# Integral $Q$-Learning & Explorized Policy Iteration for Adaptive Optimal Control of Continuous-Time Linear Systems [⋆]

Jae Young Lee [a], Jin Bae Park [a,∗], Yoon Ho Choi [b]

[a] *Department of Electrical and Electronic Engineering, Yonsei University, 5o Yonsei-ro, Seodaemun-gu, Seoul, Korea*

[b] *Department of Electronic Engineering, Kyonggi University, 94-6 Yiui-dong, Yeongtong-gu, Suwon, Kyonggi-Do, Korea*

**Abstract**

This paper proposes an integral $Q$-learning for continuous-time (CT) linear time-invariant (LTI) systems, which solves a linear quadratic regulation (LQR) problem in real time for a given system and a value function, without knowledge about the system dynamics $A$ and $B$. Here, $Q$-learning is referred to as a family of reinforcement learning methods which find the optimal policy by interaction with an uncertain environment. In the evolution of the algorithm, we first develop an explorized policy iteration (PI) method which is able to deal with known exploration signals. Then, the integral $Q$-learning algorithm for CT LTI systems is derived based on this PI and the variants of $Q$-functions derived from the singular perturbation of the control input. The proposed $Q$-learning scheme evaluates the current value function and the improved control policy at the same time, and are proven stable and convergent to the LQ optimal solution, provided that the initial policy is stabilizing. For the proposed algorithms, practical online implementation methods are investigated in terms of persistency of excitation (PE) and explorations. Finally, simulation results are provided for the better comparison and the verification of the performance.

*Key words:* $Q$-learning, adaptive control, LQR, policy iteration, optimization under uncertainties

## 1 Introduction

In engineering terminology, reinforcement learning (RL) is a class of biologically-inspired computational methods to improve the agent's action in a given uncertain environment (Powell, 2007; Si, Barto, Powell, & Wunsch, 2004; Sutton & Barto, 1998). It adjusts the agent's current action by interacting with the environment: first it observes the rewards from the environment, and then, modifies the action based on the observed information to maximize its current and future rewards. This procedure is exactly the same as and actually comes from the learning mechanisms of mammals—they interact with the environment and modifies their own actions accordingly to improve their received rewards, leading to better survival chances. These RL algorithms are investigated at first for a finite Markov decision process (MDP) (Kaelbling & Moore, 1996; Sutton & Barto, 1998), and later, for continuous-time (CT) and discrete-time (DT) dynamic systems in both control and machine learning communities (Balakrishnan, Ding, & Lewis, 2008; Si *et al.*, 2004; Lewis & Vrabie, 2009; Wang, Zhang, & Liu, 2009). These RL methods, also known as approximate dynamic programming or adaptive critics, overcome 'the curse of dimensionality' of traditional dynamic programming by forward-time iteration (Si *et al.*, 2004), and are considered as *adaptive optimal control* (Lewis & Vrabie, 2009; Si *et al.*, 2004) or model-predictive control scheme (Bertsekas, 2005; Lee & Lee, 2004; Zhang, Huang, & Lewis, 2009) in control engineering perspectives. By employing such RL methods, one can obtain the optimal policy in an uncertain noisy environment with less computational burden.

Among the RL methods, $Q$-learning, first proposed by Watkins (1989) in a finite MDP framework, has been recognized as one of the most promising and widely used RL methods in various fields of engineering (Powell, 2007; Sutton & Barto, 1998; Wang, Zhang, & Liu, 2009). For $Q$-learning in a finite MDP, the convergence to the optimal policy and its corresponding action value function was given by Watkins & Dayan (1992) with a sufficient number of explorations. Inspired by $Q$-learning for a finite MDP, $Q$-learning schemes for uncertain DT dynamic systems, also known as action-dependent heuristic dynamic programming (ADHDP), are investigated by many researchers (Al-Tamimi,

[⋆] This paper was not presented at any IFAC.

[∗] Corresponding author. Tel.: +82 2 2123 2773; fax: +82 2 362 4539.

*Email addresses:* jyounglee@yonsei.ac.kr (Jae Young Lee), jbpark@yonsei.ac.kr (Jin Bae Park ), yhchoi@kyonggi.ac.kr (Yoon Ho Choi).

Abu-Kalaf, & Lewis, 2007; Balakrishnan *et al.*, 2008; Bertsekas & Tsitsiklis, 1996; Bradtke & Ydstie, 1994; Landelius, 1997; Lewis & Vrabie, 2009; Lewis & Vamvoudakis, 2010; Prokhorov & Wunsch, 1997; Si *et al.*, 2004; Wang, Zhang, & Liu, 2009; Webos, 1992). However, many of the early $Q$-learning methods for DT dynamic systems did not guarantee the convergence to the optimal solution (Prokhorov & Wunsch, 1997; Si *et al.*, 2004; Wang *et al.*, 2009; Webos, 1992). To solve this problem, researches on developing convergence-guaranteed $Q$-learning are carried out for DT linear quadratic regulation (LQR) problems (Bradtke & Ydstie, 1994; Landelius, 1997), DT zero-sum games (Al-Tamimi *et al.*, 2007), and DT output feedback optimal control (Lewis & Vamvoudakis, 2010). In their works, the persistence of excitation (PE) condition is needed for parameter convergence and online implementation.

At each decision step, $Q$-learning for a finite MDP either randomly explores the state-action space to update the action value for the unexplored state-action pair, or exploit the action values to modify and improve the current policy. Note that without exploration, only some limited areas of action values are estimated and hence, the final updated policy could not be optimal in the whole state-action space. (Sutton & Barto, 1998). For DT LTI systems, similar results can be found from the works of Al-Tamimi *et al.* (2007); Bradtke & Ydstie (1994); Lewis & Vamvoudakis (2010). In their works, exploration, so-called probing noise, is necessary to prevent the PE condition from being lost. Without PE, the learning agent cannot update the next policy anymore. Moreover, in $Q$-learning with batch least squares (LS) (Al-Tamimi *et al.*, 2007; Lewis & Vamvoudakis, 2010), the poor excitation introduces the considerably large numerical errors at the policy evaluation step since it contains an inverse operation of a matrix where the condition number may become very large due to the poor excitation. These correspond to the $Q$-learning for a finite MDP framework where exploration noise should be suitably injected to improve the performance of the agent.

On the other hand, the early $Q$-learning schemes for CT dynamic systems were proposed independently by BairdIII (1994) and Doya (2000). Although these early methods can be applied to the general uncertain autonomous nonlinear systems, they do not guarantee the stability and convergence to the optimal solution. From the different perspectives, Murray, Cox, Lendaris, & Saek (2002) proposed an RL method for CT LQR problems, which needs the measurements of the state derivatives for online implementation. Inspired by the works of Murray, *et al.* (2002), Vrabie, Pastravanu, Abu-Kalaf, & Lewis (2009) proposed a derivative-free online RL method for CT LQR problems. This class of RL schemes is also called policy iteration (PI), and is proven to be stable and convergent to the optimal solution, provided that the initial policy is stabilizing (Murray, *et al.*, 2002; Vrabie *et al.*, 2009). However, they requires the exact knowledge of the input coupling terms in the system dynamics to update the control policy. This restriction also exists in similar RL methods for CT nonlinear systems (Dong & Farrell, 2009; Lewis & Vrabie, 2009) and the synchronous PI recently developed by Vamvoudakis & Lewis (2010).

Inspired by the work of Lewis & Vrabie (2009), our previous work proposed an RL algorithm to solve a CT LQR problem without knowing the system dynamics $A$ and $B$ (Lee, Park, & Choi, 2009). However, the method only provides an approximate solution to the LQR problem, and the internal signal becomes impulsive as the target approximate solution approaches to the exact one. Recently, Mehta & Meyn (2009) proposed a CT $Q$-learning algorithm with the connection to Pontryagin's principle, but the stability and convergence properties of the method have not yet been proven as well. In summary, to the best authors' knowledge, all the RL methods for CT dynamic systems either requires the perfect knowledge of the input coupling terms, or do not guarantee the stability and convergence.

Motivated by the work of Vrabie *et al.* (2009), this paper presents the integral $Q$-learning scheme which solves a given CT LQR problem without knowing the system dynamics $A$ and $B$. By simultaneously evaluating the current value function and the improved control policy, the proposed $Q$-learning agent solves a CT LQR problem with guaranteed stability and convergence. More specifically,

- **In Section 2**, the LQR problem and its $Q$-function are first addressed and the variants of this $Q$-function are presented via singular input perturbation. The exploration and PE with this input perturbation are also discussed.
- **In Section 3.1–3**, explorized PI is proposed and based on the results, the main integral $Q$-learning is presented. Here, explorized PI can deal with explorations, which is not for the conventional PI, and the integral $Q$-learning essentially requires this exploration to obtain the improved control policy without knowing the matrix $B$.
- **In Section 3.4**, The exploration and PE will be further investigated with the connection to least-squares online implementation of the proposed algorithms.
- **In Section 4**, to verify the effectiveness of the proposed algorithms, simulation results are provided with the comparison to the PI of Vrabie *et al.* (2009).

## 2 Preliminaries

### 2.1 Notations & Mathematical Background

In this paper, we denote $\mathbb{Z}_+$ the set of nonnegative integers and $\mathbb{M}^{m\times n}$ the set of all $m \times n$ constant matrices. Also, for a symmetric matrix $X \in \mathbb{M}^{n\times n}$, $\lambda_M(X)$ and $\lambda_m(X)$ denote the maximum and minimum eigenvalues of $X$, respectively. Throughout the paper, the spectral norm and Euclidean norm, defined as $\|A\| := \lambda_M(A^TA)^{1/2}$ and $\|x\| := (x^Tx)^{1/2}$, respectively, will be used for a matrix $A \in \mathbb{M}^{m\times n}$ and a real vector $x \in \mathbb{R}^n$. Here, $A^T$ denotes the transpose of $A$. For a sequence $\{a_k\}_{k=0}^{\infty}$ with $a_k \in \mathbb{R}^n$, $\Delta a_k$ represent the difference $\Delta a_k = a_{k+1} - a_k$, and for a continuously differentiable functional $f(x,y)$ with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, $\nabla_x f(x,y)$ denotes the gradient of $f(x,y)$ with respect to $x$.

For compact representations, we will use $\mathrm{vec}(X)$ for $X \in \mathbb{M}^{m\times n}$ as a vectorization map from a matrix into an $mn$–dimensional column vector. This $\mathrm{vec}(X)$ stacks the columns of $X$ on the top of one another. Also, we let $\mathrm{vec}^{+}(Y)$ be defined as an operator which maps a symmetric matrix $Y \in \mathbb{M}^{n\times n}$ into a vector with dimension $q_n$ $(:= n(n+1)/2)$ by stacking the columns corresponding to the diagonal and upper triangular parts of $Y$ on the top of another where the off-diagonal terms of $Y$ are doubled. Here, we define $q_n := n(n+1)/2$ for any $n \in \mathbb{N}$. Also, we let $A \otimes B$ be a Kronecker product of $A$ and $B$, and denote the Kronecker product of $A$ itself, *i.e.*, $A \otimes A$ by $\overline{A}$. The key properties for these three operations are

1. $\mathrm{vec}(AXB) = (B^T \otimes A)\,\mathrm{vec}(X)$;
2. for every $x$, $y \in \mathbb{R}^n$, there exists a permutation matrix $U \in \mathbb{M}^{mn\times mn}$ such that $(x \otimes y) = U(y \otimes x)$;
3. for every $Y = Y^T \in \mathbb{M}^{n\times n}$, there is a matrix $\Gamma \in \mathbb{M}^{n^2 \times q_n}$ with rank $(\Gamma) = q_n$ such that $\mathrm{vec}(Y) = \Gamma \mathrm{vec}^{+}(Y)$ (Murray, *et al.*, 2002).

Here, the dimensions of the matrices $A$, $B$, and $X$, and the column vectors $x$ and $y$ are assumed to be all compatible. Note that $x^T Ay = (y \otimes x)^T\,\mathrm{vec}(A) = \mathrm{vec}(A)^T(y \otimes x)$ holds as a special case of Property 1.

### 2.2 Optimality Principle & Q-Function for LQR

In this paper, we consider the infinite horizon LQR problem for the following CT LTI system

$$\dot{x}_t = Ax_t + Bu_t,\ x(0) = x_0 \tag{1}$$

with the value function

$$V_u(x_t,t) = \int_t^{\infty} c(x,u)\,d\tau \tag{2}$$

for a policy $u$, where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the state and input vectors; $A \in \mathbb{M}^{n\times n}$ and $B \in \mathbb{M}^{n\times m}$ are the system and input coupling matrices of the system (1); $c(x,u)$ is the quadratic cost function defined as $c(x,u) := x^T Sx + u^T Ru$ for some $S = C^T C \geq 0$ $(C \in \mathbb{M}^{p\times n})$ and $R > 0$. Here, $u_t$, $u(t)$ and simply $u$ will be used interchangeably for the input of the system (1), and the following will be assumed throughout the paper (Lewis & Syrmos, 1995):

**Assumption 1** *The triple $(A,B,C)$ is at least stabilizable and detectable.*

For a given policy $u = -Kx$, define $A_K$ as the closed-loop matrix $A_K := A - BK$. If $u = -Kx$ is stabilizing for the system (1), then, the value function (2) is finite (Kleinman, 1968), and without loss of generality, $V_u$ can be represented as the time-invariant formula $V_u(x_t) = x_t^T Px_t$. Now, let $V^*(x) = x^T P^* x$ and $u^* = -K^* x$ be the optimal value function and control, that is, $V^*(x) := \min_u V_u(x)$ and $u^* := \arg\min_u V_u(x)$. Then, by Bellman's optimality principle, $V^*(x)$ satisfies the optimality equation (Lewis & Syrmos, 1995; Lewis & Vrabie, 2009):

$$V^*(x_t) = \min_{\substack{u(\tau),\\ \tau\in[t,t+T]}} \left[ \int_t^{t+T} c(x,u)\,d\tau + V^*(x_{t+T}) \right], \tag{3}$$

which is the basis of the CT PI of Vrabie *et al.* (2009) and also the connection between $V^*(x_t)$ and the CT $Q$-function $Q^*(x,u)$. Dividing both sides of (3) by $T$ and limiting $T \to 0^+$ yields the infinitesimal version thereof:

$$\min_u \left( c(x,u) + \dot{V}^*(x) \right) = 0. \tag{4}$$

Mehta & Meyn (2009) mentioned that the CT $Q$-function $Q^*(x,u)$ is the function of two variables within the minimum on the left side of (4) which is closely related to Hamiltonian $\mathscr{H}(x,u,p) = c(x,u) + p^T(Ax+Bu)$. Therefore, $Q^*(x,u)$ is given by $Q^*(x,u) = c(x,u) + \dot{V}^*(x)$, and (4) can be rewritten in terms of the $Q$-function as

$$\min_u Q^*(x,u) = 0, \tag{5}$$

so that $u^*$ can be obtained by minimizing $Q^*(x,u)$. Here, by $\dot{V}^*(x) = (\nabla_x V^*(x))^T(Ax+Bu)$, $Q^*(x,u)$ can be represented as the following quadratic formula:

$$Q^*(x,u) = \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{bmatrix} H_{11}^* & H_{12}^* \\ \star & H_{22}^* \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{6}$$

where $H_{11}^* := A^T P^* + P^* A + S$, $H_{12}^* := P^* B$, and $H_{22}^* := R$. By solving $\nabla_u Q^*(x,u) = 0$, one obtains the optimal solution $u^*$ minimizing (6) as

$$u^* = -K^* x = -(H_{22}^*)^{-1}(H_{12}^*)^T x, \tag{7}$$

which is a key equation of the proposed $Q$-learning. From (7) and the definitions of $H_{12}^*$ and $H_{22}^*$, one has $K^* = R^{-1}B^T P^*$. Furthermore, substituting the minimum solution (7) into (5) and rearranging the equation according to the definitions of $H_{11}^*$, $H_{12}^*$, and $H_{22}^*$, one obtains the well-known algebraic Riccati equation (ARE):

$$A^T P^* + P^* A - P^* BR^{-1}B^T P^* + S = 0, \tag{8}$$

which has the unique solution $P^*$ by Assumption 1 (Lewis & Syrmos, 1995). These equations (6)–(8) actually provide the relations between $Q$-function $Q^*(x,u)$ and the optimal solution $(V^*,u^*)$. Note that there is no direct connection between $H_{11}^*$ of $Q^*(x,u)$ and $u^*$ (see (7)), and only (8) provides the dependent relation $H_{11}^* = H_{12}^*(H_{22}^*)^{-1}(H_{12}^*)^T = (K^*)^T R(K^*)$. That is, though it contains the information about $(A,P^*)$, $H_{11}^*$ is an unnecessary redundant term when the learning process of $u^*$ is considered. This motivates the introduction of $\varepsilon$-integral $Q$-function in the next section which contains the information about $(P^*, H_{12}^*, H_{22}^*)$, instead of $(H_{11}^*, H_{12}^*, H_{22}^*)$.

### 2.3 $\varepsilon$-Integral $Q$-functions

To introduce $\varepsilon$-integral $Q$-function, which is closely related with the $Q$-function $Q^*(x,u)$, consider the additional input dynamics

$$\varepsilon\dot{u} = v,\ \ u(0) = u_0 \tag{9}$$

perturbed by $\varepsilon$, where $\varepsilon > 0$ is a small constant, and $v_t \in \mathbb{R}^m$ is the virtual policy which drives $u$. Then, the input dynamics (9) has the following property:

**Lemma 1** *Suppose $v = -R^{-1}K_1^T x - R^{-1}K_2 u + w_t$ is applied to* (9) *with some $K_1$, $K_2$, and $w_t$. Then, the additional dynamics* (9) *can be rewritten as $u = \mathscr{T}^{\varepsilon}_{K_2}(s)\big[-K_1^T x + Rw_t\big]$, with the Laplace variable $s$ where $\mathscr{T}^{\varepsilon}_{K_2}(s)$ is the low pass filter defined by $\mathscr{T}^{\varepsilon}_{K_2}(s) := (\varepsilon sR + K_2)^{-1}$.*

This lemma will be widely used in the paper, and the proof is trivial (just take the Laplace transform of (9) with $v$ given in Lemma 1). In the sequel, consider the LQR problem for the dynamics (1) and (9) with the compact representation:

$$\dot{z} = Fz + G^{\varepsilon}v,\ \ z(0) = z_0 \tag{10}$$

and the quadratic value function, denoted by $Q_I^v(x_t,u_t,\varepsilon)$, for the system (10)

$$Q_I^v(x_t,u_t,\varepsilon) := \int_t^{\infty} c(x,u) + v^T Rv\, d\tau, \tag{11}$$

where $F := \left[\begin{smallmatrix} A & B \\ 0 & 0\end{smallmatrix}\right]$, $G^{\varepsilon} := [\,0 \quad I_m/\varepsilon\,]^T$ $z := [\,x^T \quad u^T\,]^T$, and $z_0 := [\,x_0^T \quad u_0^T\,]^T$. Here, $c(x,u)$ can be rewritten by the simple quadratic form $c(x,u) = z^T\Sigma z$ with $\Sigma := \operatorname{diag}\{S,R\}$. This LQR problem is theoretically involved with the singular perturbation theory and cheap optimal control (Kokotovic, Khalil, & J. O'Reilly, 1986) and since Assumption 1 trivially implies the stabilizability and detectabillity of $(F, G^{\varepsilon}, \Sigma^{1/2})$, by the arguments in Section 2.2 (or by the singular perturbation theory), there exists the unique solution $Q_I^*(x,u,\varepsilon)$ and $v^*$ which are given as follows:

$$Q_I^*(x,u,\varepsilon) = [\,x^T \quad u^T\,]\begin{bmatrix} H_{11}^{\varepsilon} & \varepsilon H_{12}^{\varepsilon} \\ \star & \varepsilon H_{22}^{\varepsilon}\end{bmatrix}\begin{bmatrix} x \\ u\end{bmatrix}, \tag{12}$$

$$v^*(t) = -R^{-1}(H_{12}^{\varepsilon})^T x(t) - R^{-1}H_{22}^{\varepsilon}u(t). \tag{13}$$

where $H^{\varepsilon} := \left[\begin{smallmatrix} H_{11}^{\varepsilon} & \varepsilon H_{12}^{\varepsilon} \\ \star & \varepsilon H_{22}^{\varepsilon}\end{smallmatrix}\right] \geq 0$ is the solution of the ARE:

$$F^T H^{\varepsilon} + H^{\varepsilon}F + \Sigma = \begin{bmatrix} H_{12}^{\varepsilon}R^{-1}(H_{12}^{\varepsilon})^T & H_{12}^{\varepsilon}R^{-1}H_{22}^{\varepsilon} \\ \star & H_{22}^{\varepsilon}R^{-1}H_{22}^{\varepsilon}\end{bmatrix}, \tag{14}$$

which can be block-wisely decomposed as

$$A^T H_{11}^{\varepsilon} + H_{11}^{\varepsilon}A - H_{12}^{\varepsilon}R^{-1}(H_{12}^{\varepsilon})^T + S = 0, \tag{15}$$

$$\varepsilon A^T H_{12}^{\varepsilon} + H_{11}^{\varepsilon}B - H_{12}^{\varepsilon}R^{-1}H_{22}^{\varepsilon} = 0, \tag{16}$$

$$\varepsilon(B^T H_{12}^{\varepsilon} + (H_{12}^{\varepsilon})^T B) - H_{22}^{\varepsilon}R^{-1}H_{22}^{\varepsilon} + R = 0. \tag{17}$$

Here, we consider $Q_I^*(x,u,\varepsilon)$ as the $\varepsilon$-integral $Q$-function for the LQR problem (1)–(2). Now, define the $\varepsilon$-approximate $Q$-function $Q^*(x,u,\varepsilon)$ as $Q^*(x,u,\varepsilon) := (v^*)^T Rv^*$ with abuse of notation. Then, substituting (13) and (15) into $Q^*(x,u,\varepsilon)$ yields the following formula

$$Q^*(x,u,\varepsilon) = [\,x^T \quad u^T\,]\begin{bmatrix} A^T H_{11}^{\varepsilon} + H_{11}^{\varepsilon}A + S & H_{12}^{\varepsilon}R^{-1}H_{22}^{\varepsilon} \\ \star & H_{22}^{\varepsilon}R^{-1}H_{22}^{\varepsilon}\end{bmatrix}\begin{bmatrix} x \\ u\end{bmatrix} \tag{18}$$

which corresponds to (6) of $Q^*(x,u)$. Furthermore, define the $\varepsilon$-approximate optimal policy $(u^{\varepsilon})^*$ as the policy $u$ generated by (9) with $v = v^*$. Then, by Lemma 1, one has

$$(u^{\varepsilon})^* = -\mathscr{T}^{\varepsilon}_{H_{22}^{\varepsilon}}(s)\big[(H_{12}^{\varepsilon})^T x\big], \tag{19}$$

the approximate version of (7). In fact, $Q_I^*(x,u,\varepsilon)$ is related with $V_u(x)$ and $Q^*(x,u,\varepsilon)$ under $u = (u^{\varepsilon})^*$:

**Proposition 1** *Consider the $\varepsilon$-integral $Q$-function* (11) *with the system* (10)*. Then, $Q_I^*(x,u,\varepsilon) \geq 0$ given by* (12) *satisfies $H_{11}^{\varepsilon} \geq 0$, $H_{22}^{\varepsilon} \geq 0$, and under $u = (u^{\varepsilon})^*$ with the initial condition $u(t) = u_t$,*

$$Q_I^*(x_t,u_t,\varepsilon) = \int_t^{\infty} Q^*(x_\tau,u_\tau,\varepsilon)\, d\tau + V_u(x_t). \tag{20}$$

**Proof.** $Q_I(x,0,\varepsilon) = x^T H_{11}^{\varepsilon}x \geq 0$ and $Q_I(0,u,\varepsilon) = u^T H_{22}^{\varepsilon}u \geq 0$ proves $H_{11}^{\varepsilon} \geq 0$ and $H_{22}^{\varepsilon} \geq 0$, respectively. (20) can be obtained by substituting $Q^*(z,\varepsilon) = (v^*)^T Rv^*$ into (11). □

Now, consider the limit case $\varepsilon \to 0$ and denote $H_{jk}$ ($j,k = 1,2,\ j \leq k$) by $H_{jk} := \lim_{\varepsilon\to 0^+} H_{jk}^{\varepsilon}$. Then, we state the following lemma and proposition concerning the convergence of $Q^*(x,u,\varepsilon)$, $Q_I^*(x,u,\varepsilon)$, and $(u^{\varepsilon})^*$.

**Lemma 2** *Consider the $\varepsilon$-integral function $Q_I^*(x,u,\varepsilon)$ in the limit case $\varepsilon \to 0^+$. Then, it satisfies $H_{11} = P^*$, $H_{12} = H_{12}^*$, and $H_{22} = H_{22}^*$.*

**Proof.** This can be easily proven by taking the limits $\varepsilon \to 0^+$ of all those equations (15)–(17) and rearranging the results. For a complete proof, see Lemma 1 in Lee *et al.* (2009). □

**Proposition 2** *Consider the $Q$-functions $Q^*(x,u)$, $Q^*(x,u,\varepsilon)$, and $Q_I^*(x,u,\varepsilon)$ defined above with the system* (10)*. Then, for all $(x,u) \in \mathbb{R}^{n+m}$ we have in the limit $\varepsilon \to 0^+$ the followings:*

- $Q^*(x,u,\varepsilon) \to Q^*(x,u)$,
- $Q_I^*(x,u,\varepsilon) \to V^*(x)$, $\ (u^{\varepsilon})^* \to u^*$.

**Proof.** The proof can be done by taking the limits $\varepsilon \to 0^+$ of (19)–(20) and applying Lemma 2 to the resulting equations. Here, note that $\lim_{\varepsilon\to 0^+} \mathscr{T}^{\varepsilon}_{H^{\varepsilon}_{22}}(s) = H_{22}^{-1}$. □

**Remark 1** $Q_I^*(x,u,\varepsilon)$ is actually derived based on the spirit of $Q$-functions presented by Bradtke & Ydstie (1994); Lewis & Vrabie (2009); Sutton & Barto (1998). Minimized with respect to the action or control $u$, the $Q$-functions in the references actually become the optimal value function. Similarly, $Q_I^*(x,u,\varepsilon)$ becomes $V^*(x)$ when optimized with respect to $\varepsilon > 0$ in the input dynamics (9). Note that as $\varepsilon \to 0^+$, the integral term in (20) vanishes and $Q_I^*(x,u,\varepsilon) \to V^*(x)$, the minimizing solution in terms of $\varepsilon > 0$.

### 2.4 Explorations

For the discussion of explorations, assume that $w_t$ is any given non-zero measurable signal which is exactly known *a priori* and bounded by $w_M > 0$, *i.e.*, $\sup_{t\geq 0}\|w_t\| \leq w_M$. Throughout the paper, this $w_t$ will be called an exploration signal (or simply exploration) which plays a key role in both consistently exciting the signal $x_t$ and relaxing the assumption of perfectly known $B$, as will be presented in Section 3. From now on, suppose both a virtual policy $v_t$ and an exploration $w_t$ are applied at the same time to the input dynamics. That is, instead of (9), consider the following input dynamics, explored by $w_t$:

$$\varepsilon \dot{u} = v + w, \;\; u(0) = u_0 \tag{21}$$

Then, similarly to (10), the dynamics (1) and (21) can be compactly rewritten as

$$\dot{z} = Fz + G^{\varepsilon}[v+w], \;\; z(0) = z_0. \tag{22}$$

Note that (21) can be represented as $u = \mathscr{T}^{\varepsilon}_{K_2}(s)[-K_1^T x + Rw]$ by Lemma 1 when $v = -R^{-1}K_1^T x - R^{-1}K_2 u$ is applied with some $K_1$ and $K_2$. Applying this fact into (22), one obtains

$$\dot{x} = Ax + B\cdot\{\mathscr{T}^{\varepsilon}_{K_2}(s)[-K_1^T x + Rw]\}. \tag{23}$$

If $v$ is a stabilizing policy for the system (10), (23) can be further simplified by letting $K_2 = R$ and limiting $\varepsilon \to 0^+$ (Kokotovic *et al.*, 1986) as shown below:

$$\dot{x} = Ax + B[u+w], \;\; x(0) = x_0, \tag{24}$$

where $u$ is given by $u = -Kx$ with the gain matrix $K = R^{-1}K_1^T$. For the investigation of $w_t$, the following assumption is needed:

**Assumption 2** *The exploration $w$ is piecewise constant where the transitions are allowed only at the discrete time instants $(t,\, t+T,\, t+2T,\, t+3T,\, \cdots)$.*

This assumption will be used only in Section 3.4 where some conditions on $w_t$ of (24) are presented for the online implementation of the proposed algorithms. Applying $u = -Kx$ and $w_t$ satisfying Assumption 2 to (24) and defining $x_k := x_{\tau+kT}$ and $w_k := w_{\tau+kT}$ for some $\tau \geq 0$ and $T > 0$, one can reformulate (24) as $x_{k+1} = A_d x_k + B_d w_k$ where $A_d := e^{A_K T}$, $B_d := E_d B$, and $E_d := \int_0^T e^{A_K t}\, dt$. Furthermore, expanding $\bar{x}_{k+1} := x_{k+1} \otimes x_{k+1}$ by using the Kronecker product properties shown in Section 2.1, one has the following DT equivalent dynamics in terms of $\bar{x}_k$:

$$\bar{x}_{k+1} = \bar{A}_d \bar{x}_k + \begin{bmatrix} \Xi & \bar{B}_d \end{bmatrix} \varpi_k \tag{25}$$

where $\Xi := \begin{bmatrix} I & U \end{bmatrix}(A \otimes B_d)$ and $\varpi_k := \begin{bmatrix} (x_k \otimes w_k)^T & \bar{w}_k^T \end{bmatrix}^T \in \mathbb{R}^{mn}$; $U$ is the permutation matrix already shown in Section 2.1. Here, (25) plays a central role in the analysis of exploration in terms of the notion of PE precisely defined below:

**Definition 1 (Willems *et al.*, 2005)** *A bounded DT signal $s_k \in \mathbb{R}^r$ ($k \in \mathbb{Z}_+$) is persistently exciting of order $L \in \mathbb{N}$ if there are no $a_1$, $a_2$, ..., $a_L \in \mathbb{R}^r$, not all zero, such that $\sum_{l=1}^{L} a_l^T s_{k+l-1} = 0$ for all $k \in \mathbb{Z}_+$.*

**Proposition 3** *Assume a bounded DT signal $s_k \in \mathbb{R}^r$ ($k \in \mathbb{Z}_+$) is persistently exciting of order $L \in \mathbb{N}$. Then, there exist $\beta_1,\ \beta_2 > 0$ such that for all $k \in \mathbb{Z}_+$,*

$$\beta_1 I \leq \sum_{l=0}^{L-1} s_{k+l}s_{k+l}^T \leq \beta_2 I. \tag{26}$$

**Proof.** Since $s_k \in \mathbb{R}^r$ is bounded, so is $\sum_{l=0}^{L-1} s_{k+l}s_{k+l}^T$, the existence of $\beta_2 > 0$. Considering the quadratic formula $x^T(\sum_{l=0}^{L-1} s_{k+l}s_{k+l}^T)x = \sum_{l=0}^{L-1}(x^T s_{k+l})^2$ for any nonzero vector $x$, by PE of $s_k$ and $2xy \leq x^2 + y^2$, one has

$$0 \neq \left(\sum_{l=0}^{L-1} x^T s_{k+l}\right)^2 \leq L \sum_{l=0}^{L-1} (x^T s_{k+l})^2. \tag{27}$$

Therefore, $\sum_{l=0}^{L} s_{k+l}s_{k+l}^T$ is positive definite and the existence of $\beta_1 > 0$ is proven. □

## 3 Main Results

In this section, we first present the explorized policy iteration algorithm, and then, based on the results, develop the novel integral $Q$-learning scheme. The practical implementations of the algorithms based on least squares are also discussed in relation to the exploration $w$.

### 3.1 Explorized Policy Iteration

The proposed explorized PI is aimed at finding the optimal solutions $V^*(x)$ and $u^*$ online for the system (23) with the known exploration $w_t$ and uncertain/unknown system matrix $A$. The distinguishing feature of this algorithm from the PI proposed by Vrabie *et al.* (2009) lies in the exploration $w_t$. By virtue of $w_t$, the agent can autonomously explore the state-space to efficiently update the policy and its

**Algorithm 1: Explorized Policy Iteraion**

```
Let $P_0 = 0$ and $u_1 = -K_1 x$ be any stabilizing policy.
$i \leftarrow 0$
do {
    $i \leftarrow i + 1$
    Let $w_t$ be any nonzero exploration.
    Apply the input $u = u_i$ with exploration $w_t$ to (24).
    Policy Evaluation: Find $V_i(x) = x^T P_i x$ satisfying
        $$V_i(x_t) + 2\Phi_i(t,T) = \int_t^{t+T} c(x,u_i)\; d\tau + V_i(x_{t+T}) \qquad (28)$$
        $$\text{where } \Phi_i(t,T) = \int_t^{t+T} x^T P_i B w \, d\tau. \qquad (29)$$
    Policy Improvement:
        $$K_{i+1} = R^{-1} B^T P_i, \;\; u_{i+1}(t) = -K_{i+1} x_t \qquad (30)$$
} until $\|P_i - P_{i-1}\| < \delta$.
```

value function. To deal with the exploration, an additional term, denoted by $\Phi_i(t,T)$ , should be incorporated into the PI of Vrabie *et al.* (2009) as shown in Algorithm 1.

In Algorithm 1, $V_i(x_t) = x_t^T P_i x_t$ is the value function for the policy $u_i$. Note that if $w(t) \equiv 0$ for all the iterations, Algorithm 1 becomes the PI of Vrabie *et al.* (2009). Now, we will prove the stability and convergence of Algorithm 1. For notational convenience, define $A_i$, $M_i$, and $C_i$ as $A_i := A - BK_i$, $M_i := S + K_i^T R K_i$, and $C_i := \|RK_i\| / \lambda_m(M_i)$. Then, (24) with $u_i = -K_i x$ can be represented as $\dot{x} = A_i x + Bw$.

**Lemma 3** *If $A_i$ is stable, then, one step recursion* (28)–(30) *of Algorithm 1 is equivalent to solving the following Lyapunov equation for $P_i > 0$:*

$$(A_i)^T P_i + P_i A_i = -M_i. \tag{31}$$

**Proof.** Assume that $A_i$ is stable. Then, for $M_i > 0$, there is $P_i > 0$ such that (31) holds. Considering the Lyapunov function $V_i(x_t) = x_t^T P_i x_t$ and its derivative $\dot{V}_i(x_t) = x_t^T \left[A_i^T P_i + P_i A_i\right] x_t + 2u_{i+1}^T R w$ along the system $\dot{x} = A_i x + Bw$, one has

$$\begin{aligned} \int_t^{t+T} x^T S x + u_i^T R u_i \; d\tau \;&=\; \int_t^{t+T} x_\tau^T M_i x_\tau \;\; d\tau \\ &= -\int_t^{t+T} \dot{V}_i(x_\tau) - 2x^T P_i B w \, d\tau \\ &= V_i(x_t) - V_i(x_{t+T}) + 2\Phi_i(t,T), \end{aligned}$$

which completes the proof. □

**Theorem 1** *Suppose $S > 0$ and $(V_i, u_i)$ is updated by Algorithm 1. If the initial policy $u_1$ is stabilizing, then, $A_i$ is stable and the closed-loop system $\dot{x} = A_i x + Bw$ is uniformly ultimately bounded (UUB) $\forall i \in \mathbb{N}$, with each ultimate bound $\|x\| \le 2w_M C_i$ for $i \ne 1$. Furthermore, as $i \to \infty$, $V_i$ and $u_i$ converge to the optimal solution $V^*$ and $u^*$, respectively.*

**Proof.** This is proven by mathematical induction. First, assume $A_i$ is stable and consider the Lyapunov function candidate $V_i(x_t) = x_t^T P_i x_t$ for the $i$-th system $\dot{x} = A_i x + Bw$. Then, taking the time derivative of $V_i(x_t)$ and following the similar procedure of Vrabie *et al.* (2009), one obtains

$$\dot{V}_i(x) \le -x^T M_i x - 2x^T R K_i w \tag{35}$$

where Lemma 3 provides the substitution of (31) in the procedure of the derivation. Since $S > 0$ is assumed, $M_i$ is obviously positive definite, so one has from (35)

$$\dot{V}_i(x) \le -\lambda_m(M_i)\, \|x\|^2 + 2w_M \, \|RK_i\| \, \|x\| \, . \tag{36}$$

Therefore, $\dot{V}_i(x_t) < 0$ holds for $x$ satisfying $\|x\| > 2w_M \cdot C_i$. By Lyapunov's theorem (Khalil, 2002) and induction, this proves $A_i$ is stable and the system $\dot{x} = A_i x + Bw$ is UUB with the ultimate bound $2w_M C_i$, for all $i \in \mathbb{Z}_+$. Now, by the equivalence of (31) and Kleinman (1968)'s Newton method, the convergence $V_i \to V^*$ and $u_i \to u^*$ can be proven under an initial stabilizing $u_1$ and $S > 0$ (Vrabie *et al.*, 2009). □

**Algorithm 2: $\varepsilon$-Approximate Integral $Q$-learning**

```
Let $H_{11}^{[0]} = 0$ and $u = \mathscr{T}_0^{\varepsilon}(s)\left[-(H_{12}^{[0]})^T x + Rw\right]$ be any stabilizing
  policy where $\mathscr{T}_0^{\varepsilon}(s) = (\varepsilon s R + H_{22}^{[0]})^{-1}$.
$i \leftarrow 0$
do {
    $i \leftarrow i + 1$
    Let $w_t$ be any nonzero exploration.
    Apply the input $u = \mathscr{T}_{i-1}^{\varepsilon}(s)\left[-(H_{12}^{[i-1]})^T x + Rw\right]$ to the system
      $\dot{x} = Ax + Bu$.
    Policy Evaluation: Find $Q_I^{[i]}(x_t, u_t, \varepsilon) = z_t^T H_i^{\varepsilon} z_t$ satisfying
        $$Q_I^{[i]}(x_t, u_t, \varepsilon) + 2\Phi_i(t,T,\varepsilon) \qquad (32)$$
        $$\quad = \int_t^{t+T} \left[c(x,u) + Q_i(x,u,\varepsilon)\right] d\tau + Q_I^{[i]}(x_{t+T}, u_{t+T}, \varepsilon),$$
        $$\text{where } \Phi_i(t,T,\varepsilon) = \int_t^{t+T} \left[x^T H_{12}^{[i]} w + u^T H_{22}^{[i]} w\right] d\tau.$$
    Policy Improvement:
        $$\mathscr{T}_i^{\varepsilon}(s) = (\varepsilon s R + H_{22}^{[i]})^{-1} \qquad (33)$$
        $$u = \mathscr{T}_i^{\varepsilon}(s)\left[-(H_{12}^{[i]})^T x + Rw\right] \qquad (34)$$
} until $\left\|H_i^{\varepsilon} - H_{i-1}^{\varepsilon}\right\| < \delta$.
```

### 3.2 $\varepsilon$-Approximate Integral Q-Learning

By applying Algorithm 1 to the system (22) with the $\varepsilon$-integral $Q$-function (11), we derive in this subsection $\varepsilon$-approximate integral $Q$-learning which is shown in Algorithm 2 as an approximate version of the proposed integral $Q$-learning. When Algorithm 1 is applied to (22) and (11), the virtual policy $v = v_i$ at $i$-th iteration is given by

$$v_i = -R^{-1}(H_{12}^{[i-1]})^T x - R^{-1} H_{22}^{[i-1]} u. \tag{37}$$

Then, one obtains (33)–(34) by substituting (37) into (22) and applying Lemma 1. In Algorithm 2, $Q_I^{[i]}(x_t, u_t, \varepsilon) =$

$z_t^T H_i^\varepsilon z_t$ is the $\varepsilon$-integral $Q$-function for the policy (33)–(34) at $i$-th iteration, with $H_i^\varepsilon$ partitioned as

$$H_i^\varepsilon = \begin{bmatrix} H_{11}^{[i]} & \varepsilon H_{12}^{[i]} \\ \star & \varepsilon H_{22}^{[i]} \end{bmatrix} \tag{38}$$

which is actually the performance index (11) for the system (10) when $v = v_i$. In (32) of line 7, $Q_i(x,u,\varepsilon)$ is defined by $Q_i(x,u,\varepsilon) = v_i^T R v_i$ and can be considered an estimate of the $\varepsilon$-approximate $Q$-function at $i$-th iteration (see the definition of $Q^*(x,u,\varepsilon)$). In addition, by substituting (37) into $Q_i(x,u,\varepsilon) = v_i^T R v_i$, it can be represented as the quadratic form $Q_i(x,u,\varepsilon) = z^T \Pi_i z$ where

$$\Pi_i := \begin{bmatrix} H_{12}^{[i-1]} R^{-1} (H_{12}^{[i-1]})^T & H_{12}^{[i-1]} R^{-1} H_{22}^{[i-1]} \\ \star & H_{22}^{[i-1]} R^{-1} H_{22}^{[i-1]} \end{bmatrix}.$$

By Theorem 1 and Lemma 3, one can see the following three key properties of Algorithm 2 which are essentially employed to derive exact integral $Q$-learning in the next subsection:

1. Algorithm 2 guarantees the stability and convergence to $(u^\varepsilon)^*$ and $Q^*(x,u,\varepsilon)$ by Theorem 1. In this case, the corresponding ultimate bound in Theorem 1 becomes $\|x\| \leq 2w_M D_i$ where $D_i$ is defined as

$$D_i := \big\| \big[ (H_{12}^{[i-1]})^T \;\; (H_{22}^{[i-1]})^T \big] \big\| / \lambda_m(\Sigma + \Pi_i).$$

2. According to the recursion (31) in Lemma 3, when the initial policy is stabilizing, $H_i^\varepsilon$ obtained by Algorithm 2 satisfies the recursion

$$F_i^T H_i^\varepsilon + H_i^\varepsilon F_i = -\Pi_i - \Sigma, \tag{39}$$

where $F_i := F - G^\varepsilon R^{-1} G^\varepsilon H_{i-1}^\varepsilon$. Furthermore, decomposing (39) block-wisely, one has the following set of recursive matrix equations:

$$\begin{aligned} & A^T H_{11}^{[i]} + H_{11}^{[i]} A - [H_{12}^{[i-1]} R^{-1} (H_{12}^{[i]})^T + H_{12}^{[i]} R^{-1} (H_{12}^{[i-1]})^T] \\ & = -H_{12}^{[i-1]} R^{-1} (H_{12}^{[i-1]})^T - S, \qquad (40) \\ & \varepsilon A^T H_{12}^{[i]} + H_{11}^{[i]} B - [H_{12}^{[i-1]} R^{-1} H_{22}^{[i]} + H_{12}^{[i]} R^{-1} H_{22}^{[i-1]}] \\ & = -H_{12}^{[i-1]} R^{-1} H_{22}^{[i-1]}, \qquad (41) \\ & \varepsilon (B^T H_{12}^{[i]} + H_{12}^{[i]} B) - [H_{22}^{[i-1]} R^{-1} H_{22}^{[i]} + H_{22}^{[i]} R^{-1} H_{22}^{[i-1]}] \\ & = -H_{22}^{[i-1]} R^{-1} H_{22}^{[i-1]} - R. \qquad (42) \end{aligned}$$

3. Algorithm 2 guarantees the monotonicity $0 \leq H^\varepsilon \leq H_{i+1}^\varepsilon \leq H_i^\varepsilon$, *i.e.*,

$$0 \leq Q_I^*(x_t,u_t,\varepsilon) \leq Q_I^{[i+1]}(x_t,u_t,\varepsilon) \leq Q_I^{[i]}(x_t,u_t,\varepsilon). \tag{43}$$

This can be obtained by the monotonicity of Kleinman (1968)'s method and its equivalence to Algorithm 1 (Vrabie *et al.*, 2009). By (43), in the sense of minimizing $Q_I(x,u,\varepsilon)$, the policy $u$ becomes better as the iteration runs. Moreover, from (43), one has

$$0 \leq H_{11}^\varepsilon \leq H_{11}^{[i+1]} \leq H_{11}^{[i]} \tag{46}$$
$$0 \leq H_{22}^\varepsilon \leq H_{22}^{[i+1]} \leq H_{22}^{[i]}. \tag{47}$$

for all $i \in \mathbb{Z}_+$. The former is obtained by letting $u_t = 0$ in (43), and the latter by letting $x_t = 0$ in (43). From this, we have the following essential lemma:

**Lemma 4** *If $H_{22}^{[i]}$ is evaluated by Algorithm 2 with the initial matrix $H_{22}^{[0]} = H_{22}^\varepsilon$, then, $H_{22}^{[i]} = H_{22}^\varepsilon$ holds $\forall i \in \mathbb{N}$.*

**Proof.** If $H_{22}^{[0]} = H_{22}^\varepsilon$, then, $0 \leq H_{22}^\varepsilon \leq \cdots \leq H_{22}^{[1]} \leq H_{22}^{[0]} = H_{22}^\varepsilon$ holds by (47), so $H_{22}^{[1]} = H_{22}^{[2]} = \cdots = H_{22}^\varepsilon$. □

---**Algorithm 3: Integral *Q*-learning for Adaptive LQR**---------

```
Let $u_1 = -K_1 x$ be any stabilizing policy for (1).
$i \leftarrow 0$ and set $H_{11}^{[0]} = 0$ and $H_{12}^{[0]} = RK_1^T$.
do {
$i \leftarrow i+1$
Let $w_t$ be any nonzero exploration.
Apply the input $u = u_i$ with exploration $w_t$ to (24).
Policy Evaluation:  Find $H_{11}^{[i]}$ and $H_{12}^{[i]}$ satisfying
    $x_t^T H_{11}^{[i]} x_t + 2\Phi_i(t,T) = \int_t^{t+T} c(x,u_i)\, d\tau + x_{t+T}^T H_{11}^{[i]} x_{t+T},$   (44)
    where $\Phi_i(t,T) = \int_t^{t+T} x^T H_{12}^{[i]} w\, d\tau.$
Policy Improvement:
    $K_{i+1} = R^{-1}(H_{12}^{[i]})^T,\ u_{i+1} = -K_{i+1} x,$   (45)
} until $\|H_{11}^{[i]} - H_{11}^{[i-1]}\| + \|H_{12}^{[i]} - H_{12}^{[i-1]}\| < \delta.$
```

**Remark 2** While Algorithm 1 does not require the knowledge of $A$, it needs the known matrix $B$. On the contrary, by virtue of the additional input dynamics (21), Algorithm 2 does not need the knowledge of both matrices $A$ and $B$.

### *3.3 Integral Q-Learning: True Adaptive Optimal Control*

Based on Algorithm 2 and its key properties, we now derive the novel integral $Q$-learning algorithm for CT LTI system (24) with the exploration $w_t$. The key distinction of the resultant algorithm from Algorithm 2 is that it does not require the additional input dynamics (21) and guarantees the convergence to the true LQR solutions $V^*(x)$ and $u^*$. Note that by the stability of Algorithm 2, $\mathscr{T}_i^\varepsilon(s)$ is always stable, so that one can limit $\varepsilon \to 0^+$ while maintaining the stability of (34) (Kokotovic *et al.*, 1986).

**Lemma 5** *Consider Algorithm 2 in the limit $\varepsilon \to 0^+$. Then, as $\varepsilon \to 0^+$, the followings hold $\forall i \in \mathbb{Z}_+$:*

*1)* $\mathscr{T}_i^\varepsilon(s) \to (H_{22}^{[i]})^{-1}, \;\; Q_I^{[i]}(x,u,\varepsilon) \to x^T H_{11}^{[i]} x$

*2)* $\mathscr{T}_i^\varepsilon(s) \big[ -(H_{12}^{[i]})^T x + Rw \big] \to u_i + (H_{22}^{[i]})^{-1} Rw$

*where* $u_i = -(H_{22}^{[i]})^{-1}(H_{12}^{[i]})^T x$.

**Proof.** The first part is obvious if one considers the limit $\varepsilon \to 0^+$ of (33) and (38), and the second part is the trivial application of the first part to (34). □

Now, let $Q_i(x,u)$ be the approximate $Q$-function in the limit, that is, $Q_i(x,u) := \lim_{\varepsilon\to 0^+} Q_i(x,u,\varepsilon)$. Then, the application of Lemma 5 to (32) in the limit $\varepsilon \to 0^+$ yields

$$\bullet\ x_t^T H_{11}^{[i]} x_t + 2\int_t^{t+T} \left[x^T H_{12}^{[i]} w + u^T H_{22}^{[i]} w\right] d\tau = \int_t^{t+T} \left[c(x,u) + Q_i(x,u)\right] d\tau + x_{t+T}^T H_{11}^{[i]} x_{t+T}, \quad (48)$$

$$\bullet\ u = u_i + (H_{22}^{[i]})^{-1} R w \quad (49)$$

which can be further simplified by noting that $\lim_{\varepsilon\to 0^+} H_{22}^{\varepsilon} = H_{22} = H_{22}^* = R$ holds by Lemma 2 and thus, that if one has $H_{22}^{[0]} = R$, then, $H_{22}^{[i]} = R$ holds for all $i \in \mathbb{Z}_+$ by Lemma 4. Therefore, substituting $H_{22}^{[i]} = R$ into (48)–(49) and rearranging the equations yields the true integral $Q$-learning shown in Algorithm 3 for the system (24) with an exploration $w_t$.

Now, for notational convenience in the analysis of Algorithm 3, redefine $P_i$, $K_i$, $A_i$, $M_i$, $C_i$, and $D_i$ with abuse of notations as

$$P_i = H_{11}^{[i]}, \qquad K_i := R^{-1}(H_{12}^{[i-1]})^T, \quad (50)$$
$$A_i := A - BK_i, \qquad M_i := S + K_i^T R K_i, \quad (51)$$
$$C_i := \frac{\|RK_i\|}{\lambda_m(M_i)}, \qquad D_i := \frac{\|R\begin{bmatrix}K_i & I_m\end{bmatrix}\|}{\lambda_m(\Sigma+\Pi_i)}. \quad (52)$$

Here, $M_i > 0$ holds if $S > 0$, which further guarantees $\Sigma + \Pi_i > 0$ by Schur complement of (53). Note that for Algorithm 3, the matrices $\Pi_i$ and $\Sigma + \Pi_i$ are represented as

$$\Pi_i = \begin{bmatrix} K_i^T R K_i & K_i^T R \\ \star & R \end{bmatrix}, \quad \Sigma + \Pi_i = \begin{bmatrix} M_i & K_i^T R \\ \star & 2R \end{bmatrix}. \quad (53)$$

Here, considering $H_{11}^* = (K^*)^T R K^*$ and $H_{12}^* = (K^*)^T R$, one can see from (6) and (53) that $\lim_{K_i\to K^*} \Pi_i = H^*$, *i.e.*, $Q_i(x,u) \to Q^*(x,u)$ as $K_i \to K^*$. This implies under $K_i \to K^*$, $\Pi_i$ is an approximate of $H^*$ at $i$-th iteration. By the application of Theorem 1 to Algorithm 3 with $S > 0$, the stability and convergence $K_i \to K^*$ and $P_i \to P^*$ are all guaranteed with the corresponding ultimate bound $\|x\| \le 2w_M D_i$. Therefore, $Q_i(x,u)$ obtained by Algorithm 3 surely converges to $Q^*(x,u)$. Furthermore, the following lemma allows to investigate the further properties regarding Algorithm 3.

**Lemma 6** *Under the notations* (50)–(52), *assume $A_i$ is stable. Then, one-step recursion* (44)–(45) *of Algorithm 3 is equivalent to solving the following matrix iterative formula:*

$$H_{11}^{[i]} B = H_{12}^{[i]}, \quad (54)$$
$$A_i^T H_{11}^{[i]} + H_{11}^{[i]} A_i = -M_i, \quad (55)$$

**Proof.** Note that Algorithm 2 is equivalent to solving the iterative formula (40)–(42), and that Algorithm 3 is the limiting case $\varepsilon \to 0$ of Algorithm 2 with the substitution of $H_{22}^{[i]} = R$ for all $i \in \mathbb{Z}_+$. Assuming $A_i$ is Hurwitz, taking the limit $\varepsilon \to 0^+$ of (41) and (42), and substituting $H_{22}^{[i]} = H_{22}^{[i-1]} = R$ into the results yield (54) and $R = R$, respectively. Therefore, substituting (54) into (40) and considering the definitions of $K_i$ and $A_i$, one has (55), which completes the proof. □

With the notations (50)–(52) and Lemma 6, the same procedure of the proof of Theorem 1 also proves the stability and convergence with each $i$-th ultimate bound $\|x\| \le 2w_M C_i$. Furthermore, since $P_i > 0$ is guaranteed by $S > 0$, $B$ can be obtained by (54) as $B = (H_{11}^{[i]})^{-1} H_{12}^{[i]}$. The following theorem states all the results from the above discussions with the notations (50)–(52).

**Theorem 2** *Suppose $S > 0$ and $(P_i, K_i)$ defined by* (50) *is updated by Algorithm 3. Then, the followings hold under an initial stabilizing policy $u_1$:*

- *$A_i$ is stable and the closed-loop system $\dot{x} = A_i x + Bw$ is UUB for all $i \in \mathbb{N}$, with each ultimate bound $\|x\| \le 2w_M \cdot \min\{C_i, D_i\}$ for $i \ne 1$.*
- *As $i$ goes to $\infty$, $(P_i, K_i)$ and $Q_i(x,u)$ converge to $(P^*, K^*)$ and $Q^*(x,u)$, respectively.*
- *$B$ can be expressed as $B = (H_{11}^{[i]})^{-1} H_{12}^{[i]}$ for all $i \in \mathbb{N}$.*

**Remark 3** Algorithm 3 does not contain $A$ and $B$ explicitly, which implies that the iteration can be executed without the knowledge of $A$ and $B$. Instead, the exploration $w_t$ is needed to learn the parameters of the controller. This exploration corresponds to the probing noise in DT $Q$-learning (Al-Tamimi *et al.*, 2007; Bradtke & Ydstie, 1994; Lewis & Vamvoudakis, 2010) as well as the exploration in a finite MDP (Powell, 2007; Si *et al.*, 2004; Sutton & Barto, 1998).

**Remark 4** According to Theorem 2, $B$ can be obtained by Algorithm 3 as $B = (H_{11}^{[i]})^{-1} H_{12}^{[i]}$ after 1st-iteration. After $B$ is obtained, other algorithms such as explorized PI (Algorithm 1) and the PI of Vrabie *et al.* (2009), can be also used to find the solutions $P^*$ and $K^*$ online.

### *3.4 Online Implementation*

To implement Algorithm 1 and 3, the iterative formula (28) and (44) should be modified by using the properties among Kronecker product and the operators $\mathrm{vec}(\cdot)$, and $\mathrm{vec}^{+}(\cdot)$ already listed in Section 2.1. By using those properties, we have $x^T P_i x = \overline{x}^T \Gamma \mathrm{vec}^{+}(P_i)$ and

$$\int_t^{t+T} x^T H_{12}^{[i]} w \, d\tau = \left[ \int_t^{t+T} (w \otimes x)^T \, d\tau \right] \mathrm{vec}(H_{12}^{[i]})$$
$$\int_t^{t+T} x^T P_i B w \, d\tau = \left[ \int_t^{t+T} (Bw \otimes x)^T \, \Gamma \, d\tau \right] \mathrm{vec}^{+}(P_i)$$

Now, using the above expressions, both (28) and (44) can be rewritten for any $k \in \mathbb{M}$ by a general compact form

$$\alpha_k^T \, \theta_i = y_k \tag{56}$$

where $y_k := \int_{t+(k-1)T}^{t+kT} c(x, u_i) \, d\tau$. Here, $\theta_i$ and $\alpha_k$ are defined as $\theta_i := \mathrm{vec}^{+}(P_i)$ and $\alpha(t) := \Gamma^T \left[ \overline{x}_{t+(k-1)T} - \overline{x}_{t+kT} + \int_{t+(k-1)T}^{t+kT} Bw \otimes x \, d\tau \right]$ for (28) of Algorithm 1, and

$$\theta_i := \left[ (\mathrm{vec}^{+}(H_{11}^{[i]}))^T, \; 2\,\mathrm{vec}^T(H_{12}^{[i]}) \right]^T \tag{57}$$
$$\alpha_k := \left[ (\overline{x}_{t+(k-1)T} - \overline{x}_{t+kT})^T \Gamma, \; \int_{t+(k-1)T}^{t+kT} (w \otimes x)^T \, d\tau \right]^T \tag{58}$$

for (44) of Algorithm 3. Here, we should find at each iteration the unique $N_{min}$ parameters $\theta_i \in \mathbb{R}^{N_{min}}$ satisfying (56). Here, $N_{min} = q_n$ for (28) in Algorithm 1 and $N_{min} = q_n + nm$ for (44) in Algorithm 3. However, there is only 1-dimensional equation (56), so that the uniqueness of the solution is not guaranteed. This kind of difficulty can be solved by least squares (LS) method (Al-Tamimi *et al.*, 2007; Lee *et al.*, 2009; Vrabie *et al.*, 2009). In this paper, we exactly evaluate $\theta_i$ at each iteration by solving the LS equation:

$$\theta_i = (\mathscr{A} \mathscr{A}^T)^{-1} \mathscr{A} \mathscr{Y}, \tag{59}$$

where $\mathscr{A} := \begin{bmatrix} \alpha_1 & \cdots & \alpha_N \end{bmatrix}$ and $\mathscr{Y} := \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix}^T$ for $N \geq N_{min}$. For the LS (59) uniquely solvable, $\mathscr{A}$ should have the full-rank $N_{min}$ and $N \geq N_{min}$ is obviously necessary for $\mathrm{rank}(\mathscr{A}) = N_{min}$. Note that $\mathscr{A}\mathscr{A}^T$ and $\mathscr{A}\mathscr{Y}$ are expressed as $\mathscr{A}\mathscr{A}^T = \sum_{l=0}^{N-1} \alpha_{k+l} \alpha_{k+l}^T$ and $\mathscr{A}\mathscr{Y} = \sum_{l=0}^{N-1} \alpha_{k+l} y_{k+l}$, respectively. By Proposition 3, $\alpha_k$ should be persistently exciting at least with order $N_{min}$ for the existence of $(\mathscr{A}\mathscr{A}^T)^{-1}$ which is equivalent to $\mathrm{rank}(\mathscr{A}) = N_{min}$.

**Remark 5** if $\alpha_k$ is persistently exciting with order $N_{min}$ and $\mathscr{A}\mathscr{A}^T$ and $\mathscr{A}\mathscr{Y}$ are perturbed to $\mathscr{A}\mathscr{A}^T + \Delta\mathscr{A}\mathscr{A}^T$ and $\mathscr{A}\mathscr{Y} + \Delta\mathscr{A}\mathscr{Y}$ by unexpected noises or uncertainties, resulting in the perturbation $\theta_i + \Delta\theta_i$ by (59), then, by (27) and the argument of linear algebra, the error $\Delta\theta_i$ is bounded as

$$\Delta\theta_i \text{ from } \Delta\mathscr{A}\mathscr{A}^T : \frac{\|\Delta\theta_i\|}{\|\theta_i + \Delta\theta_i\|} \leq \|\Delta\mathscr{A}\mathscr{A}^T\| / \beta_1, \tag{60}$$
$$\Delta\theta_i \text{ from } \Delta\mathscr{A}\mathscr{Y} : \frac{\|\Delta\theta_i\|}{\|\theta_i\|} \leq \frac{\beta_2 \|\Delta\mathscr{A}\mathscr{Y}\|}{\beta_1 \| \sum_{l=0}^{N-1} \alpha_{k+l} y_{k+l} \|}, \tag{61}$$

Here, note that the larger $\beta_1$, the smaller bounds (60)–(61) the relative errors have. In (61), the PE of $y_k$ of order $N_\alpha \geq N$ is desirable to prevent $\| \sum_{l=0}^{N-1} \alpha_{k+l} y_{k+l} \|$ from being zero.

**Remark 6** The condition $\mathrm{rank}(\mathscr{A}) = q_n + nm$ of Algorithm 3 is less conservative than that of the existing DT $Q$-learning (Lewis & Vrabie, 2009) where $N \geq q_{n+m}$ is necessary for updating the DT $Q$-function.

In the sequel, we focus on Algorithm 3 and give a condition for PE of $\alpha_k$. This can be easily extended to Algorithm 1. Suppose the exploration $w_t$ satisfies Assumption 2, and hence, $w_t \equiv w_{k-1}$ holds over the interval $[t + (k-1)T, \, t + kT)$. Then, considering (25) and the notations $x_k$ and $w_k$ in Section 2.4 with $\tau = t$, we have

$$\int_{t+(k-1)T}^{t+kT} w \otimes x \, d\tau = w_{k-1} \otimes E_d^{[i]} x_{k-1}, \tag{62}$$
$$\Delta\overline{x}_k = \overline{A}_d^{[i]} \Delta\overline{x}_{k-1} + \begin{bmatrix} \Xi^{[i]} & \overline{B}_d^{[i]} \end{bmatrix} \Delta\varpi_{k-1}, \tag{63}$$

where the superscript $[i]$ implies that the matrix is formulated with the control gain $K = K_i$. For the further discussion, let $\xi_{k+l}$ be defined as $\xi_{k+l+1} := w_{k+l} \otimes x_{k+l}$. Then, $\alpha_k$ is expressed as $\alpha_k = \mathscr{E} \chi_k$ where $\mathscr{E} := \mathrm{diag}\{-\Gamma^T, I \otimes E_d^{[i]}\}$ and $\chi_k := [\Delta\overline{x}_k^T \;\; \xi_k^T]^T$. Now, noting $\xi_{k+1} = (I \otimes A_d^{[i]})(\xi_k + \Delta w_{k-1} \otimes x_{k-1}) + (I \otimes B_d^{[i]})(w_k \otimes w_{k-1})$ and combining this with (63), the following DT dynamic equation is derived.

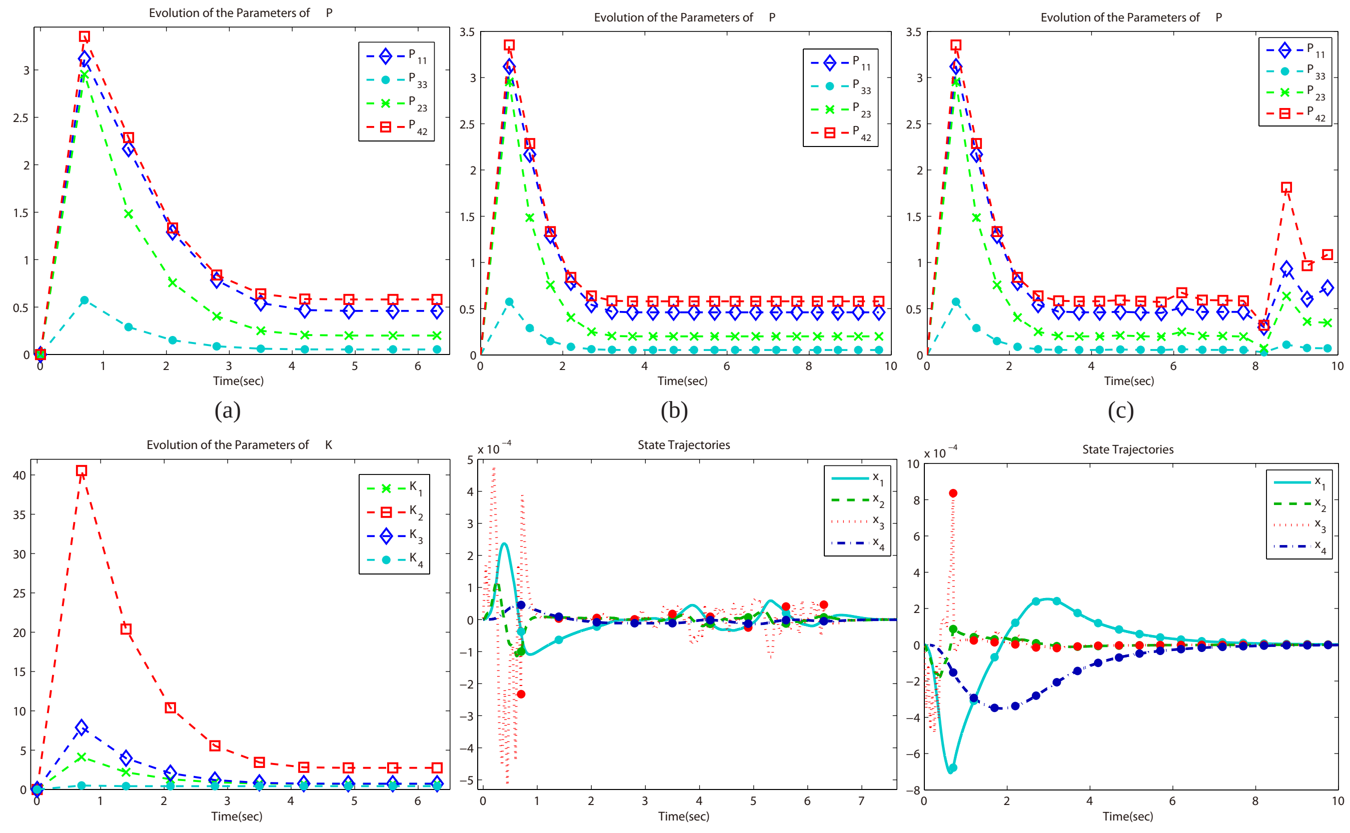
$$\chi_{k+1} = \mathscr{F} \chi_k + \mathscr{G} \phi_k \tag{64}$$

$$\text{where} \begin{cases} \mathscr{F} := \begin{bmatrix} \overline{A}_d^{[i]} & 0 \\ 0 & I \otimes A_d^{[i]} \end{bmatrix}, \; \mathscr{G} := \begin{bmatrix} [\Xi^{[i]} \; \overline{B}_d^{[i]}] & 0 & 0 \\ 0 & I \otimes A_d^{[i]} & I \otimes B_d^{[i]} \end{bmatrix} \\ \phi_k := [\Delta\varpi_{k-1}^T, \;\; (\Delta w_{k-1} \otimes x_{k-1})^T, \;\; (w_k \otimes w_{k-1})^T]^T \end{cases} .$$

**Proposition 4** *Consider the LS solution* (59) *with* (57)–(58) *for Algorithm 3. Suppose $N$ satisfies $N \geq n(n+m)$ and* (64) *is controllable. If $\phi_k$ is persistently exciting with order $n(n+m)$, then, $\alpha_k$ is persistently exciting with order $q_n + nm$.*

**Proof.** First, note that if (64) is controllable and $\phi_k$ is persistently exciting with order $n(n+m) = \dim(\chi_k)$, then, $[\chi_1 \; \cdots \; \chi_N]$ with $N \geq \dim(\chi)$ always has the full rank $n(n+m)$ (Willems *et al.*, 2005, Corollary 2). Since $\mathrm{rank}(\mathscr{E}) = q_n + mn$, we have $\mathrm{rank}(\mathscr{A}) = \mathrm{rank}(\mathscr{E}[\chi_1 \; \cdots \; \chi_N]) = \mathrm{rank}(\mathscr{E}) = q_n + mn$, which implies the PE of $\alpha_k$. □

Proposition 4 states that $\phi_k$ should be at least persistently exciting of order $\dim(\chi)$ to guarantee the uniqueness of $\theta_i$ in (59). For this persistently exciting $\phi_k$, a random exploration $w_k$ sampled from a probability distribution can be a rational choice since a random input is persistently exciting of *any order* (Willems *et al.*, 2005) so that it could increase



(a) (b) (c)

(d) (e) (f)

Fig. 1. (a)(d)(e): simulation results for the integral $Q$-learning; (b):simulation results for the explorized PI with $B$ estimated by $Q$-learning; (c)(f):simulation results for the PI (Vrabie *et al.*, 2009) with $B$ estimated by $Q$-learning.

the order of persistent excitation of $\phi_k$.

By virtue of the differential equations $\dot{V}(t) = c(x,u_i)$ and $\dot{W}(t) = (Xw \otimes x)^T$ for some matrix $X$, the integral terms in $y(t)$ and $\alpha(t)$ can be simplified as $\int_t^{t+T} c(x,u_i)\,d\tau = V(t+T) - V(t)$ and $\int_t^{t+T} (Xw \otimes x)^T\,d\tau = W(t+T) - W(t)$, respectively.

## 4 Simulation Results

In this section, a numerical simulation for the following online load frequency control of a power system (Vrabie *et al.*, 2009) is carried out to verify the effectiveness of the proposed algorithm.

$$A = \begin{bmatrix} -0.0665 & 11.5 & 0 & 0 \\ 0 & -2.5 & 2.5 & 0 \\ -9.5 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 13.736 \\ 0 \end{bmatrix}.$$

In this simulation, the parameters of the value function (2) are given by $Q = I$, $R = 1$ which yields the following LQR solution:

$$P^* = H_{11}^* = \begin{bmatrix} 0.4600 & 0.6911 & 0.0519 & 0.4642 \\ 0.6911 & 1.8668 & 0.2002 & 0.5800 \\ 0.0519 & 0.2002 & 0.0533 & 0.0302 \\ 0.4642 & 0.5800 & 0.0302 & 2.2106 \end{bmatrix},$$

$$\begin{aligned} K^* &= R^{-1}(H_{12}^*)^T \\ &= \begin{bmatrix} 0.7135 & 2.7499 & 0.7323 & 0.4142 \end{bmatrix}. \end{aligned}$$

and we assume zero initial condition which is not acceptable in the absence of exploration $w_t$. Here, $w_t$ is chosen as $w_t = w_k$ for $t \in [kT, (k+1)T)$ where $w_k$ is sampled from a uniform distribution $[-w_M,\ w_M]$ with $w_M$ determined by $w_M = x_M/2\max\{C_i, D_i\}$ with $x_M = 10^{-3}$ to guarantee the UUB $\|x\| \le x_M$ by Theorem 2. Here, $w_M$ cannot be determined before the first iteration since the system $(A,B)$ is unknown. So, at the first iteration $w_M = 10^{-3}$ is used for the exploration. The $Q$-learning parameters are set to $T = 0.05$ [s] and $N = N_{min} = 14$ so the LS solution (59) is obtained every $NT = 14 \times 0.05 = 0.7$ [s].

The simulation results for integral $Q$-learning are described in Figs. 1(a),(d),(e). As can be seen from Figs. 1(a),(d), $P_i$ and $K_i$ are updated simultaneously and shown to converge. Fig. 1(e) shows the state trajectory explored by $w_t$ where $w_t = 10^{-3}$ yields a rather heavy oscillations before the first iteraion, but after 0.7 [s], the UUB $\|x\| \leq 2w_M \min\{C_i, D_i\}$ is used to determine $w_M$, resulting in the very small bounded oscillations. After the parameters converge (6.3 [s]), the exploration $w_t$ is not applied to the system anymore, and the states become stationary and converge to zero.

Next, the explorized PI (Algorithm 1) and the PI of Vrabie *et al.* (2009) are simulated after $B$ is obtained at the first iteration, which is illustrated in Remark 4. Since $N = N_{min}$ ($N_{min} = 10$) is used for both PI algorithm, they can find LQR solution in a reduced time. Here, PI of Vrabie *et al.* (2009) is inherently unable to solve zero initial condition problem alone, but with the excitation made by the $Q$-learning agent, PI of Vrabie *et al.* (2009) can find the LQR solution as shown in Fig 1(c). However, as the state becomes stationary (Fig. 1(f)), some deviation from the solution is introduced after 8.2 [s] since the poor excitation causes the large numerical errors. On the other hand, as shown in Fig. 1(b), this kind of problem never happen when the exploration $w_t$ is injected to the system by explorized PI agent.

## 5 Conclusions

This paper proposed the explorized PI and combined with the introduced $\varepsilon$-integral $Q$-function, presented an integral $Q$-learning scheme which solves CT LQR problem in real time, without knowledge about the system dynamics $A$ and $B$. By virtue of the exploration & singular input perturbation, the assumption of perfectly known $B$ was relaxed, and the stability and convergence was mathematically proven, provided that the initial policy is stabilizing. For the implementation via LS, the PE closely related with explorations was investigated and a sufficient condition for guaranteeing the solvability of LS was given. Though there are still a number of remaining works concerning explorations, PE, robustness, and implementations, these works can be provided as a basis for developing $Q$-learning & adaptive optimal control schemes in CT framework, with stability and convergence considerations.

## Acknowledgements

The authors appreciate the Associate Editor and anonymous reviewers for their valuable suggestions. This work was supported by the Brain Korea 21 Project in 2010–present and by the Human Resources Development of the Korea Insitute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (No. 20104010100590).

## References

Al-Tamimi, A., Abu-Khalaf, M., & Lewis, F. L. (2007) Model-free $Q$-learning designs for discrete-time zero-sum games with application to $H_\infty$ control, *Automatica, 43*(3), 473–481.

Anderson, B. D. O. & Moore, J. B. (1989) *Linear optimal control*, Englewood Cliffs, NJ: Prentice Hall.

Baird III, L. C. (1994) Reinforcement learning in continuous-time: advantage updating, In: *Proc. of ICNN, 4*, 2448–2453.

Balakrishnan, S. N., Ding, D., & Lewis, F. L. (2008) Issues on stability of ADP feedback controllers for dynamical systems, *IEEE Trans. Syst., Man, Cybern.-Part B, 38*(4), 913–917.

Bertsekas, D. P. & Tsitsiklis, J., N. (1996) *Neuro-dynamic programming*, Belmont, MA: Athena Scientific.

Bertsekas, D. P. & Tsitsiklis, J., N. (2005) Dynamic programming and suboptimal control: a survey from ADP to MPC, *European Journal of Control, 11*, 310–334.

Bradtke, S. J. & Ydstie, B. E. (1994) Adaptive linear quadratic control using policy iteration, In: *Proc. ACC*, 3475–3479.

Dong, W. & Farrell, J. A. (2009) Adaptive approximately optimal control of unknown nonlinear systems based on locally weighted learning, In: *Proc. CDC*, 345–350.

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

Kaelbling, L., P. & Moore, A., W. (1996) Reinforcement learning: a survey, *Journal of Artificial Intelligence Research, 4*, 237–285.

Khalil, H. K. (2002) *Nonlinear systems*, Prentice Hall.

Lewis, F. L. & Syrmos, V. (1995) *Optimal control*, 2nd ed. New York: Wiley.

Kleinman, D. (1968) On the iterative technique for Riccati equation computations, *IEEE Trans. Automatic Control, 13*(1), pp. 114–115.

Kokotovic, P., Khalil, H. H., & O'Reilly, J. (1986) Singular Perturbation Methods in Control: Analysis and Design, *Academic Press, Inc.*

Landelius, T. (1997) *Reinforcement learning and distributed local model synthesis*, Ph.D. dissertation, Sweden: Linkoping University.

Lee J. M. & Lee J. H.(2004) Approximate dynamic programming strategies and their applicability for process control: a review and future directions, *Int. Journal of Cont., Auto., and Syst. (IJCAS), 2*(3), 263–278.

Lee, J. Y., Park, J. B., & Choi, Y. H. (2009) Model-free approximate dynamic programming for continuous-time linear systems, In: *Proc. CDC*, 5009–5014.

Lewis, F. L. & Vrabie, D. (2009) Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE circuits and systems magazine*, 9(3), 32–50.

Lewis, F., L. & Vamvoudakis, K., G. (2010) Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, *IEEE Trans. Syst., Man, Cybern.-Part B, 41*(1), 14–25.

Mehta, P. & Meyn, S. (2009) $Q$-learning and Pontryagins minimum principle, In: *Proc. CDC*, 3598–3605.

Murray, J. J., Cox, C. J., Lendaris, G. G., & Saeks, R. (2002) Adaptive dynamic programming, *IEEE Trans. Syst., Mans, and Cybern.-Part C, 32*(2), 140–153.

Kokotovic, P., Khalil, H., H., & O'Reilly, J. (1986) *Singular perturbation methods in control: analysis and design,* Academic Press, Inc..

Powell, W., B. (2007) *Approximate dynamic programming: solving the curses of dimensionality*, Wiley-Interscience.

Prokhorov, D. V. & Wunsch II, D. C. (1997) Adaptive critic designs, *IEEE Trans. Neural Networks, 8*(5), 997–1007.

Si, J., Barto, A. G., Powell, W. B., & Wunsch, D. (2004) *Handbook of learning and approximate dynamic programming,* Wiley-IEEE Press.

Sutton, R. S. & Barto, A. G. (1998) *Reinforcement learning–an introduction*, MIT Press, Cambridge, Massachussetts.

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

Vrabie, D., Pastravanu, O., Abu-Khalaf, M., & Lewis, F. L. (2009) Adaptive optimal control for continuous-time linear systems based on policy iteration, *Automatica, 45*(2), 477–484.

Wang, F., Y., Zhang, H., & Liu, D. Adaptive dynamic programming: an introduction, *IEEE Computational Intelligence Magazine*, *4*(3), 39–47.

Watkins, C. J. C. H. & Dayan, P. (1989) *Learning from delayed rewards,* Ph.D. Dissertation, Cambridge Univ., Cambridge, U.K.

Watkins, C. J. C. H. & Dayan, P. (1992) *Q*-learning, *Machine Learning, 8*, 279–292.

Webos, P., J. (1992) Approximate dynamic programming for real-time control and neural modeling, In: D. A. White and D.A. Sofge, eds., *Handbook of Intelligent Control*, New York: Van Nostrand Reinhold.

Willems, J., C., Rapisarda, P., Markovsky, I., & Moor, B., L., M. (2005) A note on persistency of excitation, *Systems & Control Letters, 54*(4), 325–329.

Zhang, H., Huang, J., Lewis, F., L. (2009) Algorithm and stability of ATC receding horizon control, In: *IEEE Symp. ADPRL*, 28–35.