)BR^{-1}B^{\mathsf{T}}g(t) - z^{\mathsf{T}}(t)Qz(t)$$

with the terminal condition $h(t_f) = z^{\mathsf{T}}(t_f)Fz(t_f)$.

By solving the differential equations (4) backward using the boundary conditions, we can obtain the standard solution of the FHLQT problem. Once the system dynamics, reference trajectory and the value function are specified, we can independently compute $P(t)$ and $g(t)$ before the system operates in the forward direction from its initial condition.

*Remark 1:* The feedback and feedforward parts of the control input are calculated in a backward-in-time manner which is not practical for real-time control. The standard solution described in this section is a kind of offline methods which require the complete system dynamics. To obtain the time-varying control input online with partial knowledge of the system dynamics, we construct an augmented problem and establish an online learning algorithm.

## 3 Augmented FHLQT problem and the augmented solution

In this section, we formulate the FHLQT problem into a related augmented problem represented by the augmented state which is made up of the system state and the reference trajectory. The augmented solution is derived using the complete system dynamics. The equivalence between the augmented solution and the standard solution is provided.

### 3.1 Augmented FHLQT problem

We consider the reference trajectory $z(t)$ which is generated by the following linear command generator system

$$\dot{z}(t) = Dz(t)$$

where $D$ is a constant matrix with initial condition $z(t_0) = z_0$. The command generator can generate many useful reference trajectories, such as step signals, sinusoidal waveforms and damped sinusoids.

*Lemma 1:* The FHLQT problem described in Section 2 can be transformed to an augmented problem with a quadratic value function.

*Proof:* The augmented system dynamics can be described as

$$\begin{aligned}\dot{X}(t) &= \begin{bmatrix} A & \mathbf{0}\\ \mathbf{0} & D\end{bmatrix}X(t) + \begin{bmatrix} B\\ \mathbf{0}\end{bmatrix}u(t)\\ &= MX(t) + Nu(t)\end{aligned} \tag{5}$$

where $X(t) = \left[x(t)^{\mathsf{T}},\ z(t)^{\mathsf{T}}\right]^{\mathsf{T}} = [x_1,\ x_2,\ \ldots,\ x_l]^{\mathsf{T}} \in \mathbb{R}^l$ is the augmented system state. We set $l = n + m$.

Using the augmented system state, the first term of the value function (2) can be rewritten as

$$\begin{aligned}\frac{1}{2}e^{\mathsf{T}}(t_{\mathrm{f}})Fe(t_{\mathrm{f}}) &= \frac{1}{2}[z(t_{\mathrm{f}}) - Cx(t_{\mathrm{f}})]^{\mathsf{T}}F[z(t_{\mathrm{f}}) - Cx(t_{\mathrm{f}})] \\ &= \frac{1}{2}X^{\mathsf{T}}(t_{\mathrm{f}})[-C \quad I_m]^{\mathsf{T}}F[-C \quad I_m]X(t_{\mathrm{f}}) \\ &= \frac{1}{2}X^{\mathsf{T}}(t_{\mathrm{f}})L^{\mathsf{T}}FLX(t_{\mathrm{f}})\end{aligned}$$

where $L = [-C,\ I_m]$, and $I_m$ denotes the $m \times m$ identity matrix. The second term of the value function (2) can be rewritten as

$$\begin{aligned}&\frac{1}{2}\int_t^{t_{\mathrm{f}}}\left[e^{\mathsf{T}}(\tau)Qe(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau \\ &= \frac{1}{2}\int_t^{t_{\mathrm{f}}}\left[[z(\tau) - Cx(\tau)]^{\mathsf{T}}Q[z(\tau) - Cx(\tau)] + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau \\ &= \frac{1}{2}\int_t^{t_{\mathrm{f}}}\left[X^{\mathsf{T}}(\tau)L^{\mathsf{T}}QLX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau\end{aligned}$$

On the basis of the above conclusions, the value function (2) with quadratic form can be rewritten as

$$\begin{aligned}V(x(t), t) &= \frac{1}{2}X^{\mathsf{T}}(t_{\mathrm{f}})HX(t_{\mathrm{f}}) \\ &\quad + \frac{1}{2}\int_t^{t_{\mathrm{f}}}\left[X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau \end{aligned}\tag{6}$$

where $H = L^{\mathsf{T}}FL$, $W = L^{\mathsf{T}}QL$. This completes the proof. □

### 3.2 Augmented FHLQT solution

In this subsection, we will derive the augmented solution for the augmented problem according to [1]. A theorem is presented to demonstrate the equivalence between the augmented solution and the standard solution.

Using the definition of the Hamiltonian along with the augmented system (5) and the value function (6), we formulate the Hamiltonian as

$$\begin{aligned}\mathcal{H}(X(t), u(t), \lambda(t)) &= \frac{1}{2}X^{\mathsf{T}}(t)WX(t) \\ &\quad + \frac{1}{2}u^{\mathsf{T}}(t)Ru(t) + \lambda^{\mathsf{T}}(t)[MX(t) + Nu(t)]\end{aligned}$$

where $\lambda(t)$ is the costate vector of $l$-order. For notation simplicity, we use $(\cdot)_*$ to represent that the functions between the parenthesis are optimal ones. For instance, $(\partial\mathcal{H}/\partial u)_* = \partial\mathcal{H}(X^*(t), u^*(t), \lambda^*(t))/\partial u^*(t)$. Differentiating the Hamiltonian with respect to control $u$, we can obtain the optimal control $u^*(t)$ using the control relation as

$$\left(\frac{\partial\mathcal{H}}{\partial u}\right)_* = 0 \Rightarrow Ru^*(t) + N^{\mathsf{T}}\lambda^*(t) = 0$$

leading to

$$u^*(t) = -R^{-1}N^{\mathsf{T}}\lambda^*(t) \tag{7}$$

The optimal state and optimal costate equations can be represented as

$$\begin{aligned}\dot{X}^*(t) &= +\left(\frac{\partial\mathcal{H}}{\partial\lambda}\right)_* \Rightarrow \dot{X}^*(t) = MX^*(t) + Nu^*(t) \\ \dot{\lambda}^*(t) &= -\left(\frac{\partial\mathcal{H}}{\partial X}\right)_* \Rightarrow \dot{\lambda}^*(t) = -WX^*(t) - M^{\mathsf{T}}\lambda^*(t)\end{aligned}$$

We substitute the optima control (7) in the above state and costate equations and obtain a canonical system, also called Hamiltonian system as

$$\begin{bmatrix}\dot{X}^*(t) \\ \dot{\lambda}^*(t)\end{bmatrix} = \begin{bmatrix}M & -NR^{-1}N^{\mathsf{T}} \\ -W & -M^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}X^*(t) \\ \lambda^*(t)\end{bmatrix} \tag{8}$$

The general boundary condition at optimal manner is produced here as

$$\lambda^*(t_{\mathrm{f}}) = \left(\frac{\partial V(t)}{\partial x(t_{\mathrm{f}})}\right)_* = \left(\frac{\partial[\frac{1}{2}X^{\mathsf{T}}(t_{\mathrm{f}})HX(t_{\mathrm{f}})]}{\partial X(t_{\mathrm{f}})}\right)_* = HX^*(t_{\mathrm{f}}) \tag{9}$$

To formulate a closed-loop optimal control, that is, to obtain the optimal control $u^*(t)$ which is a function of the costate $\lambda^*(t)$ as seen from (7), as a function of the state $X^*(t)$, we examine the terminal condition on $\lambda^*(t)$ given by (9). This in fact relates the costate in terms of the state at the final time $t_{\mathrm{f}}$. Similarly, we can connect the costate with the state for the complete interval $[t_0, t_{\mathrm{f}}]$. We assume a transformation

$$\lambda^*(t) = S(t)X^*(t) \tag{10}$$

where $S(t)$ is a positive definite symmetric matrix to be determined and is called the Riccati coefficient matrix. We can easily see that with (10), the optimal control (7) becomes

$$u^*(t) = -R^{-1}N^{\mathsf{T}}S(t)X^*(t)$$

which is now a negative feedback of the optimal state $X^*(t)$. Differentiating (10) with regard to time $t$, we obtain

$$\dot{\lambda}^*(t) = \dot{S}(t)X^*(t) + S(t)\dot{X}^*(t) \tag{11}$$

Using the optimal control (7), the state and costate systems (8), and the transformation (10), we obtain

$$\begin{aligned}\dot{X}^*(t) &= MX^*(t) - NR^{-1}N^{\mathsf{T}}S(t)X^*(t) \\ \dot{\lambda}^*(t) &= -WX^*(t) - M^{\mathsf{T}}S(t)X^*(t)\end{aligned}\tag{12}$$

Substituting the state and costate relations (12) in (11), we have

$$\begin{aligned}&-WX^*(t) - M^{\mathsf{T}}S(t)X^*(t) = \dot{S}(t)X^*(t) \\ &+ S(t)\left[MX^*(t) - NR^{-1}N^{\mathsf{T}}S(t)X^*(t)\right] \\ &\Rightarrow \left[\dot{S}(t) + S(t)M + M^{\mathsf{T}}S(t) + W - S(t)NR^{-1}N^{\mathsf{T}}S(t)\right]X^*(t) = 0\end{aligned}\tag{13}$$

The relation (13) should be satisfied for all time $t \in [t_0, t_{\mathrm{f}}]$ and for any choice of the initial state $X^*(t_0)$. Also, $S(t)$ is not dependent on the initial state and it follows that (13) should hold for any value of $X^*(t)$. This clearly means that the Riccati coefficient matrix $S(t)$ should satisfy the matrix differential equation

$$\dot{S}(t) + S(t)M + M^{\mathsf{T}}S(t) + W - S(t)NR^{-1}N^{\mathsf{T}}S(t) = 0$$

This is the matrix differential equation of the Riccati type, and it is often called the matrix DRE. The matrix DRE can also be written in an equivalent form as

$$\dot{S}(t) = -S(t)M - M^{\mathsf{T}}S(t) - W + S(t)NR^{-1}N^{\mathsf{T}}S(t) \tag{14}$$

Comparing the boundary condition (9) and the Riccati transformation (10), we have the terminal condition on $S(t)$ as

$$\lambda^*(t_{\mathrm{f}}) = S(t_{\mathrm{f}})X^*(t_{\mathrm{f}}) = HX^*(t_{\mathrm{f}}) \Rightarrow S(t_{\mathrm{f}}) = H$$

Before we summarise the solution for the augmented problem, a lemma which makes the value function be a simpler form is presented.

*Lemma 2:* The optimal value function for the augmented system (5) can be represented as

$$V^*(X^*(t),t)=\frac{1}{2}X^{*\mathsf{T}}(t)S(t)X^*(t) \tag{15}$$

*Proof:* For convenience, we omit the optimal symbol '$*$' during the proof procedure. We note that

$$\begin{aligned}\int_t^{t_\mathrm{f}}\frac{\mathrm{d}}{\mathrm{d}\tau}\left[\frac{1}{2}X^{\mathsf{T}}(\tau)S(\tau)X(\tau)\right]\mathrm{d}\tau &= -\frac{1}{2}X^{\mathsf{T}}(t)S(t)X(t)\\ &\quad+\frac{1}{2}X^{\mathsf{T}}(t_\mathrm{f})S(t_\mathrm{f})X(t_\mathrm{f})\end{aligned}$$

Substituting $(1/2)X^{\mathsf{T}}(t_\mathrm{f})S(t_\mathrm{f})X(t_\mathrm{f})$ into the value function (6), we obtain

$$\begin{aligned}V(X(t),t)&=\frac{1}{2}X^{\mathsf{T}}(t)S(t)X(t)\\ &\quad+\frac{1}{2}\int_t^{t_\mathrm{f}}\Big[X^{\mathsf{T}}(\tau)WX(\tau)+u^{\mathsf{T}}(\tau)Ru(\tau)\\ &\quad+\frac{\mathrm{d}}{\mathrm{d}\tau}\left[X^{\mathsf{T}}(\tau)S(\tau)X(\tau)\right]\Big]\mathrm{d}\tau\\ &=\frac{1}{2}X^{\mathsf{T}}(t)S(t)X(t)+\frac{1}{2}\int_t^{t_\mathrm{f}}\Big[X^{\mathsf{T}}(\tau)WX(\tau)\\ &\quad+u^{\mathsf{T}}(\tau)Ru(\tau)+\dot{X}^{\mathsf{T}}(\tau)S(\tau)X(\tau)\\ &\quad+X^{\mathsf{T}}(\tau)\dot{S}(\tau)X(\tau)+X^{\mathsf{T}}(\tau)S(\tau)\dot{X}(\tau)\Big]\mathrm{d}\tau\end{aligned}$$

Now, using (8) for the optimal state $X^*(t)$ and the optimal control $u^*(t)$, we obtain

$$\begin{aligned}V(X(t),t)&=\frac{1}{2}X^{\mathsf{T}}(t)S(t)X(t)+\frac{1}{2}\int_t^{t_f}X^{\mathsf{T}}(\tau)\Big[W+M^{\mathsf{T}}S(\tau)\\ &\quad+S(\tau)M-S(\tau)NR^{-1}N^{\mathsf{T}}S(\tau)+\dot{S}(\tau)\Big]X(\tau)\,\mathrm{d}\tau\end{aligned}$$

Finally, using the matrix DRE (14), the integral part becomes zero. We obtain

$$V^*(X^*(t),t)=\frac{1}{2}X^{*\mathsf{T}}(t)S(t)X^*(t)$$

This completes the proof. □

We summarise the procedure for solving the augmented FHLQT problem as follows.

*Step 1*: Solve the matrix DRE

$$\dot{S}(t)=-S(t)M-M^{\mathsf{T}}S(t)-W+S(t)NR^{-1}N^{\mathsf{T}}S(t)$$

with the terminal condition $S(t_\mathrm{f})=H$.

*Step 2*: Solve the optimal augmented state $\dot{X}^*(t)$ from

$$\dot{X}^*(t)=\left[M-NR^{-1}N^{\mathsf{T}}S(t)\right]X^*(t)$$

with initial condition $X(t_0)=X_0$.

*Step 3*: Obtain the optimal control $u^*(t)$ as

$$u^*(t)=-R^{-1}N^{\mathsf{T}}S(t)X^*(t)$$

*Step 4*: Obtain the optimal value function as

$$V^*(X^*(t),t)=\frac{1}{2}X^{*\mathsf{T}}(t)S(t)X^*(t)$$

Now we introduce a theorem to testify the equivalence between the augmented solution and the standard solution.

*Theorem 1:* The solution of the FHLQR problem for the augmented system is the same as the standard solution described in Section 2.

*Proof:* We rewrite the Riccati coefficient matrix $S(t)$ as

$$S(t)=\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}$$

where $S_{11}(t)\in\mathbb{R}^{n\times n}$, $S_{22}(t)\in\mathbb{R}^{m\times m}$, $S_{12}(t)$ and $S_{21}(t)$ have appropriate dimensions.

Substituting $S(t)$ into the matrix DRE (14), we obtain (see (16))

Dealing with the terminal condition of $S(t)$, we obtain

$$\begin{aligned}\begin{bmatrix}S_{11}(t_\mathrm{f}) & S_{12}(t_\mathrm{f})\\ S_{21}(t_\mathrm{f}) & S_{22}(t_\mathrm{f})\end{bmatrix}&=\begin{bmatrix}-C^{\mathsf{T}}\\ I_m\end{bmatrix}F\begin{bmatrix}-C & I_m\end{bmatrix}\\ &=\begin{bmatrix}C^{\mathsf{T}}FC & -C^{\mathsf{T}}F\\ -FC & F\end{bmatrix}\end{aligned} \tag{17}$$

For $S_{11}(t)$, we have

$$\left.\begin{aligned}\dot{S}_{11}(t)&=-S_{11}(t)A-A^{\mathsf{T}}S_{11}(t)-C^{\mathsf{T}}QC\\ &\quad+S_{11}(t)BR^{-1}B^{\mathsf{T}}S_{11}(t)\\ S_{11}(t_\mathrm{f})&=C^{\mathsf{T}}FC\end{aligned}\right\}\Rightarrow S_{11}(t)=P(t)$$

Now we consider the optimal control $u^*(t)$ for the augmented system

$$\begin{aligned}u^*(t)&=-R^{-1}N^{\mathsf{T}}S(t)X^*(t)\\ &=-R^{-1}\begin{bmatrix}B^{\mathsf{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}\begin{bmatrix}x^*(t)\\ z(t)\end{bmatrix}\\ &=-R^{-1}B^{\mathsf{T}}S_{11}(t)x^*(t)-R^{-1}B^{\mathsf{T}}S_{12}(t)z(t)\end{aligned}$$

$$\begin{aligned}\begin{bmatrix}\dot{S}_{11}(t) & \dot{S}_{12}(t)\\ \dot{S}_{21}(t) & \dot{S}_{22}(t)\end{bmatrix}&=-\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}\begin{bmatrix}A & \mathbf{0}\\ \mathbf{0} & D\end{bmatrix}-\begin{bmatrix}A^{\mathsf{T}} & \mathbf{0}\\ \mathbf{0} & D^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}-\begin{bmatrix}-C^{\mathsf{T}}\\ I_m\end{bmatrix}Q\begin{bmatrix}-C & I_m\end{bmatrix}\\ &\quad+\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}\begin{bmatrix}B\\ \mathbf{0}\end{bmatrix}R^{-1}\begin{bmatrix}B^{\mathsf{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}\\ &\Longrightarrow\begin{cases}\dot{S}_{11}(t)=-S_{11}(t)A-A^{\mathsf{T}}S_{11}(t)-C^{\mathsf{T}}QC+S_{11}(t)BR^{-1}B^{\mathsf{T}}S_{11}(t)\\ \dot{S}_{12}(t)=-S_{12}(t)D-A^{\mathsf{T}}S_{12}(t)+C^{\mathsf{T}}Q+S_{11}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)\\ \dot{S}_{21}(t)=-S_{21}(t)A-D^{\mathsf{T}}S_{21}(t)+QC+S_{21}(t)BR^{-1}B^{\mathsf{T}}S_{11}(t)\\ \dot{S}_{22}(t)=-S_{22}(t)D-D^{\mathsf{T}}S_{22}(t)-Q+S_{21}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)\end{cases}\end{aligned} \tag{16}$$

For the second term of this formulation, we assume that

$$f(t) = -S_{12}(t)z(t)$$

Differentiating both sides of this equation and substituting (16) and the relation $S_{11}(t) = P(t)$, we have

$$\begin{aligned}
\dot{f}(t) &= -\dot{S}_{12}(t)z(t) - S_{12}(t)\dot{z}(t)\\
&= -\big[-S_{12}(t)D - A^{\mathsf{T}}S_{12}(t) + C^{\mathsf{T}}Q\\
&\quad + S_{11}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)\big]z(t) - S_{12}(t)Dz(t)\\
&= A^{\mathsf{T}}S_{12}(t)z(t) - C^{\mathsf{T}}Qz(t) - S_{11}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)z(t)\\
&= -A^{\mathsf{T}}f(t) - C^{\mathsf{T}}Qz(t) + S_{11}(t)BR^{-1}B^{\mathsf{T}}f(t)\\
&= -\Big[A - BR^{-1}B^{\mathsf{T}}S_{11}(t)\Big]^{\mathsf{T}} f(t) - C^{\mathsf{T}}Qz(t)
\end{aligned}$$

The terminal condition satisfies

$$f(t_{\mathrm{f}}) = -S_{12}(t_{\mathrm{f}})z(t_{\mathrm{f}}) = C^{\mathsf{T}}Fz(t_{\mathrm{f}})$$

Compared with the standard solution, we notice that the vector $f(t) = g(t)$. As a result, the optimal control $u^*(t)$ is equal to the standard solution $u^*(t)$ presented in (3).

Substituting (16) and the relations $S_{11}(t) = P(t)$ and $g(t) = -S_{12}(t)z(t)$ into the augmented optimal value function (15), we have

$$\begin{aligned}
V^*(X^*(t), t) &= \frac{1}{2}X^{*\mathsf{T}}(t)S(t)X^*(t)\\
&= \frac{1}{2}\begin{bmatrix} x^{*\mathsf{T}}(t) & z^{\mathsf{T}}(t)\end{bmatrix}\begin{bmatrix} S_{11}(t) & S_{12}(t)\\ S_{21}(t) & S_{22}(t)\end{bmatrix}\begin{bmatrix} x^*(t)\\ z(t)\end{bmatrix}\\
&= \frac{1}{2}x^{*\mathsf{T}}(t)S_{11}(t)x^*(t) + \frac{1}{2}x^{*\mathsf{T}}(t)S_{12}(t)z(t)\\
&\quad + \frac{1}{2}z^{\mathsf{T}}(t)S_{21}(t)x^*(t) + \frac{1}{2}z^{\mathsf{T}}(t)S_{22}(t)z(t)\\
&= \frac{1}{2}x^{*\mathsf{T}}(t)S_{11}(t)x^*(t) - x^{*\mathsf{T}}(t)g(t)\\
&\quad + \frac{1}{2}z^{\mathsf{T}}(t)S_{22}(t)z(t)
\end{aligned}$$

We assume that $o(t) = z^{\mathsf{T}}(t)S_{22}(t)z(t)$. Differentiating $o(t)$ with respect to $t$ and substituting (16) and the relation $g(t) = -S_{12}(t)z(t)$, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}[o(t)] &= \frac{\mathrm{d}}{\mathrm{d}t}[z^{\mathsf{T}}(t)S_{22}(t)z(t)]\\
&= \dot{z}^{\mathsf{T}}(t)S_{22}(t)z(t) + z^{\mathsf{T}}(t)\dot{S}_{22}(t)z(t) + z^{\mathsf{T}}(t)S_{22}(t)\dot{z}(t)\\
&= z^{\mathsf{T}}(t)D^{\mathsf{T}}S_{22}(t)z(t) + z^{\mathsf{T}}(t)S_{22}(t)Dz(t)\\
&\quad + z^{\mathsf{T}}(t)\big[-S_{22}(t)D - D^{\mathsf{T}}S_{22}(t) - Q\\
&\quad + S_{21}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)\big]z(t)\\
&= z^{\mathsf{T}}(t)S_{21}(t)BR^{-1}B^{\mathsf{T}}S_{12}(t)z(t) - z^{\mathsf{T}}(t)Qz(t)\\
&= [S_{12}(t)z(t)]^{\mathsf{T}}BR^{-1}B^{\mathsf{T}}[S_{12}(t)z(t)] - z^{\mathsf{T}}(t)Qz(t)\\
&= g^{\mathsf{T}}(t)BR^{-1}B^{\mathsf{T}}g(t) - z^{\mathsf{T}}(t)Qz(t)
\end{aligned}$$

The terminal condition satisfies

$$o(t_{\mathrm{f}}) = z^{\mathsf{T}}(t_{\mathrm{f}})S_{22}(t_{\mathrm{f}})z(t_{\mathrm{f}}) = z^{\mathsf{T}}(t_{\mathrm{f}})Fz(t_{\mathrm{f}})$$

Compared with the standard solution, we notice that the vector $o(t) = h(t)$. The optimal value function $V^*(t)$ is equal to the standard one in Section 2. We have the conclusion that the solution of the FHLQR problem for augmented system is the same as the standard solution. This completes the proof. □

*Remark 2:* The fact that the matrix $S(t)$ is symmetric for all $t \in [t_0, t_{\mathrm{f}}]$, that is, $S(t) = S^{\mathsf{T}}(t)$ can easily be shown. First, from the formulation of the augmented problem we note that, the matrices $H$, $W$ and $R$ are symmetric and therefore the matrix $NR^{-1}N^{\mathsf{T}}$ is also symmetric. Now transposing both sides of the matrix DRE (14), we notice that both $S(t)$ and $S^{\mathsf{T}}(t)$ are solutions of the same differential equation and satisfy the same terminal condition (9).

*Remark 3:* Typically, we compute $S(t)$ backward in an offline manner and store them separately, and feed these stored values when the system is operating in the forward direction in the interval $t \in [t_0, t_{\mathrm{f}}]$. In this procedure, we need the exact knowledge of the system matrices $M$ and $N$ to obtain the optimal control policy.

*Remark 4:* We do not need the controllability condition on the system for solving the optimal feedback control. As long as we deal with a finite time system, the contribution of those uncontrollable states to the value function is still a finite quantity.

## 4 Online learning algorithm and its implementation

In this section, we establish an NN-based online learning algorithm to obtain the solution of augmented FHLQT problem with partially unknown system dynamics. Compared with the infinite horizon problem, a time-varying Riccati equation is developed. The online algorithm consists of an online integral PI method and an online tuning law for different time intervals of the time-varying Riccati equation.

For the system dynamics (5), we consider a value function with infinite horizon

$$\Lambda(t) = \frac{1}{2}\int_t^{\infty}[X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)]\,\mathrm{d}\tau \tag{18}$$

According to the optimal control theory [1], the optimal control with respect to this value function is given by

$$\mu^*(t) = -R^{-1}N^{\mathsf{T}}\bar{S}X(t) \tag{19}$$

where $\bar{S} \in \mathbb{R}^{l\times l}$ is a constant positive definite symmetric matrix, which is the solution of the non-linear matrix algebraic Riccati equation (ARE)

$$\bar{S}M + M^{\mathsf{T}}\bar{S} + W - \bar{S}NR^{-1}N^{\mathsf{T}}\bar{S} = 0 \tag{20}$$

Using the constant matrix $\bar{S}$, the value function can be represented in a quadratic form as

$$\Lambda(t) = \frac{1}{2}X^{\mathsf{T}}(t)\bar{S}X(t) \tag{21}$$

Now we consider the relationship between the solution of the matrix DRE (14) and the solution of the ARE (20). We make a simple time transformation $\tau = t_{\mathrm{f}} - t$. Then, in $\tau$ scale we can consider the final time $t_{\mathrm{f}}$ as the 'starting time', $S(t_{\mathrm{f}})$ as the 'initial condition' and $\bar{S}$ as the 'steady-state solution' of the matrix DRE. As $t_{\mathrm{f}} \to \infty$, the 'transient solution' is pushed to near $t_{\mathrm{f}}$ which is at infinity. Then for the beginning time interval, the matrix $S(t)$ becomes a steady state, that is, a constant matrix $\bar{S}$ which is the solution of the ARE (20), as shown in Fig. 1.

According to Fig. 1, the matrix $S(t)$ in DRE becomes the constant matrix $\bar{S}$ during the steady-state interval. We give the following Riccati equation to solve $S(t)$ during the interval $[t_0, t_{\mathrm{f}}]$

$$S(t) = \begin{cases}\bar{S}, & t \in [t_0, t_1)\\ S(t), & t \in [t_1, t_{\mathrm{f}}]\end{cases}$$

where $t_1$ is the terminal time of the steady-state interval. We will establish the online integral PI method to calculate the steady-state

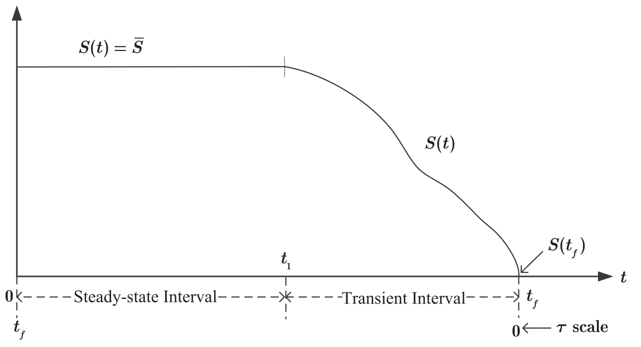

**Fig. 1** *Interpretation of the constant matrix $\bar{S}$*

interval solution $\bar{S}$ and the online tuning law to solve the transient interval solution $S(t)$.

### 4.1 Solution for steady-state interval using online integral PI

In this subsection, we discuss the implementation of the online integral PI method during the steady-state interval. $\bar{S}$ is the solution of the ARE (20). To obviate the need for the complete knowledge of the system dynamics, the IRL algorithm [25] can be used to solve the ARE. The IRL is a PI method which uses an equivalent formulation of the Lyapunov equation that does not involve the system dynamics. Hence, it is central to the development of the online integral PI method for continuous-time systems. To obtain the IRL Bellman equation, noting that for time interval $\Delta t > 0$, the value function satisfies

$$\Lambda(t) = \Lambda(t+\Delta t) + \frac{1}{2}\int_t^{t+\Delta t} [X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)]\,\mathrm{d}\tau$$

The expression (21) yields the IRL Bellman equation

$$X(t)^{\mathsf{T}}\bar{S}X(t) - X^{\mathsf{T}}(t+\Delta t)\bar{S}X(t+\Delta t) = \int_t^{t+\Delta t}[X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)]\,\mathrm{d}\tau \tag{22}$$

The last term of (22) is known as the integral reinforcement.

Now we give a lemma to state the equivalence of the ARE (20) and the IRL Bellman equation (22).

*Lemma 3:* The IRL Bellman equation (22) and the ARE (20) have the same solution $\bar{S}$.

*Proof:* Considering the IRL Bellman equation (22), the derivative of the value function (21) along the trajectory of the system can be calculated as

$$\begin{aligned}\lim_{\Delta t\to 0}\frac{1}{2}\frac{X^{\mathsf{T}}(t+\Delta t)\bar{S}X(t+\Delta t) - X(t)^{\mathsf{T}}\bar{S}X(t)}{\Delta t} &= \lim_{\Delta t\to 0} -\frac{1}{2}\frac{\int_t^{t+\Delta t}\left[X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\right]\mathrm{d}\tau}{\Delta t} \\ &= -\frac{1}{2}\left[X^{\mathsf{T}}(t)WX(t) + u^{\mathsf{T}}(t)Ru(t)\right] = \dot{\Lambda}(t)\end{aligned} \tag{23}$$

The derivative of the value function (21) can also be calculated as

$$\dot{\Lambda}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}X(t)^{\mathsf{T}}\bar{S}X(t)\right] = \frac{1}{2}\left[\dot{X}(t)^{\mathsf{T}}\bar{S}X(t) + X(t)^{\mathsf{T}}\bar{S}\dot{X}(t)\right] \tag{24}$$

Equations (23) and (24) yield the following relation

$$\dot{X}^{\mathsf{T}}(t)\bar{S}X(t) + X^{\mathsf{T}}(t)\bar{S}\dot{X}(t) = -[X^{\mathsf{T}}(t)WX(t) + u^{\mathsf{T}}(t)Ru(t)] \tag{25}$$

Substituting the optimal control (19) into (25), we have

$$\begin{aligned}&[MX(t)+Nu(t)]^{\mathsf{T}}\bar{S}X(t) + X(t)^{\mathsf{T}}\bar{S}[MX(t)+Nu(t)] \\ &\quad = -X^{\mathsf{T}}(t)WX(t) - X^{\mathsf{T}}(t)\bar{S}NR^{-1}N^{\mathsf{T}}\bar{S}X(t)\end{aligned}$$

and after some basic manipulations, we obtain the following equation

$$\bar{S}M + M^{\mathsf{T}}\bar{S} + W - \bar{S}NR^{-1}N^{\mathsf{T}}\bar{S} = 0$$

which is the ARE in (20). The proof is completed. □

Equation (22) which is derived from (18) and (19) plays an important role in relaxing the assumption of knowing the system dynamics, since $M$ does not appear in the equation. It means that the algorithm can be implemented without knowing the system dynamics $M$, but the knowledge of $N$ is still required.

*Remark 5:* We solve the linear quadratic problem over infinite horizon to obtain the 'steady-state solution' $\bar{S}$ in this subsection. The admissibility of control is required to guarantee the existence of $\bar{S}$. So an admissible control is needed to implement this online integral PI method.

We will discuss the NN-based implementation of the established online integral PI method. A critic NN is used to approximate the value function. We assume that for the system, $\Lambda(t)$ is represented on a compact set $\Omega$ by single-hidden-layer NN as

$$\Lambda(t) = \frac{1}{2}X(t)^{\mathsf{T}}\bar{S}X(t) = \frac{1}{2}s^{\mathsf{T}}\chi(t)$$

where

$$\begin{aligned}s^{\mathsf{T}} &= [s_{11}, s_{12}, \ldots, s_{1l}, s_{22}, s_{23}, \ldots, s_{l-1,l}, s_{ll}] \\ \chi^{\mathsf{T}}(t) &= [x_1^2, 2x_1x_2, \ldots, 2x_1x_l, x_2^2, \ldots, 2x_{l-1}x_l, x_l^2]\end{aligned}$$

$s_{ij}$ is the $i$th-row and $j$th-column elements of $\bar{S}$, $s \in \mathbb{R}^{[\{l(l+1)\}/2]}$ is unknown bounded ideal weight parameters which will be determined by the established integral PI method, and $\chi(t) \in \mathbb{R}^{[\{l(l+1)\}/2]}$ is the continuously differentiable activation functions. Since the ideal weights are unknown, the outputs of the critic NN is

$$\Lambda^{(i)}(t) = \frac{1}{2}(\hat{s}^i)^{\mathsf{T}}\chi(t) = \Lambda(t) - \varepsilon^i \tag{26}$$

where $\hat{s}^i$ is the current estimated weight vector and $\varepsilon^i \in \mathbb{R}$ is the bounded NN approximation errors.

Using the expression (26), (22) can be rewritten in a general form

$$\psi_k^{\mathsf{T}}\hat{s}^i = \theta_k \tag{27}$$

with

$$\begin{aligned}\theta_k &= \int_{t+(k-1)\Delta t}^{t+k\Delta t}\left[X^{\mathsf{T}}(\tau)WX(\tau) + u^{(i)\mathsf{T}}(\tau)Ru^{(i)}(\tau)\right]\mathrm{d}\tau \\ \psi_k &= \chi(t+(k-1)\Delta t) - \chi(t+k\Delta t)\end{aligned}$$

where the measurement time is from $t+(k-1)\Delta t$ to $t+k\Delta t$, $\Delta t$ is the time interval. Since (27) is only a one-dimensional equation, we cannot guarantee the uniqueness of the solution. Similar to [27], we use the least-square-based method to solve the parameter vector over a compact set $\Omega$. For any positive integer $K$, we denote

$\Phi = [\psi_1, \psi_2, \ldots, \psi_K]$ and $\Theta = [\theta_1, \theta_2, \ldots, \theta_K]^{\mathsf{T}}$. Then, we have the following $K$-dimensional equation

$$\Phi^{\mathsf{T}} \hat{s}^i = \Theta$$

If $\Phi^{\mathsf{T}}$ has full column rank, the weight parameters can be solved by

$$\hat{s}^i = (\Phi\Phi^{\mathsf{T}})^{-1}\Phi\Theta \tag{28}$$

Therefore we need to guarantee that the number of collected points $K$ satisfies $K \geq \operatorname{rank}(\Phi) = [\{l(l+1)\}/2]$, which will make $(\Phi\Phi^{\mathsf{T}})^{-1}$ exist. The least squares problem in (28) can be solved in real time by collecting enough data points generated from the system.

### 4.2 Solution for transient interval using online tuning law

In this subsection, we will derive an online tuning law to obtain the solution $S(t)$ of the DRE with the terminal condition $S(t_{\mathrm{f}}) = J$ during time interval $[t_1, t_{\mathrm{f}}]$. We assume that the value function $V(t)$ is represented by single-layer NN as

$$V(t) = \frac{1}{2}X(t)^{\mathsf{T}}S(t)X(t) = \frac{1}{2}s^{\mathsf{T}}(t)\chi(t)$$

We define the ideal time-varying weights of the critic network

$$s^{\mathsf{T}}(t) = [s_{11}, s_{12}, \ldots, s_{1l}, s_{22}, s_{23}, \ldots, s_{l-1,l}, s_{ll}]$$

where we omit the time $t$ in the elements of $S(t)$.

When we consider the time-varying function $S(t)$ for the Bellman equation (22), there is a residual error caused by the estimated value function. We assume that $S(t)$ is a constant matrix during the time interval $[t, t+\Delta t]$. The residual error can be expressed as

$$\begin{aligned} e_1(t) &= X^{\mathsf{T}}(t+\Delta t)S(t)X(t+\Delta t) - X(t)^{\mathsf{T}}S(t)X(t) \\ &\quad + \int_t^{t+\Delta t} [X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)]\,\mathrm{d}\tau \end{aligned}$$

By defining the expressions

$$\begin{aligned} \theta(t) &= \int_t^{t+\Delta t} \Big[X^{\mathsf{T}}(\tau)WX(\tau) + u^{\mathsf{T}}(\tau)Ru(\tau)\Big]\,\mathrm{d}\tau \\ \psi(t) &= \chi(t) - \chi(t+\Delta t) \end{aligned}$$

the residual error $e_1(t)$ can be rewritten as

$$e_1(t) = \theta(t) - \psi^{\mathsf{T}}(t)\hat{s}(t)$$

Next, the terminal constraint $S(t_{\mathrm{f}}) = H$ need to be satisfied. The constraint error is given as

$$e_2(t) = j - \hat{s}(t)$$

where $j$ is defined as

$$j^{\mathsf{T}} = [j_{11}, j_{12}, \ldots, j_{1l}, j_{22}, j_{23}, \ldots, j_{l-1,l}, j_{ll}]$$

In order to minimise both the residual error and the constraint error, we give the following online parameters tuning law

$$\hat{s}(t+\Delta t) = \hat{s}(t) + \alpha\frac{\psi(t)e_1(t)}{\psi^{\mathsf{T}}(t)\psi(t)+1} + \alpha\frac{e_2(t)}{(1+t_{\mathrm{f}}-t)^c} \tag{29}$$

where $\alpha$ is the learning rate satisfying $0 < \alpha < 1$, and $c$ is a predefined positive constant.

*Theorem 2:* The parameters update law of the value function is given by (29). Within the finite time interval $t \in [t_1, t_{\mathrm{f}}]$, there exists a positive constant learning rate $0 < \alpha < 1$ such that the value function parameter estimation error is bounded.

*Proof:* We consider the following Lyapunov function candidate given by

$$\Pi(t) = \tilde{s}^{\mathsf{T}}(t)\tilde{s}(t)$$

where $\tilde{s}(t) = s(t) - \hat{s}(t)$. Using this expression, we have

$$\begin{aligned} e_1(t) &= \psi^{\mathsf{T}}(t)s(t) - \psi^{\mathsf{T}}(t)\hat{s}(t) = \psi^{\mathsf{T}}(t)\tilde{s}(t) \\ e_2(t) &= j - [s(t) - \tilde{s}(t)] = j - s(t) + \tilde{s}(t) \end{aligned}$$

We define $\tilde{s}(t+\Delta t) = s(t) - \hat{s}(t+\Delta t)$ and obtain

$$\begin{aligned} \tilde{s}(t+\Delta t) &= \tilde{s}(t) + \hat{s}(t) - \hat{s}(t+\Delta t) \\ &= \tilde{s}(t) - \alpha\frac{\psi(t)e_1(t)}{\psi^{\mathsf{T}}(t)\psi(t)+1} - \alpha\frac{e_2(t)}{(1+t_{\mathrm{f}}-t)^c} \end{aligned}$$

Then using online parameter tuning law (29), the first difference of $\Pi(t)$ can be derived as

$$\begin{aligned} \Delta\Pi(t) &= \tilde{s}^{\mathsf{T}}(t+\Delta t)\tilde{s}(t+\Delta t) - \tilde{s}^{\mathsf{T}}(t)\tilde{s}(t) \\ &= \tilde{s}^{\mathsf{T}}(t)\tilde{s}(t) - 2\alpha\frac{\tilde{s}^{\mathsf{T}}(t)\psi(t)e_1(t)}{\psi^{\mathsf{T}}(t)\psi(t)+1} - 2\alpha\frac{\tilde{s}^{\mathsf{T}}(t)e_2(t)}{(1+t_{\mathrm{f}}-t)^c} \\ &\quad + \alpha^2\frac{\psi^{\mathsf{T}}(t)\psi(t)e_1^2(t)}{[\psi^{\mathsf{T}}(t)\psi(t)+1]^2} + \alpha^2\frac{e_2^{\mathsf{T}}(t)e_2(t)}{(1+t_{\mathrm{f}}-t)^{2c}} \\ &\quad + 2\alpha^2\frac{\psi^{\mathsf{T}}(t)e_1(t)e_2(t)}{[\psi^{\mathsf{T}}(t)\psi(t)+1](1+t_{\mathrm{f}}-t)^c} - \tilde{s}^{\mathsf{T}}(t)\tilde{s}(t) \\ &\leq -\alpha(1-\alpha)\left[\frac{\psi^{\mathsf{T}}(t)\psi(t)}{\psi^{\mathsf{T}}(t)\psi(t)+1} + \frac{1}{(1+t_{\mathrm{f}}-t)^c}\right]\tilde{s}^{\mathsf{T}}(t)\tilde{s}(t) \\ &\quad + 2\alpha^2\frac{\psi^{\mathsf{T}}(t)\psi^{\mathsf{T}}(t)\tilde{s}(t)[j - s(t) + \tilde{s}(t)]}{[\psi^{\mathsf{T}}(t)\psi(t)+1](1+t_{\mathrm{f}}-t)^c} - 2\alpha\frac{\tilde{s}^{\mathsf{T}}(t)[j - s(t)]}{(1+t_{\mathrm{f}}-t)^c} \\ &\quad + \alpha^2\frac{[j - s(t)]^{\mathsf{T}}[j - s(t)] + 2[j - s(t)]^{\mathsf{T}}\tilde{s}(t)}{(1+t_{\mathrm{f}}-t)^{2c}} \\ &\leq -\alpha(1-\alpha)\left(\frac{\Psi}{\Psi+1} + \Xi\right)\tilde{s}^{\mathsf{T}}(t)\tilde{s}(t) + 2\alpha^2\frac{\|j - s(t) + \tilde{s}(t)\|^2}{(1+t_{\mathrm{f}}-t)^c} \end{aligned}$$

where $\Psi = \min_{t\in[t_1,t_{\mathrm{f}}]}[\psi^{\mathsf{T}}(t)\psi(t)]$, $\Xi = [1/\{[(1+t_{\mathrm{f}}-t_1)^c\}]$. Since the learning rate $\alpha$ is selected as $0 < \alpha < 1$, the first term of $\Delta\Pi(t)$ is negative, and the second term $\Upsilon = 2\alpha^2[\{\|j - s(t) + \tilde{s}(t)\|^2\}/\{(1+t_{\mathrm{f}}-t)^c\}]$ is bounded. Using standard Lyapunov stability theory, the value function parameter estimation error can be proven to be bounded with a bound which is dependent upon initial condition of the system and the fixed final time instant $t_{\mathrm{f}}$.

Assume that the initial value function parameter estimation error is bounded such that $\|\tilde{s}(t_1)\|^2 \leq \Gamma_0$. According to standard Lyapunov stability theory, value function parameter estimation error at time $t$ can be expressed as

$$\begin{aligned} \Pi(t) &= \Delta\Pi(t) + \Delta\Pi(t-\Delta t) + \cdots + \Delta\Pi(t_1) + \Pi(t_1) \\ &= \sum_{i=0}^{N_t-1}\Delta\Pi(t_1 + i\Delta t) + \Pi(t_1) \end{aligned}$$

where

$$N_t = \left\lceil\frac{t - t_1}{\Delta t}\right\rceil, \quad \lceil x \rceil$$

is the ceiling operation represents the smallest integer not less than $x$. Note that $\Delta t$ is a small sampling interval. The bound for the value function parameter estimation error $\Gamma_t$ can be expressed as

$$\begin{aligned}\Gamma_t &= \|\tilde{s}(t)\|^2 = \Pi(t) = \sum_{i=0}^{N_t-1} \Delta\Pi(t_1 + i\Delta t) + \Pi(t_1)\\ &\le \sum_{i=0}^{N_t-1}[-\beta(1-\beta)^i\Pi(t_1)] + \sum_{i=1}^{N_t-1}[\beta(1-\beta)^{i-1}\Upsilon] + \Pi(t_1)\\ &\le -\beta\frac{1-(1-\beta)^{N_t}}{\beta}\Pi(t_1) + \Pi(t_1) + \beta\frac{1-(1-\beta)^{N_t-1}}{\beta}\Upsilon\\ &\le (1-\beta)^{N_t}\Gamma_0 + [1-(1-\beta)^{N_t-1}]\Upsilon\end{aligned}$$

where $\beta = \alpha(1-\alpha)\,[[\Psi/(\Psi+1)] + \Xi]$. Since $0 < \alpha < 1$, we know $0 < \beta < 1$. The value function estimation error $\Gamma_t$ is dependent upon initial bound $\Gamma_0$ and $\Upsilon$.

The proof is completed. □

We have already obtained the Riccati coefficient matrix $S(t)$ during the interval $t \in [t_0, t_f]$ using the online integral PI method and the online tuning law. Next, we will describe the online learning algorithm which can be used to solve the augmented FHLQT problem with partially unknown system dynamics.

**Algorithm 1 Online learning algorithm**

```
Part I: Steady-state interval
1: Give a small positive real number $\epsilon$. Let $i = 0$ and start
   with an initial $\bar{S}^{(0)}$ which makes the control policy $u^{(0)}(t)$ be admissible.
2: Policy evaluation:
   On the basis of the Riccati coefficients $\bar{S}^{(i)}$ and control policy
   $u^{(i)}(t)$, solve the following Bellman equation for $\bar{S}^{(i+1)}$
       $X(t)^{\mathsf{T}}\bar{S}^{(i+1)}X(t) - X(t+\Delta t)^{\mathsf{T}}\bar{S}^{(i+1)}X(t+\Delta t)$
       $= \int_t^{t+\Delta t}[X^{\mathsf{T}}(\tau)WX(\tau) + u^{(i)\mathsf{T}}(\tau)Ru^{(i)}(\tau)]\,d\tau$
3: Policy improvement:
   Update the control policy using
       $u^{(i+1)}(t) = -R^{-1}N^{\mathsf{T}}\bar{S}^{(i+1)}X(t)$
4: If $\|\bar{S}^{(i+1)} - \bar{S}^{(i)}\| \le \epsilon$, set $t_1 = t$, obtain the steady-state solution,
   and go to Part II; else, set $i = i + 1$ and go to Step 2.

Part II: Transient interval
1: Start with $\bar{S}$ when $t = t_1$.
2: Policy evaluation:
   On the basis of the Riccati coefficients online tuning law,
   update $S(t+\Delta t)$ using
       $\hat{s}(t+\Delta t) = \hat{s}(t) + \alpha\frac{\psi(t)e_1(t)}{\psi^{\mathsf{T}}(t)\psi(t)+1} + \alpha\frac{e_2(t)}{(1+t_f-t)^c}$
3: Policy improvement:
   Update the control policy using
       $u(t+\Delta t) = -R^{-1}N^{\mathsf{T}}S(t+\Delta t)X(t+\Delta t)$
4: Repeat Step 2 and Step 3 while $t < t_f$.
```

*Remark 6:* This algorithm is a kind of PI algorithms which consist of policy evaluation and policy improvement. For the two different time intervals, the policy evaluation is implemented using (22) and (29), and the policy improvement is implemented using (19) where the knowledge of system dynamics $N$ is required. In [26], Jiang and Jiang presented a novel approach for continuous-time linear systems with completely unknown dynamics. Note that the method can be used to avoid the knowledge of $N$.

*Remark 7:* The convergence of the optimal solution can be obtained under the persistence of excitation (PE) condition. The PE condition can be satisfied by injecting a known probing noise into the control input. As in [28], one can consider the effect of the noise into the IRL Bellman equation to avoid affecting the convergence of the learning process.

## 5 Simulation

In this section, we provide a simulation example to demonstrate the effectiveness of the online learning algorithm. Compared with the standard solution, the algorithm derived in Section 4 is implemented online without the knowledge of $M$. We use this algorithm to obtain the feedback control law and plot all the time histories of optimal states and control.

We consider the following second-order example to illustrate the linear quadratic tracking control. A second-order plant

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= -2x_1(t) - 3x_2(t) + u(t)\\ y(t) &= x_1(t)\end{aligned} \tag{30}$$

is to be controlled to minimise the following value function

$$V(t) = [e(t_f)]^2 + \int_t^{t_f}\left([e(\tau)]^2 + 0.002[u(\tau)]^2\right)d\tau \tag{31}$$

The initial condition $x(0) = [-0.5,\ 0]^{\mathsf{T}}$. The final time $t_f$ is specified at 20 s and the final state $x(t_f)$ is free. It is required to keep the output $y(t)$ close to the reference trajectory $z(t) = \cos t$. $z(t)$ is generated by the command generator system $\dot{c}(t) = -c(t)i$ where $i^2 = -1$ with the initial value $c(0) = 1$. The value function indicates that the state $x_1(t)$ is to be kept close to the reference trajectory.

We identify the various matrices in the present tracking system by comparing state (30) and the value function (31) with the corresponding (1) and (2), respectively, of the general formulation of the problem described in Section 2, we obtain

$$A = \begin{bmatrix} 0 & 1\\ -2 & -3\end{bmatrix},\quad B = \begin{bmatrix}0\\1\end{bmatrix},\quad C = \begin{bmatrix}1 & 0\end{bmatrix}$$
$$F = 2,\quad Q = 2,\quad R = 0.004$$

According to the standard solution, we use complete system dynamics to solve the following differential equations

$$\begin{aligned}\dot{P}_{11} &= 250P_{12}^2 + 4P_{12} - 2\\ \dot{P}_{12} &= 250P_{12}P_{22} - P_{11} + 3P_{12} + 2P_{22}\\ \dot{P}_{22} &= 250P_{22}^2 - 2P_{12} + 6P_{22}\\ \dot{g}_1 &= (250P_{12} + 2)g_2 - 2\cos t\\ \dot{g}_2 &= -g_1 + (250P_{22} + 3)g_2\end{aligned}$$

with the terminal conditions $P_{11}(20) = 2$, $P_{12}(20) = 0$, $P_{22}(20) = 0$, $g_1(20) = 2\cos(20)$ and $g_2(20) = 0$, where

$$P(t) = \begin{bmatrix}P_{11}(t) & P_{12}(t)\\ P_{12}(t) & P_{22}(t)\end{bmatrix},\quad g(t) = \begin{bmatrix}g_1(t)\\ g_2(t)\end{bmatrix}$$

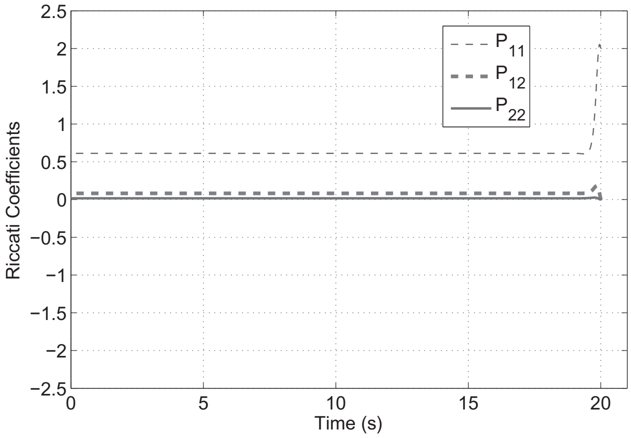

**Fig. 2** *Standard solution of Riccati coefficients P(t)*

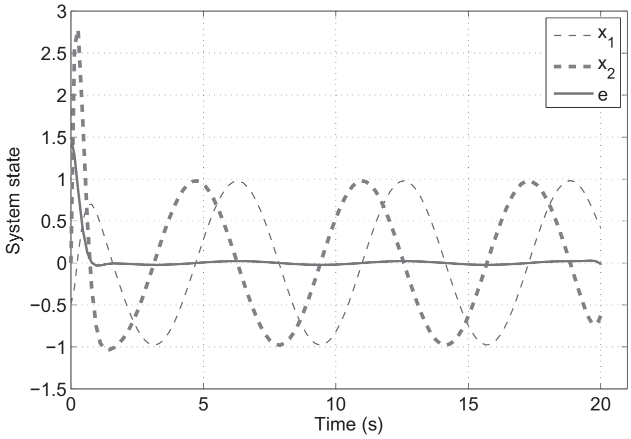

**Fig. 3** *Standard solution of system state x(t) and tracking error e(t)*

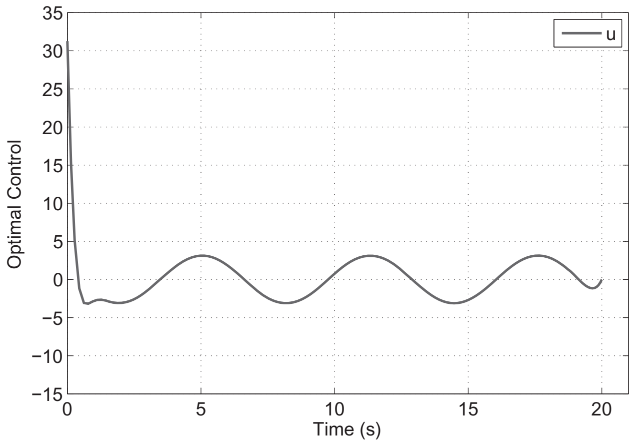

**Fig. 4** *Standard solution of the optimal control u(t)*

Figs. 2–4 illustrate the evolutions of the Riccati coefficients, system states and optimal control, respectively. It is clear that the state $x_1(t)$ can track the reference trajectory.

Now we assume that the system drift dynamics and the command generator dynamics are unknown, that is, we cannot use the knowledge of $M$ when the online learning algorithm is applied. Algorithms 1 is implemented online to solve the augmented FHLQT problem. Compared with the augmented formulation (5)

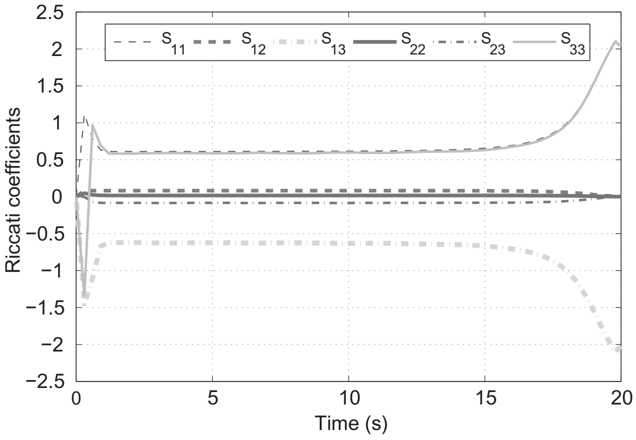

**Fig. 5** *Augmented solution of Riccati coefficients S(t)*

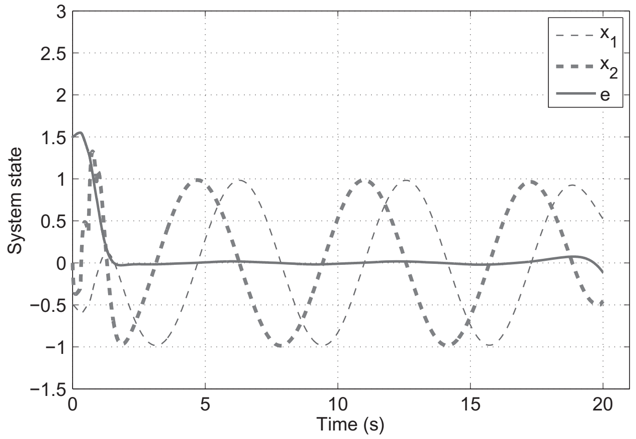

**Fig. 6** *Augmented solution of system state x(t) and tracking error e(t)*

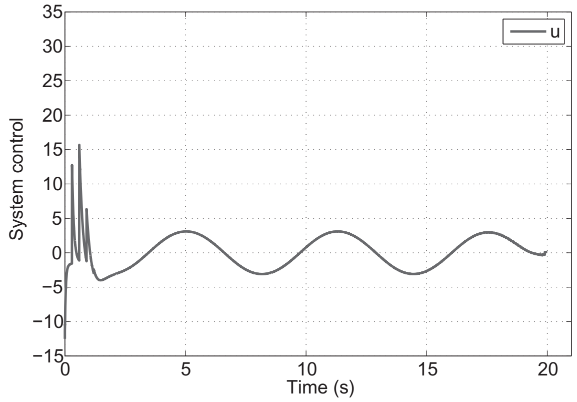

**Fig. 7** *Augmented solution of system control u(t)*

and (6), the corresponding matrices can be represented as

$$M = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -3 & 0 \\ 0 & 0 & -i \end{bmatrix}, \quad N = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad L = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$J = \begin{bmatrix} 2 & 0 & -2 \\ 0 & 0 & 0 \\ -2 & 0 & 2 \end{bmatrix} = W, \quad R = 0.004$$

The $3 \times 3$ symmetric Riccati coefficient matrix $S(t)$ can be represented as

$$S(t) = \begin{bmatrix} s_{11}(t) & s_{12}(t) & s_{13}(t) \\ s_{12}(t) & s_{22}(t) & s_{23}(t) \\ s_{13}(t) & s_{23}(t) & s_{33}(t) \end{bmatrix}$$

The activation functions are chosen as

$$\chi^{\mathsf{T}}(t) = [x_1^2, 2x_1x_2, 2x_1x_3, x_2^2, 2x_2x_3, x_3^2]$$

The weight parameters of the critic NN are represented as

$$s^{\mathsf{T}}(t) = [s_{11}(t), s_{12}(t), s_{13}(t), s_{22}(t), s_{23}(t), s_{33}(t)]$$

Using the online integral PI method, we solve the 'steady-state solution' $\bar{S}$ of the matrix DRE. To implement this algorithm, we let the integer $K = 6$, the time period $\Delta t = 0.05$ s and the initial weights as $s^{(0)\mathsf{T}} = [1,\ 0,\ 6,\ 0.1,\ 0.1,\ 1]$. The least squares problem is solved after six samples are acquired, and the weights of the critic NN are updated every 0.3 s. It is clear that the weights approximately converge to the steady ones after six updates at $t = 1.8$ s in Fig. 5.

To implement the online parameters tuning law, we let the time period $\Delta t = 0.1$ s, learning rate $\alpha = 0.6$ and the constant $c = 4$. We obtain the near optimal solution $S(t)$ of the matrix DRE during the time interval $[1.8, 20]$ s. The system states and control are obtained at the same time interval. Figs. 5–7 illustrate the evolutions of the Riccati coefficients, system states and optimal control with partially system dynamics. It is clear that using the derived algorithm the state $x_1(t)$ can track the reference trajectory during the simulation.

## 6 Conclusion

A neural-network-based online learning algorithm was established using PI to solve the FHLQT problem for partially unknown linear time-invariant continuous-time systems. On the basis of the augmented system, the augmented solution which is equivalent to the standard solution of the FHLQT problem was obtained. Compared with the infinite horizon problem, the time-varying Riccati equation was developed to obtain the augmented solution with partially unknown system dynamics. The online learning algorithm consists of an online integral PI method and an online tuning law for different time intervals of the time-varying Riccati equation. A simulation example was given to show the efficiency of the proposed algorithm.

## 7 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, 61304086 and 61374105, in part by Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of SKLMCCS.

## 8 References

1 Lewis, F.L., Vrabie, D., Syrmos, V.: 'Optimal control' (Wiley, 2012, 3rd edn.)
2 Naidu, D.S.: 'Optimal control systems' (CRC Press, 2003)
3 Bellman, R.E.: 'Dynamic programming' (Princeton University Press, 1957)
4 Wang, F.-Y., Zhang, H., Liu, D.: 'Adaptive dynamic programming: an introduction', *IEEE Comput. Intell. Mag.*, 2009, **4**, (2), pp. 39–47
5 Zhang, H., Wei, Q., Liu, D.: 'An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games', *Automatica*, 2001, **47**, (1), pp. 207–214
6 Liu, D., Wei, Q.: 'Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems', *IEEE Trans. Cybern.*, 2013, **43**, (2), pp. 779–789
7 Wang, F.-Y., Jin, N., Liu, D., Wei, Q.: 'Adaptive dynamic programming for finite horizon optimal control of discrete-time nonlinear systems with $\epsilon$-error bound', *IEEE Trans. Neural Netw.*, 2011, **22**, (1), pp. 24–36
8 Liu, D., Javaherian, H., Kovalenko, O., Huang, T.: 'Adaptive critic learning techniques for engine torque and air-fuel ratio control', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2008, **38**, (4), pp. 988–993
9 Jiang, Y., Jiang, Z.-P.: 'Robust adaptive dynamic programming for large-scale systems with an application to multimachine power systems', *IEEE Trans. Circuits Syst. II, Express Briefs*, 2012, **59**, (10), pp. 693–697
10 Zhao, D., Zhang, Z., Dai, Y.: 'Self-teaching adaptive dynamic programming for Gomoku', *Neurocomputing*, 2012, **78**, (1), pp. 23–29
11 Jiang, Y., Jiang, Z.-P.: 'Robust adaptive dynamic programming with an application to power systems', *IEEE Trans. Neural Netw. Learn. Syst.*, 2013, **24**, (7), pp. 1150–1156
12 Yang, X., Liu, D., Huang, Y.: 'Neural-network-based online optimal control for uncertain nonlinear continuous-time systems with control constraints', *IET Control Theory Appl.*, 2013, **7**, (17), pp. 2037–2047
13 Wang, D., Liu, D.: 'Neuro-optimal control for a class of unknown nonlinear dynamic systems using SN-DHP technique', *Neurocomputing*, 2013, **121**, pp. 218–225
14 Liu, D., Li, H., Wang, D.: 'Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm', *Neurocomputing*, 2013, **110**, pp. 92–100
15 Li, H., Liu, D.: 'Optimal control for discrete-time affine nonlinear systems using general value iteration', *IET Control Theory Appl.*, 2012, **6**, (18), pp. 2725–2736
16 Liu, D., Wang, D., Yang, X.: 'An iterative adaptive dynamic programming algorithm for optimal control of unknown discretetime nonlinear systems with constrained inputs', *Inf. Sci.*, 2013, **220**, pp. 331–342
17 Lewis, F.L., Vrabie, D.: 'Reinforcement learning and adaptive dynamic programming for feedback control', *IEEE Circuits Syst. Mag.*, 2009, **9**, (3), pp. 32–50
18 Sutton, R.S., Barto, A.G.: 'Reinforcement learning: an introduction' (Cambridge University Press, 1998), vol. 1
19 Si, J., Barto, A.G., Powell, W.B., Wunsch, D.C.: 'Handbook of learning and approximate dynamic programming' (John Wiley & Sons, 2004)
20 Bradtke, S.J., Ydstie, B.E., Barto, A.G.: 'Adaptive linear quadratic control using policy iteration'. American Control Conf., Baltimore, MD, USA, June 1994, pp. 3475–3479
21 Liu, D., Yang, X., Li, H.: 'Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics', *Neural Comput. Appl.*, 2013, **23**, (7–8), pp. 1843–1850
22 Wang, D., Liu, D., Li, H., Ma, H.: 'Neural-network-based robust optimal control design for a class of uncertain nonlinear systems via adaptive dynamic programming', *Inf. Sci.*, 2014, **282**, pp. 167–179
23 Wang, D., Liu, D., Li, H.: 'Policy iteration algorithm for online design of robust control of a class of continuous-time nonlinear systems', *IEEE Trans. Autom. Sci. Eng.*, 2014, **11**, (2), pp. 627–632
24 Yang, X., Liu, D., Wang, D.: 'Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints', *Int. J. Control*, 2014, **87**, (2), pp. 553–566
25 Vrabie, D., Lewis, F.: 'Neural network approach to continuoustime direct adaptive optimal control for partially unknown nonlinear systems', *Neural Netw.*, 2009, **22**, (3), pp. 237–246
26 Jiang, Y., Jiang, Z.-P.: 'Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics', *Automatica*, 2012, **48**, (10), pp. 2699–2704
27 Lee, J.Y., Park, J.B., Choi, Y.H.: 'Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems', *Automatica*, 2012, **48**, (10), pp. 2850–2859
28 Lee, J.Y., Park, J.B., Choi, Y.H.: 'Integral reinforcement learning with explorations for continuous-time nonlinear systems'. Int. Joint Conf. on Neural Networks, Brisbane, QLD, Australia, June 2012, pp. 1042–1047
29 Liu, D., Li, H., Wang, D.: 'Online synchronous approximate optimal learning algorithm for multiplayer nonzero-sum games with unknown dynamics', *IEEE Trans. Syst. Man Cybern. Syst.*, 2014, **44**, (8), pp. 1015–1027
30 Li, H., Liu, D., Wang, D., Yang, X.: 'Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics', *IEEE Trans. Autom. Sci. Eng.*, 2014, **11**, (3), pp. 706–714
31 Liu, D., Wang, D., Li, H.: 'Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach', *IEEE Trans. Neural Netw. Learn. Syst.*, 2014, **25**, (2), pp. 418–428
32 Dierks, T., Jagannathan, S.: 'Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics'. 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf., Shanghai, China, December 2009, pp. 6750–6755
33 Huang, Y., Liu, D.: 'Neural-network-based optimal tracking control scheme for a class of unknown discrete-time nonlinear systems using iterative ADP algorithm', *Neurocomputing*, 2014, **125**, pp. 46–56
34 Zhang, H., Cui, L., Zhang, X., Luo, Y.: 'Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method', *IEEE Trans. Neural Netw.*, 2011, **22**, (12), pp. 2226–2236
35 Heydari, A., Balakrishnan, S.N.: 'Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics', *IEEE Trans. Neural Netw. Learn. Syst.*, 2013, **24**, (1), pp. 145–157
36 Nguyen, T., Gajic, Z.: 'Finite horizon optimal control of singularly perturbed systems: a differential Lyapunov equation approach', *IEEE Trans. Autom. Control*, 2010, **55**, (9), pp. 2148–2152
37 Wang, D., Liu, D., Wei, Q.: 'Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach', *Neurocomputing*, 2012, **78**, (1), pp. 14–22

38 Modares, H., Lewis, F.L.: ‘Online solution to the linear quadratic tracking problem of continuous-time systems using reinforcement learning’. 52nd IEEE Annual Conf. on Decision and Control, Firenze, Italy, December 2013, pp. 3851–3856

39 Modares, H., Lewis, F.L.: ‘Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning’, *IEEE Trans. Autom. Control*, 2014, **59**, (11), pp. 3051–3056

40 Modares, H., Lewis, F.L.: ‘Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning’, *Automatica*, 2014, **50**, (7), pp. 1780–1792

41 Kiumarsi, B., Lewis, F.L., Modares, H., Karimpour, A., Naghibi-Sistani, M.: ‘Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics’, *Automatica*, 2014, **50**, (4), pp. 1167–1175

42 Kiumarsi, B., Lewis, F.L.: ‘Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems’, *IEEE Trans. Neural Netw. Learn. Syst.*, 2015, **26**, (1), pp. 140–151

43 Song, R., Lewis, F.L., Wei, Q., Xiao, W.: ‘Optimal tracking control for a class of continuous time complex-valued systems based on adaptive dynamic programming algorithm’, 33rd Chinese Control Conf., Nanjing, China, July 2014, pp. 8974–8978

44 Vamvoudakis, K.G., Lewis, F.L.: ‘Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem’, *Automatica*, 2010, **46**, (5), pp. 878–888