Proceedings of Machine Learning Research vol 120:1–10, 2020 2nd Annual Conference on Learning for Dynamics and Control

# Hamilton-Jacobi-Bellman Equations for Q-Learning in Continuous Time

**Jeongho Kim** JHKIM206@SNU.AC.KR
*Institute of New Media and Communications, Seoul National University, Seoul 08826, South Korea*

**Insoon Yang** INSOONYANG@SNU.AC.KR
*Department of Electrical and Computer Engineering, Automation and System Research Institute, Seoul National University, Seoul 08826, South Korea*

**Editors:** A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

## Abstract

In this paper, we introduce Hamilton–Jacobi–Bellman (HJB) equations for Q-functions in continuous-time optimal control problems with Lipschitz continuous controls. The standard Q-function used in reinforcement learning is shown to be the unique viscosity solution of the HJB equation. A necessary and sufficient condition for optimality is provided using the viscosity solution framework. By using the HJB equation, we develop a Q-learning method for continuous-time dynamical systems. A DQN-like algorithm is also proposed for high-dimensional state and control spaces. The performance of the proposed Q-learning algorithm is demonstrated using 1-, 10- and 20-dimensional dynamical systems.

**Keywords:** Hamilton-Jacobi-Bellman equation, Optimal control, Q-learning, Reinforcement learning, Deep Q-Networks.

## 1. Introduction

Q-learning is one of the most popular reinforcement learning methods that seek efficient control policies without the knowledge of an explicit system model Watkins and Dayan (1992). The key idea of Q-learning is to combine dynamic programming and stochastic approximation in a way to estimate the optimal state-action value function, also called the *Q-function*, by using trajectory samples. For discrete-time Markov decision processes, Q-learning has been extensively studied (see Bertsekas (2019); Matni et al. (2019) and the references therein), while the literature on continuous-time Q-learning is sparse. In discrete time, the Bellman equation for Q-functions can be defined by using dynamic programming in a straightforward manner. However, the corresponding Bellman equation for continuous-time Q-functions has not yet been fully characterized despite some prior attempts using HJB equations Doya (2000); Munos (2000). A variant of Q-function is used in Lee et al. (2012); Mehta and Meyn (2009), which has a different meaning from the Q-function in reinforcement learning. In other literature, a Q-function similar to that of reinforcement learning was introduced, but with function-valued control input Palanisamy et al. (2015) or heavily utilizing the linear-time-invariant (LTI) system structure Vamvoudakis (2017). A similar model-free approach for LTI systems has been also studied in Jiang and Jiang (2012); Vrabie et al. (2009), although an associated Q-function is not specifically defined. A continuous-time Q-function was also considered to prove the convergence of stochastic approximation Devraj and Meyn (2017).

© 2020 J. Kim & I. Yang.

In this paper, we consider continuous-time deterministic optimal control problems with Lipschitz continuous controls. We show that the associated Q-function corresponds to the unique viscosity solution of a Hamilton–Jacobi–Bellman (HJB) equation in a particular form. In the viscosity solution framework, even when it is not differentiable, the Q-function can be used to verify the optimality of a given control and to design an optimal control strategy. We use the proposed HJB equation to derive an integral equation that the optimal Q-function and optimal control trajectories should satisfy. Based on this equation, we propose a Q-learning algorithm for continuous-time dynamical systems. For high-dimensional state and control spaces, we also propose a DQN-like algorithm by using deep neural networks (DNNs) as a function approximator Mnih et al. (2015). This opens a new avenue of research that connects viscosity solution theory for HJB equations and Q-learning domain. The performance of the proposed Q-learning algorithm is tested through a set of numerical experiments with 1-, 10- and 20-dimensional systems.

## 2. Continuous-Time Q-Functions and HJB Equations

Consider a controlled dynamical system of the form

$$\dot{x}(t) = f(x(t), u(t)), \quad t > 0, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the system state and $u(t) \in \mathbb{R}^m$ is the control input. Let $\mathbb{U} := \{u : [0,T] \to \mathbb{R}^m \mid u \text{ measurable}\}$ be the set of admissible controls. The standard finite-horizon optimal control problem can be formulated as

$$\inf_{u \in \mathbb{U}_1} J_{\boldsymbol{x}}(u) := \inf_{u \in \mathbb{U}_1} \left\{ \int_0^T r(x(t), u(t))\, \mathrm{d}t + q(x(T)) \right\} \tag{2}$$

with $x(0) = \boldsymbol{x}$, where $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $q : \mathbb{R}^n \to \mathbb{R}$ are running and terminal cost functions of interest, respectively, and $\mathbb{U}_1$ is a subset of $\mathbb{U}$. The Q-function $Q : \mathbb{R}^n \times \mathbb{R}^m \times [0,T] \to \mathbb{R}$ of (2) is defined by

$$Q(\boldsymbol{x}, \boldsymbol{u}, t) := \inf_{u \in \mathbb{U}_1} \left\{ \int_t^T r(x(s), u(s))\, \mathrm{d}s + q(x(T)) \;\middle|\; x(t) = \boldsymbol{x}, u(t) = \boldsymbol{u} \right\}, \tag{3}$$

which represents the minimal cost incurred from time $t$ to $T$ when starting from $x(t) = \boldsymbol{x}$ with $u(t) = \boldsymbol{u}$. In particular, when $\mathbb{U}_1 = \mathbb{U}$, the Q-function reduces to the standard optimal value function $v : \mathbb{R}^n \times [0,T] \to \mathbb{R}$, defined by $v(\boldsymbol{x}, t) := \inf_{u \in \mathbb{U}} \left\{ \int_t^T r(x(s), u(s))\, \mathrm{d}s + q(x(T)) \;\middle|\; x(t) = \boldsymbol{x} \right\}$.

**Proposition 1** *Suppose that $\mathbb{U}_1 = \mathbb{U}$. Then, the Q-function (3) corresponds to $v$ for each $\boldsymbol{u} \in \mathbb{R}^m$, i.e., $Q(\boldsymbol{x}, \boldsymbol{u}, t) = v(\boldsymbol{x}, t)$ for all $(\boldsymbol{x}, \boldsymbol{u}, t) \in \mathbb{R}^n \times \mathbb{R}^m \times [0,T]$.*

**Proof** Fix $(\boldsymbol{x}, \boldsymbol{u}, t) \in \mathbb{R}^n \times \mathbb{R}^m \times [0,T]$. Let $\varepsilon$ be an arbitrary positive constant. Then, there exists $u \in \mathbb{U}$ such that $\int_t^T r(x(s), u(s))\, \mathrm{d}s + q(x(T)) < v(\boldsymbol{x}, t) + \varepsilon$, where $x(s)$ satisfies (1) with $x(t) = \boldsymbol{x}$ in the Carathéodory sense: $x(s) = \boldsymbol{x} + \int_t^s f(x(\tau), u(\tau))\, \mathrm{d}\tau$. We now construct a new control $\tilde{u} \in \mathbb{U}_1 = \mathbb{U}$ as $\tilde{u}(s) := \boldsymbol{u}$ if $s = t$; $\tilde{u}(s) := u(s)$ if $s > t$. Such a modification of controls at a single point does not affect the trajectory or the total cost. Therefore, we have

$$v(\boldsymbol{x}, t) \le Q(\boldsymbol{x}, \boldsymbol{u}, t) \le \int_t^T r(x(s), \tilde{u}(s))\, \mathrm{d}s + q(x(T)) < v(\boldsymbol{x}, t) + \varepsilon.$$

Since $\varepsilon$ was arbitrary, we conclude that $v(\boldsymbol{x}, t) = Q(\boldsymbol{x}, \boldsymbol{u}, t)$ for any $\boldsymbol{u} \in \mathbb{R}^m$. ■

Thus, if $\mathbb{U}_1$ is chosen to be the entire set of measurable function $\mathbb{U}$, the Q-function has no additional interesting property. Motivated by this observation, we restrict the control to be a Lipschitz continuous function. Since any Lipschitz continuous function is differentiable almost everywhere, we define the set of admissible controls $\mathbb{U}_1 \subset \mathbb{U}$ as $\mathbb{U}_1 := \{u \in \mathbb{U} \mid \|\dot{u}\|_{L^\infty} \leq M \text{ a.e.}\}$, where $M$ is a fixed constant. Then, for any $u \in \mathbb{U}_1$, there exists a unique measurable function $a : [0, T] \to \mathbb{R}^m$ with $|a(t)| \leq M$ such that the following ODE holds a.e.: $\dot{u}(t) = a(t)$, $0 \leq t \leq T$. Thus, from now on, we will focus on the optimal control problem (2) when the control $u$ is Lipschitz continuous such that $|\dot{u}(t)| \leq M$ a.e. By using the dynamic programming principle, we can deduce that

$$Q(\boldsymbol{x}, \boldsymbol{u}, t) = \inf_{u \in \mathbb{U}_1} \left\{ \int_t^{t+h} r(x(s), u(s))\, \mathrm{d}s + Q(x(t+h), u(t+h), t+h) \Big| x(t) = \boldsymbol{x}, u(t) = \boldsymbol{u} \right\}. \tag{4}$$

To derive the Hamilton-Jacobi equation that the Q-function satisfies, suppose for a moment that $Q \in C^1(\mathbb{R}^n \times \mathbb{R}^m \times [0, T])$. We will discard this regularity assumption on $Q$ by introducing the viscosity solution framework in Section 2.1. Then, the Taylor expansion of $Q$ in (4) yields $\inf_{u \in \mathbb{U}_1} \left\{ \frac{1}{h} \int_t^{t+h} r(x(s), u(s))\, \mathrm{d}s + \partial_t Q + \nabla_{\boldsymbol{x}} Q \cdot f(\boldsymbol{x}, \boldsymbol{u}) + \nabla_{\boldsymbol{u}} Q \cdot \dot{u}(t) + O(h) \right\} = 0$. Letting $h$ tend to zero, we arrive at the following HJB equation for the Q-function:

$$\partial_t Q + \nabla_{\boldsymbol{x}} Q \cdot f(\boldsymbol{x}, \boldsymbol{u}) + \inf_{\mathbf{a} \in \mathbb{R}^m, |\mathbf{a}| \leq M} \{\nabla_{\boldsymbol{u}} Q \cdot \mathbf{a}\} + r(\boldsymbol{x}, \boldsymbol{u}) = 0.$$

Note that $\mathbf{a}^* = -M \nabla_{\boldsymbol{u}} Q / |\nabla_{\boldsymbol{u}} Q|$ minimizes the Hamiltonian, and thus the HJB equation can be expressed as

$$\partial_t Q + \nabla_{\boldsymbol{x}} Q \cdot f(\boldsymbol{x}, \boldsymbol{u}) - M |\nabla_{\boldsymbol{u}} Q| + r(\boldsymbol{x}, \boldsymbol{u}) = 0. \tag{5}$$

In what follows, we uncover several mathematical properties of the HJB equation (5) and the Q-function.

### 2.1. Viscosity Solution: Existence and Uniqueness

In general, the Q-function is not a $C^1$-function. As a weak solution of the HJB equation, we use the framework of *viscosity solutions* Crandall et al. (1984); Crandall and Lions (1983). We begin by defining the viscosity solution of (5) in the following standard manner Bardi and Capuzzo-Dolcetta (1997); Evans (2010):

**Definition 2** *A continuous function $Q : \mathbb{R}^n \times \mathbb{R}^m \times [0, T] \to \mathbb{R}$ is a* viscosity solution *of* (5) *if*

1. *$Q(\boldsymbol{x}, \boldsymbol{u}, T) = q(\boldsymbol{x})$ for all $\boldsymbol{u} \in \mathbb{R}^m$.*
2. *For any $R \in C^1(\mathbb{R}^n \times \mathbb{R}^m \times (0, T))$, if $Q - R$ has a local maximum at $(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0)$, then $\partial_t R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0) + \nabla_{\boldsymbol{x}} R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0) \cdot f(\boldsymbol{x}_0, \boldsymbol{u}_0) - M |\nabla_{\boldsymbol{u}} R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0)| + r(\boldsymbol{x}_0, \boldsymbol{u}_0) \geq 0$.*
3. *For any $R \in C^1(\mathbb{R}^n \times \mathbb{R}^m \times (0, T))$, if $Q - R$ has a local minimum at $(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0)$, then $\partial_t R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0) + \nabla_{\boldsymbol{x}} R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0) \cdot f(\boldsymbol{x}_0, \boldsymbol{u}_0) - M |\nabla_{\boldsymbol{u}} R(\boldsymbol{x}_0, \boldsymbol{u}_0, t_0)| + r(\boldsymbol{x}_0, \boldsymbol{u}_0) \leq 0$.*

From now on, we assume the following regularity conditions on $f$, $r$ and $q$:[1]

- $(A1)$ The functions $f$, $r$ and $q$ are bounded: $\|f\|_{L^\infty} + \|r\|_{L^\infty} + \|q\|_{L^\infty} < C$.
- $(A2)$ The functions $f$, $r$ and $q$ are Lipschitz continuous: $\|f\|_{\text{Lip}} + \|r\|_{\text{Lip}} + \|q\|_{\text{Lip}} < C$, where $\|\cdot\|_{\text{Lip}}$ is a Lipschitz constant of argument.

Then, the HJB equation (5) has a unique viscosity solution, which corresponds to the Q-function.

**Theorem 3** *The Q-function (3) is the unique viscosity solution of the HJB equation (5). Moreover, it is a bounded and Lipschitz continuous function.*

**Proof** To accommodate the Lipschitz continuity constraint on controls, we consider an augmented system $\dot{x}(t) = f(x(t), u(t)), \dot{u}(t) = a(t)$ with $|a(t)| \leq M$, where $u(t)$ and $a(t)$ are interpreted as a new state and a new input, respectively. Let $z(t) := (x(t), u(t))$ be the augmented state, and $F(\boldsymbol{z}, \boldsymbol{a}) := (f(\boldsymbol{x}, \boldsymbol{u}), \boldsymbol{a})$ be the augmented vector field. Then, the Q-function can be expressed as $Q(\boldsymbol{z}, t) = \inf_{|a| \leq M} \left\{ \int_t^T r(z(s))\,\mathrm{d}s + \tilde{q}(z(T)) \,\middle|\, z(t) = \boldsymbol{z} \right\}$, where $\tilde{q}(\boldsymbol{z}) = q(\boldsymbol{x})$. The HJB equation (5) can be rewritten as $\partial_t Q + H(\nabla_{\boldsymbol{z}} Q, \boldsymbol{z}) = 0$ with $Q(\boldsymbol{z}, T) \equiv \tilde{q}(\boldsymbol{z})$, where the Hamiltonian $H = H(\boldsymbol{p}, \boldsymbol{z})$ is defined by $H(\boldsymbol{p}, \boldsymbol{z}) := H(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{z}_1, \boldsymbol{z}_2) = \boldsymbol{p}_1 \cdot f(\boldsymbol{z}_1, \boldsymbol{z}_2) - M|\boldsymbol{p}_2| + r(\boldsymbol{z}_1, \boldsymbol{z}_2)$. By the assumptions (A1) and (A2), we have

$$
\begin{aligned}
&|H(\boldsymbol{p}, \boldsymbol{z}) - H(\boldsymbol{q}, \boldsymbol{z})| \leq |\boldsymbol{p}_1 - \boldsymbol{q}_1||f(\boldsymbol{z}_1, \boldsymbol{z}_2)| + M|\boldsymbol{p}_2 - \boldsymbol{q}_2| \leq (M + \|f\|_{L^\infty})|\boldsymbol{p} - \boldsymbol{q}|, \\
&|H(\boldsymbol{p}, \boldsymbol{z}) - H(\boldsymbol{p}, \boldsymbol{y})| \leq \left(|\boldsymbol{p}|\|f\|_{\text{Lip}} + \|r\|_{\text{Lip}}\right)|\boldsymbol{z} - \boldsymbol{y}|.
\end{aligned}
$$

These imply that the Hamiltonian satisfies the Lipschitz continuity conditions, and thus the standard proof for the existence and the uniqueness of viscosity solution can be directly used (e.g., Evans (2010)). Furthermore, by Bardi and Capuzzo-Dolcetta (1997), the Q-function corresponds to the unique viscosity solution. The boundedness and the Lipschitz continuity of the Q-function can be proved as in Evans (2010). ■

### 2.2. Optimal Controls

To characterize a necessary and sufficient condition for optimality of a control $u \in \mathbb{U}_1$, we consider the function $g_u(s; \boldsymbol{x}, \boldsymbol{u}, t) := \int_t^s r(x(\tau), u(\tau))\,\mathrm{d}\tau + Q(x(s), u(s), s)$, with $x(t) = \boldsymbol{x}, u(t) = \boldsymbol{u}$. By (4), we deduce that the control $u$ is optimal if and only if $s \mapsto g_u(s; \boldsymbol{x}, \boldsymbol{u}, t)$ is a constant function for each $(\boldsymbol{x}, \boldsymbol{u}, t)$. On the other hand, the dynamic programming principle implies that the function $s \mapsto g_u(s; \boldsymbol{x}, \boldsymbol{u}, t)$ is non-decreasing for any control $u \in \mathbb{U}_1$. Thus, the control $u$ is optimal if and only if the function $s \mapsto g_u(s; \boldsymbol{x}, \boldsymbol{u}, t)$ is non-increasing. If the Q-function is differentiable, this implies $\frac{\mathrm{d}}{\mathrm{d}s} g_u(s; \boldsymbol{x}, \boldsymbol{u}, t) = r(x(s), u(s)) + \nabla_{\boldsymbol{x}} Q \cdot f(x(s), u(s)) + \nabla_{\boldsymbol{u}} Q \cdot \dot{u} + \partial_t Q \leq 0$. Since Q-function satisfies HJB equation (5), we have

$$
\begin{aligned}
0 &= \partial_t Q + \nabla_{\boldsymbol{x}} Q \cdot f(x(s), u(s)) - M|\nabla_{\boldsymbol{u}} Q| + r(x(s), u(s)) \\
&\leq \partial_t Q + \nabla_{\boldsymbol{x}} Q \cdot f(x(s), u(s)) + \nabla_{\boldsymbol{u}} Q \cdot \dot{u} + r(x(s), u(s)) \leq 0.
\end{aligned}
$$

1. These assumptions can be relaxed by using a modulus associated with each function as in (Bardi and Capuzzo-Dolcetta, 1997, Chapter III.1–3).

Therefore, when $Q$ is differentiable, $u \in \mathbb{U}_1$ is optimal if and only if $\dot{u} = -M\nabla_{\boldsymbol{u}} Q/|\nabla_{\boldsymbol{u}} Q|$ with $u(t) = \boldsymbol{u}$. To obtain the rigorous principle of optimality when $Q$ is not differentiable, we use generalized derivatives of the Q-function, namely, sub- and superdifferntials of $Q$. The following optimality theorem is a direct application of the classical results in (Bardi and Capuzzo-Dolcetta, 1997, Theorem 3.39, Ch. 3).

**Theorem 4** *Suppose that $f$ and $q$ are continuously differentiable. Then, the trajectory-control pair $(x^*, u^*)$ is optimal if and only if*

$$\dot{u}^*(s) = -M\frac{p_1}{|p_1|} \quad \forall p = (p_0, p_1, p_2) \in D^{\pm}Q(x^*(s), u^*(s), s), \quad \textit{a.e. } s \geq t. \tag{6}$$

**Proof** Note that the controlled system is equivalent to the extended dynamics $\dot{z}(t) = F(z(t), a(t))$ defined in the proof of Theorem 3. Then, the assertions can be obtained by directly applying (Bardi and Capuzzo-Dolcetta, 1997, Theorem 3.39 in Chapter 3) to this augmented system. ■

At a point $(\boldsymbol{x}, \boldsymbol{u}, t)$ where $Q$ is differentiable, the sub- and superdifferentials of $Q$ are identical to the classical derivative of $Q$. Thus, at such a point, we can construct the optimal control by using $\dot{u}^* = -M\frac{\nabla_{\boldsymbol{u}} Q}{|\nabla_{\boldsymbol{u}} Q|}$. At a point where $Q$ is not differentiable, one can choose any control given by (6).

## 3. Q-Learning Using the HJB Equation

In the infinite-horizon case, we consider the following discounted cost function (with $\gamma > 0$):

$$J_{\boldsymbol{x}}(u) = \int_0^\infty e^{-\gamma t} r(x(t), u(t))\,\mathrm{d}t, \quad x(0) = \boldsymbol{x},$$

and the Q-function is defined by $Q(\boldsymbol{x}, \boldsymbol{u}) := \inf_{u \in \mathbb{U}_1} \left\{ \int_0^\infty e^{-\gamma t} r(x(t), u(t))\mathrm{d}t \mid x(0) = \boldsymbol{x}, u(0) = \boldsymbol{u} \right\}$. Again, using the dynamic programming principle, we can derive the following HJB equation:

$$\gamma Q(\boldsymbol{x}, \boldsymbol{u}) - r(\boldsymbol{x}, \boldsymbol{u}) - \nabla_{\boldsymbol{x}} Q \cdot f(\boldsymbol{x}, \boldsymbol{u}) + M|\nabla_{\boldsymbol{u}} Q| = 0. \tag{7}$$

As in Theorem 3, we can show that the Q-function is the unique viscosity solution of the HJB equation (7). A necessary and sufficient condition for optimality can be characterized in a way similar to Theorem 4.

We now discuss how the HJB equation (7) can be used to design a Q-learning algorithm for estimating $Q(\boldsymbol{x}, \boldsymbol{u})$ using sample trajectories. To provide the essential idea, we assume for a moment that $Q$ is differentiable. We then have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} Q(x(t), u(t)) &= \nabla_{\boldsymbol{x}} Q \cdot f(x(t), u(t)) + \nabla_{\boldsymbol{u}} Q \cdot \dot{u}(t) \\ &= \gamma Q(x(t), u(t)) - r(x(t), u(t)) + M|\nabla_{\boldsymbol{u}} Q| + \nabla_{\boldsymbol{u}} Q \cdot \dot{u}(t).\end{aligned}$$

Suppose now that an optimal control is employed, i.e., $\dot{u} := -M\nabla_{\boldsymbol{u}} Q/|\nabla_{\boldsymbol{u}} Q|$. Then, the time derivative of Q-function along the trajectory is further simplified as

$$\frac{\mathrm{d}}{\mathrm{d}t} Q(x(t), u(t)) = \gamma Q(x(t), u(t)) - r(x(t), u(t)). \tag{8}$$

By integrating (8) along the optimal trajectory-control pair, we obtain

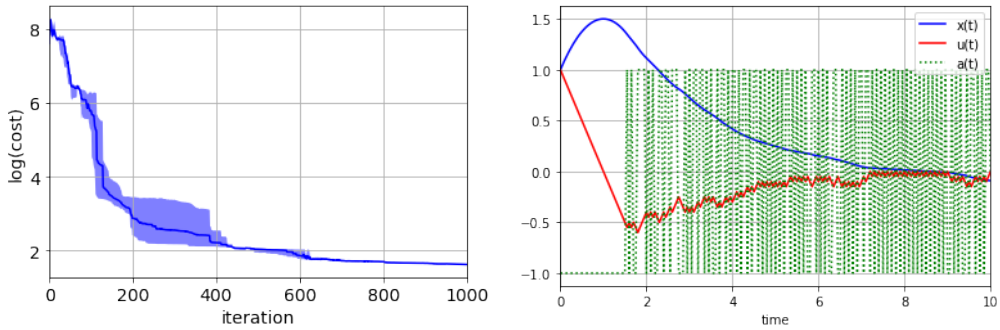
$$Q(\boldsymbol{x}, \boldsymbol{u}) = \int_0^t e^{-\gamma t} r(x(t), u(t))\,\mathrm{d}t + e^{-\gamma t} Q(x(t), u(t)), \quad \forall t \geq 0. \tag{9}$$

However, since the optimal Q-function and optimal trajectories are unknown *a priori*, we iteratively update the Q-function and the control $u$ with $\dot{u} = -M\nabla_{\boldsymbol{u}} Q/|\nabla_{\boldsymbol{u}} Q|$ by using sample data. The iteration is based on the equation (9), which is the characterizing equation of the optimal Q-function. Specifically, for a given control $\dot{u} = a$, we obtain the system trajectory starting from randomly chosen $(\boldsymbol{x}, \boldsymbol{u})$ for time interval $[0, h]$ with small $h > 0$ and collect the sample data $\int_0^h e^{-\gamma t} r(x(t), u(t))\,\mathrm{d}t$, $\boldsymbol{x}' := x(h)$ and $\boldsymbol{u}' := u(h)$. We then update a new estimate for Q-function by using (9) and sample data as $Q(\boldsymbol{x}, \boldsymbol{u}) \leftarrow \int_0^h e^{-\gamma t} r(x(t), u(t))\,\mathrm{d}t + e^{-\gamma h} Q(\boldsymbol{x}', \boldsymbol{u}')$ so that (9) holds asymptotically.

To handle high-dimensional state and control spaces, we propose a DQN-like algorithm by using DNNs as a function approximator. Let $Q_\theta(\boldsymbol{x}, \boldsymbol{u})$ denote the approximate Q-function parameterized by $\theta$. We update the network parameter $\theta$ by minimizing the mean squared error (MSE) loss between $Q_\theta(\boldsymbol{x}, \boldsymbol{u})$ and the target $y = \int_0^h e^{-\gamma t} r(x(t), u(t))\,\mathrm{d}t + e^{-\gamma h} Q_\theta(\boldsymbol{x}', \boldsymbol{u}')$. To enhance the stability of learning procedure, we use $K$ sample points of $(\boldsymbol{x}_i, \boldsymbol{u}_i)$ for defining MSE loss $L(\theta) := \frac{1}{K}\sum_{i=1}^K (Q_\theta(\boldsymbol{x}_i, \boldsymbol{u}_i) - y_i)^2$ and introduce the target network parameter $\theta^-$ in estimating target value $y_i$ as in DQN Mnih et al. (2015). The target network is slowly updated as a weighted sum of $\theta$ and itself as in Lillicrap et al. (2016). The algorithm minimizes the error between the left and right-hand sides of (9) for each iteration, making the $Q_\theta$ asymptotically satisfies (9) as much as possible. The overall procedures are summarized in Algorithm 1. Note that this algorithm does not need the knowledge of an explicit system model as in discrete-time Q-learning or DQN.

**Algorithm 1:** Continuous-Time Q-Learning

Randomly initialize network parameter $\theta$ and initialize the target network parameter $\theta^- \leftarrow \theta$;
Define the domain $\Omega := [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]^n \times [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]^m$;
**for** *iter = 1 to* $N$ **do**
  **if** $\nabla_{\boldsymbol{u}} Q_\theta(\boldsymbol{x}, \boldsymbol{u}) \neq 0$ **then**
    Set $a_\theta(\boldsymbol{x}, \boldsymbol{u})$ as $-M\frac{\nabla_{\boldsymbol{u}} Q_\theta(\boldsymbol{x}, \boldsymbol{u})}{|\nabla_{\boldsymbol{u}} Q_\theta(\boldsymbol{x}, \boldsymbol{u})|}$;
  **else**
    Set $a_\theta(\boldsymbol{x}, \boldsymbol{u})$ as a random vector of length $M$;
  **end**
  Randomly choose $K$ samples of $(\boldsymbol{x}_i, \boldsymbol{u}_i) \in \Omega$;
  **for** *i=1 to* $K$ **do**
    Obtain discounted running cost $R_i := \int_0^h e^{-\gamma t} r(x_i(t), u_i(t))\,\mathrm{d}t$ and the terminal point $(x_i(h), u_i(h))$ by using $a_\theta$;
    Set the target as $y_i := R_i + e^{-\gamma h} Q_{\theta^-}(x_i(h), u_i(h))$;
  **end**
  Update the network parameter $\theta$ by minimizing the MSE loss: $\frac{1}{K}\sum_i (Q_\theta(\boldsymbol{x}_i, \boldsymbol{u}_i) - y_i)^2$;
  Update the target network parameter as $\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-$;
**end**




(*a*) (*b*)

Figure 1: Numerical experiment result for $n = m = 1$: (a) learning curve; (b) controlled trajectory of $x(t), u(t)$ and $a(t)$.

## 4. Numerical Experiments

We consider the following linear system with an exponentially discounted quadratic cost:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad J_{\boldsymbol{x}}(u) := \int_0^\infty e^{-\gamma t}(|x(t)|^2 + |u(t)|^2)\,\mathrm{d}t, \quad x(0) = \boldsymbol{x}, \quad u(0) = \boldsymbol{u},$$

where $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. We restrict the control $u$ as a Lipschitz continuous function with Lipschitz constant 1. The parameters for simulation are chosen as $\boldsymbol{x}_{\min} = \boldsymbol{u}_{\min} = -1, \boldsymbol{x}_{\max} = \boldsymbol{u}_{\max} = 1, \gamma = 0.1, h = 0.05, \tau = 10^{-2}, K = 10$ and $N = 10^3$. As the DNNs for approximating Q-functions, we use fully connected networks consisting of an input layer with $m + n$ nodes, two hidden layers with 128 nodes and an output layer with a single node. For the two hidden layers, we use ReLU activation function. For training the networks, we use Adam optimizer with a learning rate $10^{-3}$ Kingma and Ba (2015).

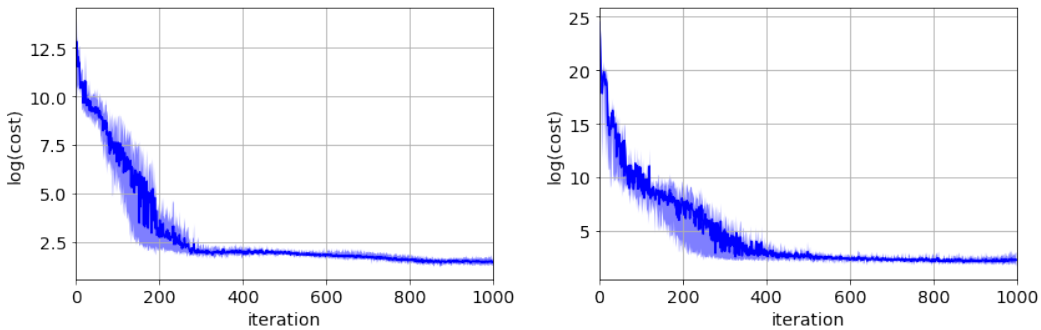
### 4.1. One-dimensional problem

As a toy example for sanity check, we first consider a one dimensional model, where $n = m = 1$, $A = 0$ and $B = 1$. In order to measure the performance of control, we fix the initial state and control $(\boldsymbol{x}, \boldsymbol{u}) = (1, 1)$ and we integrate the running cost over $[0, 10]$. Figure 1(a) shows the log of costs at each iteration of Algorithm 1. The solid line represents the learning curve averaged over five different trials and the shaded region represents the minimum and maximum of the cost over different trials. The running cost rapidly decreases as the network parameters $\theta$ is learned. We also note that the variation over five different trials vanishes as the parameters are learned. Figure 1(b) shows the trajectory of $x(t)$, $u(t)$, and $a(t)$ generated by $\dot{u}(t) = a(t) := -\frac{\nabla_{\boldsymbol{u}} Q_\theta(x(t),u(t))}{|\nabla_{\boldsymbol{u}} Q_\theta(x(t),u(t))|}$ using the learned Q-function $Q_\theta(\boldsymbol{x}, \boldsymbol{u})$ after $10^3$ iteration. The optimal control for this one dimensional problem is to drive both $x(t)$ and $u(t)$ as soon as possible to 0. Starting from $(\boldsymbol{x}, \boldsymbol{u}) = (1, 1)$, this can be done by first driving the control $u(t)$ to negative value so that $x(t)$ moves towards the origin, and then reducing the absolute value of $u(t)$ so that both $x(t)$ and $u(t)$ approaches 0 asymptotically. Such a behavior is observed in Figure 1(b). This confirms that the learned policy is near optimal.

### 4.2. 10- and 20-dimensional problems

For 10- and 20-dimensional problems, where $n = m = 10$ or $n = m = 20$, respectively, we set the coefficient matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ such that each element of $A$ is randomly sampled from $U[0,1]$ and then multiplied by 0.1, and each element of $B$ is randomly sampled from the uniform distribution $U[0,1]$ and then multiplied by 5:

$$A_{ij} = 0.1X_{ij}, \quad B_{ij} = 5Y_{ij}, \quad X_{ij}, Y_{ij} \sim U[0,1].$$




(*a*) (*b*)

Figure 2: Numerical experiment result: (a) learning curve for $n = m = 10$; (b) learning curve for $n = m = 20$.

We integrate the running cost from randomly chosen initial position and control $(\boldsymbol{x}, \boldsymbol{u}) \in [0, 0.1]^{m+n}$. The learning curves are shown in Figure 2. As in the one-dimensional problem, the solid curves denote the average of the running costs over five trials and the shaded regions represent the minimum and maximum of the running costs. The results show that the cost decreases super-exponentially in both 10- and 20-dimensional problems. Note that the $y$-axis of Figure 2(a) and Figure 2(b) are plotted with log-scale, and thus the decay of cost is rapid. Again, the results confirm that our continuous-time Q-learning algorithm presents the desired performance.

## 5. Conclusion

We introduced a Q-function for continuous-time deterministic optimal control problems with Lipschitz continuous control. By using the dynamic programming principle, we derived the corresponding HJB equation and showed that its unique viscosity solution corresponds to the Q-function. An optimality condition was also characterized in terms of the Q-function without the knowledge of system models. Using the HJB equation and the optimality condition, we construct a Q-learning algorithm and its DQN-like approximate version. The simulation results show that the proposed Q-learning algorithm is fast and stable, and that the learned controller presents a good performance, even in the 20-dimensional case.

## Acknowledgments

This work was supported in part by the Creative-Pioneering Researchers Program through SNU, and the Basic Research Lab Program through the National Research Foundation of Korea funded by the MSIT(2020R1C1C1009766).

## References

M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser, 1997.

D. P. Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

M. Crandall and P.-L. Lions. Viscosity solutions of hamilton–jacobi equations. *Transactions of the American Mathematical Society*, 277:1–42, 1983.

M. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton–Jacobi equations. *Transactions of the American Mathematical Society*, 282:487–502, 1984.

A. M. Devraj and S. Meyn. Zap q-learning. In *Advances in Neural Information Processing Systems*, pages 2235–2244, 2017.

K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.

Y. Jiang and Z.-P. Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conferences on Learning Representation (ICLR)*, 2015.

J. Y. Lee, J. B. Park, and Y. H. Choi. Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. *Automatica*, 48:2850–2859, 2012.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *International Conferences on Learning Representation (ICLR)*, 2016.

N. Matni, A. Proutiere, A. Rantzer, and S. Tu. From self-tuning regulators to reinforcement learning and back again. In *IEEE Conference on Decision and Control (CDC)*, 2019.

P. Mehta and S. Meyn. Q-learning and Pontryagin's minimum principle. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3598–3605, 2009.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

R. Munos. A study of reinforcement learning in the continuous case by the means of viscosity solutions. *Machine Learning*, 40:265–299, 2000.

M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb. Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems. *IEEE Transactions on Cybernetics*, 45:165–176, 2015.

K. G. Vamvoudakis. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems & Control Letters*, 100:14–20, 2017.

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

C. J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.