SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2015, Vol. 58 122203:1–122203:15
doi: 10.1007/s11432-015-5462-z

A novel policy iteration based deterministic Q-learning for discrete-time nonlinear systems

WEI QingLai[1] & LIU DeRong[2*]

[1]*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing* 100083*, China*

Received August 11, 2015; accepted October 23, 2015; published online November 16, 2015

Abstract In this paper, a novel iterative Q-learning algorithm, called "policy iteration based deterministic Q-learning algorithm", is developed to solve the optimal control problems for discrete-time deterministic nonlinear systems. The idea is to use an iterative adaptive dynamic programming (ADP) technique to construct the iterative control law which optimizes the iterative Q function. When the optimal Q function is obtained, the optimal control law can be achieved by directly minimizing the optimal Q function, where the mathematical model of the system is not necessary. Convergence property is analyzed to show that the iterative Q function is monotonically non-increasing and converges to the solution of the optimality equation. It is also proven that any of the iterative control laws is a stable control law. Neural networks are employed to implement the policy iteration based deterministic Q-learning algorithm, by approximating the iterative Q function and the iterative control law, respectively. Finally, two simulation examples are presented to illustrate the performance of the developed algorithm.

Keywords adaptive critic designs, adaptive dynamic programming, approximate dynamic programming, Q-learning, policy iteration, neural networks, nonlinear systems, optimal control

Citation Wei Q L, Liu D R. A novel policy iteration based deterministic Q-learning for discrete-time nonlinear systems. Sci China Inf Sci, 2015, 58: 122203(15), doi: 10.1007/s11432-015-5462-z

1 Introduction

Optimal control of nonlinear systems has been the focus of control fields for many decades [1–6]. Dynamic programming is a useful technique in handling optimal control problems, though it is often computationally untenable to perform it to obtain the optimal solutions. Characterized by strong abilities of self-learning and adaptivity, adaptive dynamic programming (ADP), proposed by Werbos [7,8], has demonstrated powerful capability to find the optimal control policy by solving the Hamilton-Jacobi-Bellman (HJB) equation forward-in-time and becomes an important brain-like intelligent optimal control method for nonlinear systems [9–15]. There were several synonyms of ADP, including "adaptive critic designs" [16], "adaptive dynamic programming" [17–20], "approximate dynamic programming" [21], "neuro-dynamic programming" [22], and "reinforcement learning" [23]. Iterative methods have widely been used in ADP to obtain the solution of the HJB equation indirectly and have received more and more attention [24–28]. According to different iteration procedures, iterative ADP algorithms are

* Corresponding author (email: derong@ustb.edu.cn)

classified into policy iteration and value iteration [29], respectively. In policy iteration algorithms, an admissible control law is necessary to initialize the algorithms [30–32]. Policy iteration algorithms for optimal control of continuous-time systems were given in [33,34]. In [35], a policy iteration algorithm for discrete-time nonlinear systems was developed. The complex-valued ADP algorithm was discussed in [36]. It successfully solved the complex-valued nonlinear system optimal control problems. Based on neuro-cognitive psychology, a novel controller based on multiple actor-critic structures was developed for unknown systems in [37]. This controller traded off fast actions based on stored behavior patterns with real-time exploration using current input-output data. The integral reinforcement learning (IRL) algorithm was presented to obtain the iterative control for unknown continuous-time systems with unknown disturbances in [38]. Off-policy learning was used to allow the dynamics to be completely unknown. On the other hand, value iteration algorithms for optimal control of discrete-time nonlinear systems were given in [22]. For value iteration algorithms, a "zero" initial value function [39–42] is generally required to guarantee the convergence properties of the iterative value functions, while the stability of the control system under the iterative control law cannot be guaranteed.

For many traditional iterative ADP algorithms, it is required to build the model of nonlinear systems and then perform the ADP algorithms to derive an improved control policy [43–50]. These iterative ADP algorithms are denoted as "model-based ADP algorithms". In contrast, Q-learning, proposed by Watkins [51,52], is a typical data-based ADP algorithm. In [16,29], Q-learning was named action-dependent heuristic dynamic programming (ADHDP). For Q-learning algorithms, Q function is used instead of performance index function in the traditional iterative ADP algorithms. Q functions depend on both system state and control, which means that they already include the information about the system and the utility function. Hence, it is easier to compute control policies from Q functions than the traditional performance index functions [53]. Because of this merit, Q-learning algorithms are preferred to unknown and model-free systems to obtain the optimal control [52,53]. In [52], a convergence proof of the Q-learning algorithm was proposed under the stochastic environment. However, we should point out that many real-world control systems are deterministic, which need deterministic convergence and stability properties to optimize the control systems. Furthermore, previous iterative Q-learning algorithms were based on value iterations [51–57]. Although the iterative Q functions were convergent to the optimum, stability of the system under the iterative control law could not be guaranteed. Thus, for previous iterative Q-learning algorithms, only the converged optimal control law can be used to control the nonlinear system, and all the iterative control laws during the iteration procedure may not be stable. This makes the computation efficiency of the previous iterative Q-learning algorithms very low. Hence, new iterative Q-learning algorithms need to be developed for deterministic nonlinear systems with property analysis method. This motivates our research.

In this paper, a novel iterative Q-learning algorithm based on *policy iteration* is developed for discrete-time deterministic nonlinear systems, which is denoted as "policy iteration based deterministic Q-learning algorithm". First, the policy iteration based deterministic Q-learning algorithm is derived. The differences between the previous Q-learning algorithms and the developed policy iteration based deterministic Q-learning algorithm are presented. Second, property analysis, including convergence and stability properties, for the developed iterative Q-learning algorithm is established. We emphasize that our theoretical contribution is to establish a new property analysis method to guarantee that any of the iterative control laws is a stable control law and simultaneously to make the iterative Q functions converge to the optimal solution monotonically. Next, neural networks are employed to implement the policy iteration based deterministic Q-learning algorithm by approximating the iterative Q function and iterative control law, respectively. Finally, simulation results will illustrate the effectiveness of the developed algorithm.

2 Problem formulation

In this paper, we will study the following discrete-time deterministic nonlinear system

$$x_{k+1} = F(x_k, u_k),\ k = 0, 1, 2, \ldots, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector and $u_k \in \mathbb{R}^m$ is the control vector. Let x_0 be the initial state and $F(x_k, u_k)$ be the system function. Let $\underline{u}_k = \{u_k, u_{k+1}, \dots\}$ be an arbitrary sequence of controls from k to ∞. The performance index function for state x_0 under the control sequence $\underline{u}_0 = \{u_0, u_1, \dots\}$ is defined as

$$J(x_0, \underline{u}_0) = \sum_{k=0}^{\infty} U(x_k, u_k), \tag{2}$$

where $U(x_k, u_k) > 0$, for $x_k, u_k \neq 0$, is the utility function.

The goal of this paper is to find an optimal control scheme which stabilizes system (1) and simultaneously minimizes the performance index function (2). For convenience of analysis, results of this paper are based on the following assumptions.

Assumption 1. The system (1) is controllable; the system state $x_k = 0$ is an equilibrium state of system (1) under the control $u_k = 0$, i.e., $F(0, 0) = 0$; the feedback control $u_k = u(x_k)$ satisfies $u_k = u(x_k) = 0$ for $x_k = 0$; the utility function $U(x_k, u_k)$ is a positive definite function of x_k and u_k.

Define the control sequence set as $\underline{\mathfrak{U}}_k = \{\underline{u}_k \colon \underline{u}_k = (u_k, u_{k+1}, \dots), \forall u_{k+i} \in \mathbb{R}^m, i = 0, 1, \dots\}$. Then, for a control sequence $\underline{u}_k \in \underline{\mathfrak{U}}_k$, the optimal performance index function is defined as $J^*(x_k) = \min_{\underline{u}_k} \{J(x_k, \underline{u}_k) \colon \underline{u}_k \in \underline{\mathfrak{U}}_k\}$. According to [51,52], the optimal Q function satisfies the Bellman equation of optimality, which is also oftentimes called the Q-Bellman equation [58],

$$Q^*(x_k, u_k) = U(x_k, u_k) + \min_{u_{k+1}} Q^*(x_{k+1}, u_{k+1}). \tag{3}$$

The optimal performance index function satisfies $J^*(x_k) = \min_{u_k} Q^*(x_k, u_k)$. The optimal control law $u^*(x_k)$ can be expressed as $u^*(x_k) = \arg\min_{u_k} Q^*(x_k, u_k)$. We know that if we obtain the optimal Q function $Q^*(x_k, u_k)$, then the optimal control law $u^*(x_k)$ and the optimal performance index function $J^*(x_k)$ can be obtained. However, the optimal Q function $Q^*(x_k, u_k)$ is generally an unknown and non-analytic function, which cannot be obtained directly by (3). Hence, a new policy iteration based Q-learning algorithm is developed to solve the Q function iteratively.

3 Policy iteration based deterministic *Q*-learning algorithm for discrete-time nonlinear systems

In this section, the policy iteration based deterministic Q-learning algorithm will be developed to obtain the optimal controller for discrete-time nonlinear systems. Stability proofs will be given to show that any of the iterative control laws is a stable control law. Convergence and optimality proofs will also be given to show that the iterative Q function will converge to the optimum.

3.1 Derivation of the policy iteration based deterministic *Q*-learning algorithm

For optimal control problems, the developed control scheme must not only stabilize the control systems, but also make the performance index function finite, i.e., the control law must be admissible [39].

Definition 1. A control law $u(x_k)$ is said to be admissible with respect to (2) on a compact set Ω if $u(x_k)$ is continuous on Ω, $u(0) = 0$, $u(x_k)$ stabilizes (1) on Ω, and $\forall x_0 \in \Omega$, $J(x_0)$ is finite.

In the developed policy iteration algorithm, the Q function and control law are updated by iterations, with the iteration index i increasing from 0 to infinity. Let $v_0(x_k)$ be an arbitrary admissible control law. For $i = 0$, let $Q_0(x_k, u_k)$ be the initial iterative Q function constructed by $v_0(x_k)$, i.e.,

$$Q_0(x_k, v_0(x_k)) = \sum_{j=0}^{\infty} U(x_{k+j}, v_0(x_{k+j})). \tag{4}$$

Thus, initial iterative Q function satisfies the following generalized Q-Bellman equation

$$Q_0(x_k, u_k) = U(x_k, u_k) + Q_0(x_{k+1}, v_0(x_{k+1})). \tag{5}$$

Then, the iterative control law is computed by

$$v_1(x_k) = \arg\min_{u_k} Q_0(x_k, u_k). \tag{6}$$

For $i = 1, 2, \ldots$, let $Q_i(x_k, u_k)$ be the iterative Q function constructed by $v_i(x_k)$, which satisfies the following generalized Q-Bellman equation

$$Q_i(x_k, u_k) = U(x_k, u_k) + Q_i(x_{k+1}, v_i(x_{k+1})), \tag{7}$$

and the iterative control law is updated by

$$v_{i+1}(x_k) = \arg\min_{u_k} Q_i(x_k, u_k). \tag{8}$$

3.2 Properties of the policy iteration based deterministic *Q*-learning algorithm

For the policy iteration algorithm of continuous-time nonlinear systems [33], it shows that any of the iterative control laws can stabilize the system. In [35], the stability for the iterative control law and the convergence properties of the policy iteration algorithm for discrete-time nonlinear systems were also proven. This is a merit of the policy iteration algorithm. In this subsection, inspired by [35], we will show that the stability and convergence properties will also hold for the developed policy iteration based deterministic Q-learning algorithm. Before the main theorems, the following lemma is necessary.

Lemma 1. For $i = 0, 1, \ldots$, let $Q_i(x_k, u_k)$ and $v_i(x_k)$ be updated by (5)–(8). Under Assumption 1, the iterative function $Q_i(x_k, u_k)$, $i = 0, 1, \ldots$, is positive definite for x_k and u_k.

Proof. First, let $i = 0$. As the iterative function $Q_0(x_k, v_0(x_k))$ is constructed by $v_0(x_k)$, according to (5), we have

$$Q_0(x_k, v_0(x_k)) = \sum_{j=0}^{\infty} U(x_{k+j}, v_0(x_{k+j})) = U(x_k, v_0(x_k)) + Q_0(x_{k+1}, v_0(x_{k+1})). \tag{9}$$

According to Assumption 1, we have $v_0(x_k) = 0$ as $x_k = 0$. As $U(x_k, u_k)$ is positive definite for x_k and u_k, we have the initial Q function $Q_0(x_k, v_0(x_k)) = \sum_{j=0}^{\infty} U(x_{k+j}, v_0(x_{k+j})) = 0$ as $x_k = 0$. For any $x_k \neq 0$, as $U(x_k, u_k)$ is positive definite for x_k, u_k, we have $Q_0(x_k, v_0(x_k)) > 0$, which proves $Q_0(x_k, v_0(x_k))$ is positive definite for x_k. According to (5), if $x_k = 0$ and $u_k = 0$, according to Assumption 1, we have $x_{k+1} = F(x_k, u_k) = 0$ and $v_0(x_{k+1}) = 0$. Then, we can get

$$Q_0(x_k, u_k) = U(x_k, u_k) + Q_0(x_{k+1}, v_0(x_{k+1})) = 0. \tag{10}$$

If $\|x_k\| + \|u_k\| \neq 0$, we can obtain that $Q_0(x_k, u_k) > 0$, which proves that $Q_0(x_k, u_k)$ is positive definite for x_k and u_k. According to the idea from (9)–(10), for $i = 0, 1, \ldots$, we can prove that iterative function $Q_i(x_k, u_k)$ is positive definite for x_k and u_k. The proof is completed.

Theorem 1. For $i = 0, 1, \ldots$, let $Q_i(x_k, u_k)$ and $v_i(x_k)$ be obtained by the policy iteration algorithm (5)–(8), where $v_0(x_k)$ is an arbitrary admissible control law. If Assumption 1 holds, then for $i = 0, 1, \ldots$, the iterative control law $v_i(x_k)$ stabilizes the nonlinear system (1).

Proof. According to (5) and (7), letting $V_i(x_k) = Q_i(x_k, v_i(x_k))$, for $i = 0, 1, \ldots$, we can get

$$V_i(x_{k+1}) - V_i(x_k) = Q_i(x_{k+1}, v_i(x_{k+1})) - Q_i(x_k, v_i(x_k)) = -U(x_k, v_i(x_k)) < 0. \tag{11}$$

According to Lemma 1 and Assumption 1, the function $V_i(x_k)$ is positive definite for x_k. Then for $i = 0, 1, \ldots$, $V_i(x_k)$ is a Lyapunov function. Thus $v_i(x_k)$ is a stable control law. The proof is completed.

From Theorem 1, we know that for $i = 0, 1, \ldots$, the nonlinear system (1) can be stabilized by the iterative control law. In the following, convergence property of the policy iteration Q-learning algorithm will be proven, which shows that the iterative Q function will be monotonically non-increasing and converge to the optimum.

Theorem 2. For $i = 0, 1, \ldots$, let $Q_i(x_k, u_k)$ and $v_i(x_k)$ be obtained by (5)–(8). If Assumption 1 holds, then the iterative Q function $Q_i(x_k, u_k)$ is monotonically non-increasing and converges to the optimal Q function $Q^*(x_k, u_k)$, as $i \to \infty$, i.e.,

$$\lim_{i\to\infty} Q_i(x_k, u_k) = Q^*(x_k, u_k), \tag{12}$$

which satisfies the optimal Q-Bellman equation (3).

Proof. The statement can be proven by the following four steps.

(1) Show that the iterative Q function $Q_i(x_k, u_k)$ is monotonically non-increasing as i increases, i.e.,

$$Q_{i+1}(x_k, u_k) \leqslant Q_i(x_k, u_k). \tag{13}$$

According to (8), we have

$$Q_i(x_k, v_{i+1}(x_k)) = \min_{u_k} Q_i(x_k, u_k) \leqslant Q_i(x_k, v_i(x_k)). \tag{14}$$

For $i = 0, 1, \ldots$, define a new iterative Q function $\mathcal{Q}_{i+1}(x_k, u_k)$ as

$$\mathcal{Q}_{i+1}(x_k, u_k) = U(x_k, u_k) + Q_i(x_{k+1}, v_{i+1}(x_{k+1})), \tag{15}$$

where $v_{i+1}(x_{k+1})$ is obtained by (8). According to (14), we can obtain

$$\begin{aligned}\mathcal{Q}_{i+1}(x_k, u_k) &= U(x_k, u_k) + Q_i(x_{k+1}, v_{i+1}(x_{k+1}))\\ &= U(x_k, u_k) + \min_{u_{k+1}} Q_i(x_{k+1}, u_{k+1})\\ &\leqslant U(x_k, u_k) + Q_i(x_{k+1}, v_i(x_{k+1}))\\ &= Q_i(x_k, u_k).\end{aligned} \tag{16}$$

As $Q_i(x_k, u_k)$, $\forall\, i = 0, 1, \ldots$, are finite functions for x_k and u_k, for any $\mathcal{N} = 0, 1, \ldots$, there exits a positive function which satisfies $\zeta(\mathcal{N}) \geqslant 0$ that satisfies $Q_{i+1}(x_\mathcal{N}, u_\mathcal{N}) \leqslant Q_i(x_\mathcal{N}, u_\mathcal{N}) + \zeta(\mathcal{N})$. Now, for any $i = 0, 1, \ldots$, we will prove that the following inequality

$$Q_{i+1}(x_k, u_k) \leqslant Q_i(x_k, u_k) + \zeta(\mathcal{N}), \tag{17}$$

holds $\forall\, k = 0, 1, \ldots, \mathcal{N}$. The inequality (17) obviously holds for $k = \mathcal{N}$. Let $u_\mathcal{N} = v_{i+1}(x_\mathcal{N})$, we can get $Q_{i+1}(x_\mathcal{N}, v_{i+1}(x_\mathcal{N})) \leqslant Q_i(x_\mathcal{N}, v_{i+1}(x_\mathcal{N})) + \zeta(\mathcal{N})$. For $k = \mathcal{N} - 1$, we have

$$\begin{aligned}Q_{i+1}(x_{\mathcal{N}-1}, u_{\mathcal{N}-1}) &= U(x_{\mathcal{N}-1}, u_{\mathcal{N}-1}) + Q_{i+1}(x_\mathcal{N}, v_{i+1}(x_\mathcal{N}))\\ &\leqslant U(x_{\mathcal{N}-1}, u_{\mathcal{N}-1}) + Q_i(x_\mathcal{N}, v_{i+1}(x_\mathcal{N})) + \zeta(\mathcal{N})\\ &= \mathcal{Q}_{i+1}(x_{\mathcal{N}-1}, u_{\mathcal{N}-1}) + \zeta(\mathcal{N})\\ &\leqslant Q_i(x_{\mathcal{N}-1}, u_{\mathcal{N}-1}) + \zeta(\mathcal{N}).\end{aligned} \tag{18}$$

So, the inequality (17) holds for $k = \mathcal{N} - 1$. Assume that the inequality (17) holds for $k = \ell + 1$, $\ell = 0, 1, \ldots, \mathcal{N}-1$. As $u_{\ell+1}$ is a free control variable, we can get $Q_{i+1}(x_{\ell+1}, v_{i+1}(x_{\ell+1})) \leqslant Q_i(x_{\ell+1}, v_{i+1}(x_{\ell+1})) + \zeta(\mathcal{N})$. For $k = \ell$ we can get

$$\begin{aligned}Q_{i+1}(x_\ell, u_\ell) &= U(x_\ell, u_\ell) + Q_{i+1}(x_{\ell+1}, v_{i+1}(x_{\ell+1}))\\ &\leqslant U(x_\ell, u_\ell) + Q_i(x_{\ell+1}, v_{i+1}(x_{\ell+1})) + \zeta(\mathcal{N})\\ &= \mathcal{Q}_{i+1}(x_\ell, u_\ell) + \zeta(\mathcal{N})\\ &\leqslant Q_i(x_\ell, u_\ell) + \zeta(\mathcal{N}).\end{aligned} \tag{19}$$

Then, we have (17) holds for $\forall\, k = 0, 1, \ldots, \mathcal{N}$. According to Lemma 1, for $i = 0, 1, \ldots$, $v_{i+1}(x_k)$ is a stable control law. Then, we have $x_\mathcal{N} \to 0$, for $\mathcal{N} \to \infty$. Letting $\mathcal{N} \to \infty$, we know that $\zeta(\mathcal{N}) \to 0$. Hence, for any $i = 0, 1, \ldots$, we have (13) holds, $\forall k = 0, 1, \ldots$. As $Q_i(x_k, u_k)$ is a non-increasing and

lower bounded sequence, i.e., $Q_i(x_k, u_k) \geqslant 0$, the limit of the iterative Q function $Q_i(x_k, u_k)$ exists as $i \to \infty$, i.e.,

$$Q_\infty(x_k, u_k) = \lim_{i\to\infty} Q_i(x_k, u_k). \tag{20}$$

(2) Show that the limit of the iterative Q function $Q_i(x_k, u_k)$ satisfies the optimal Q-Bellman equation, as $i \to \infty$.

According to (19), we can obtain

$$\begin{aligned} Q_\infty(x_k, u_k) &= \lim_{i\to\infty} Q_{i+1}(x_k, u_k) \leqslant Q_{i+1}(x_k, u_k) \leqslant \mathcal{Q}_{i+1}(x_k, u_k) \\ &= U(x_k, u_k) + Q_i(x_{k+1}, v_{i+1}(x_{k+1})) \\ &= U(x_k, u_k) + \min_{u_k} Q_i(x_{k+1}, u_{k+1}). \end{aligned} \tag{21}$$

Letting $i \to \infty$, we obtain $Q_\infty(x_k, u_k) \leqslant U(x_k, u_k) + \min\limits_{u_{k+1}} Q_\infty(x_{k+1}, u_{k+1})$. Let $\zeta > 0$ be an arbitrary positive number. There exists a positive integer p such that

$$Q_p(x_k, u_k) - \zeta \leqslant Q_\infty(x_k, u_k) \leqslant Q_p(x_k, u_k). \tag{22}$$

Hence, we can get

$$\begin{aligned} Q_\infty(x_k, u_k) &\geqslant Q_p(x_k, u_k) - \zeta \\ &= U(x_k, u_k) + Q_p(x_{k+1}, v_p(x_{k+1})) - \zeta \\ &\geqslant U(x_k, u_k) + Q_\infty(x_{k+1}, v_p(x_{k+1})) - \zeta \\ &\geqslant U(x_k, u_k) + \min_{u_{k+1}} Q_\infty(x_{k+1}, u_{k+1}) - \zeta. \end{aligned} \tag{23}$$

Since ζ is arbitrary, we have $Q_\infty(x_k, u_k) \geqslant U(x_k, u_k) + \min\limits_{u_{k+1}} Q_\infty(x_{k+1}, u_{k+1})$. Thus, we obtain

$$Q_\infty(x_k, u_k) = U(x_k, u_k) + \min_{u_{k+1}} Q_\infty(x_{k+1}, u_{k+1}). \tag{24}$$

Next, let $\mu(x_k)$ be an arbitrary admissible control law, and define a new function $\mathcal{P}(x_k, u_k)$, which satisfies

$$\mathcal{P}(x_k, u_k) = U(x_k, u_k) + \mathcal{P}(x_{k+1}, \mu(x_{k+1})). \tag{25}$$

Then, we can declare the third step of the proof.

(3) Show that for an arbitrary admissible control law $\mu(x_k)$, the converged Q function $Q_\infty(x_k, u_k)$ satisfies $Q_\infty(x_k, u_k) \leqslant \mathcal{P}(x_k, u_k)$.

The statement can be proven by mathematical induction. As $\mu(x_k)$ is an admissible control law, we have $x_k \to 0$ as $k \to \infty$. Without loss of generality, let $x_\mathcal{N} = 0$ where $\mathcal{N} \to \infty$. According to (25), we have

$$\begin{aligned} \mathcal{P}(x_k, u_k) =& U(x_k, u_k) + \lim_{\mathcal{N}\to\infty} \{U(x_{k+1}, \mu(x_{k+1})) + U(x_{k+2}, \mu(x_{k+2})) + \cdots \\ &+ U(x_{\mathcal{N}-1}, \mu(x_{\mathcal{N}-1})) + \mathcal{P}(x_\mathcal{N}, \mu(x_\mathcal{N}))\}, \end{aligned} \tag{26}$$

where $x_\mathcal{N} = 0$. According to (24), the function $Q_\infty(x_k, u_k)$ can be expressed as

$$\begin{aligned} Q_\infty(x_k, u_k) =& U(x_k, u_k) + \lim_{\mathcal{N}\to\infty} \{U(x_{k+1}, v_\infty(x_{k+1})) + U(x_{k+1}, v_\infty(x_{k+1})) \\ &+ \cdots + U(x_{\mathcal{N}-1}, v_\infty(x_{\mathcal{N}-1})) + Q_\infty(x_\mathcal{N}, u_\mathcal{N})\} \\ =& U(x_k, u_k) + \lim_{\mathcal{N}\to\infty} \left\{ \min_{u_{k+1}} \left\{ U(x_{k+1}, u_{k+1}) + \min_{u_{k+2}} \left\{ U(x_{k+2}, u_{k+2}) \right. \right. \right. \end{aligned}$$

$$+\cdots+\min_{u_{\mathcal{N}-1}}\left\{U(x_{\mathcal{N}-1},u_{\mathcal{N}-1})+\min_{u_{\mathcal{N}}}Q_\infty(x_{\mathcal{N}},u_{\mathcal{N}})\right\}\Bigg\}\Bigg\}\Bigg\}. \tag{27}$$

As $v_\infty(x_k)$ is an admissible control law, we can get $x_{\mathcal{N}}=0$ where $\mathcal{N}\to\infty$, which means $Q_\infty(x_{\mathcal{N}},u_{\mathcal{N}})=\mathcal{P}(x_{\mathcal{N}},u_{\mathcal{N}})=0$. For $\mathcal{N}-1$, according to (24), we can obtain

$$\begin{aligned}\mathcal{P}(x_{\mathcal{N}-1},u_{\mathcal{N}-1})&=U(x_{\mathcal{N}-1},u_{\mathcal{N}-1})+\mathcal{P}(x_{\mathcal{N}},\mu(x_{\mathcal{N}}))\\&\geqslant U(x_{\mathcal{N}-1},u_{\mathcal{N}-1})+\min_{u_{\mathcal{N}}}\mathcal{P}(x_{\mathcal{N}},u_{\mathcal{N}})\\&=U(x_{\mathcal{N}-1},u_{\mathcal{N}-1})+\min_{u_{\mathcal{N}-1}}Q_\infty(x_{\mathcal{N}},u_{\mathcal{N}})\\&=Q_\infty(x_{\mathcal{N}-1},u_{\mathcal{N}-1}).\end{aligned} \tag{28}$$

Assume that the statement holds for $k=\ell+1$, $\ell=0,1,\ldots$. Then for $k=l$ we have

$$\begin{aligned}\mathcal{P}(x_\ell,u_\ell)&=U(x_\ell,u_\ell)+\mathcal{P}(x_{\ell+1},\mu(x_{\ell+1}))\\&\geqslant U(x_\ell,u_\ell)+\min_{u_{\ell+1}}\mathcal{P}(x_{\ell+1},u_{\ell+1})\\&\geqslant U(x_\ell,u_\ell)+\min_{u_{\ell+1}}Q_\infty(x_{\ell+1},u_{\ell+1})\\&=Q_\infty(x_\ell,u_\ell).\end{aligned} \tag{29}$$

Hence for x_k,u_k, $k=0,1,\ldots$, the inequality

$$Q_\infty(x_k,u_k)\leqslant\mathcal{P}(x_k,u_k) \tag{30}$$

holds. Mathematical induction is completed.

(4) Show that the converged function $Q_\infty(x_k,u_k)$ equals to its optimal function $Q^*(x_k,u_k)$.

According to the definition of $Q^*(x_k,u_k)$, for $i=0,1,\ldots$, we have

$$\begin{aligned}Q_i(x_k,u_k)&=U(x_k,u_k)+Q_i(x_{k+1},v_i(x_{k+1}))\\&=U(x_k,u_k)+\sum_{j=1}^{\infty}U(x_{k+j},v_i(x_{k+j}))\\&\geqslant U(x_k,u_k)+\min_{\underline{u}_{k+1}^{\infty}}\sum_{j=1}^{\infty}U(x_{k+j},u_{k+j})\\&=U(x_k,u_k)+Q^*(x_{k+1},u^*(x_{k+1}))\\&=Q^*(x_k,u_k).\end{aligned} \tag{31}$$

Letting $i\to\infty$, we obtain

$$Q_\infty(x_k,u_k)\geqslant Q^*(x_k,u_k). \tag{32}$$

On the other hand, for an arbitrary admissible control law $\mu(x_k)$, we have (30) holds. Let $\mu(x_k)=u^*(x_k)$, where $u^*(x_k)$ is an optimal control law. Then, we get

$$Q_\infty(x_k,u_k)\leqslant Q^*(x_k,u_k). \tag{33}$$

According to (32) and (33), we can obtain (12). The proof is completed.

In Theorem 1, we have proven that for $i=0,1,\ldots$, the iterative control law is stable. According to the analysis of Theorem 2, the iterative control law can be enhanced as an admissible control law.

Theorem 3. For $i=0,1,\ldots$, let $Q_i(x_k,u_k)$ and $v_i(x_k)$ be obtained by (5)–(8), where $v_0(x_k)$ is an arbitrary admissible control law. If Assumption 1 holds, then for $i=0,1,\ldots$, the iterative control law $v_i(x_k)$ is admissible.

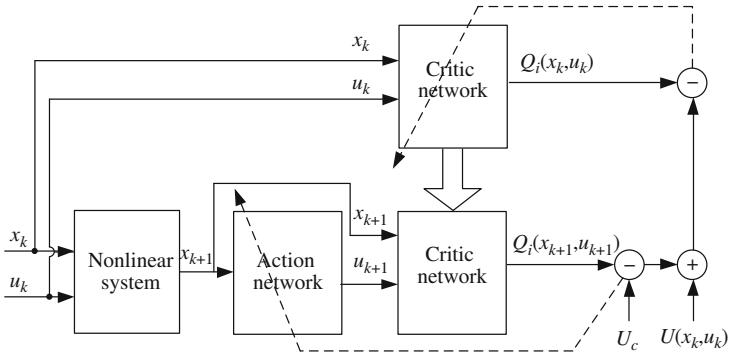

Figure 1 The structure diagram of the policy iteration based deterministic Q-learning algorithm.

Proof. Let $i = 0$. According to (5), we obtain

$$\begin{aligned} Q_0(x_k, u_k) &= U(x_k, u_k) + Q_0(x_{k+1}, v_0(x_{k+1})) \\ &= U(x_k, u_k) + \sum_{j=1}^{\infty} U(x_{k+j}, v_0(x_{k+j})) \\ &\geqslant Q_1(x_k, u_k) \\ &= U(x_k, u_k) + \sum_{j=1}^{\infty} U(x_{k+j}, v_1(x_{k+j})). \end{aligned} \tag{34}$$

As $Q_0(x_k, u_k)$ is finite for x_k, u_k, we have $\sum_{j=1}^{\infty} U(x_{k+j}, v_1(x_{k+j})) < \infty$, which means $v_1(x_k)$ is admissible. By mathematical induction, we can prove $v_i(x_k)$ is admissible for $i = 0, 1, \ldots$. The proof is completed.

4 Neural network implementation for the policy iteration based deterministic Q-learning algorithm

In this paper, backpropagation (BP) neural networks (NNs) are used to approximate $v_i(x_k)$ and $Q_i(x_k, u_k)$, respectively. The number of hidden layer neurons is denoted by τ. The weight matrix between the input layer and hidden layer is denoted by Y. The weight matrix between the hidden layer and output layer is denoted by W. Then the output of three-layer NN is represented by $\hat{F}(X, Y, W) = W^{\mathrm{T}}\sigma(Y^{\mathrm{T}}X + b)$, where $\sigma(Y^{\mathrm{T}}X + b) \in R^{\ell}$, $[\sigma(z)]_i = \frac{\mathrm{e}^{z_i} - \mathrm{e}^{-z_i}}{\mathrm{e}^{z_i} + \mathrm{e}^{-z_i}}$, $i = 1, \ldots, \tau$ are the activation functions and b is the threshold value. For convenience of analysis, only the hidden-output weight matrix W is updated during the NN training, while the input-hidden weights are fixed [25]. Hence, in the following, the NN function is simplified by the expression $\hat{F}_N(X, W) = W^{\mathrm{T}}\sigma_N(X)$, where $\sigma_N(X) = \sigma(Y^{\mathrm{T}}X + b)$.

There are two NNs, which are critic and action networks, respectively, to implement the developed Q-learning algorithm. Both NNs are chosen as three-layer BP networks. The whole structure diagram is shown in Figure 1.

4.1 The critic network

For $i = 0, 1, \ldots$, the goal of the critic network is to approximate the iterative function $Q_{i+1}(x_k, u_k)$. In the critic network, the state x_k and the control u_k are used as the input and the output is formulated as $\hat{Q}_i^j(x_k, u_k) = W_{ci}^{j\mathrm{T}}\sigma(Z_c(k))$, where $Z_c(k) = Y_c^{\mathrm{T}}\mathcal{Z}(k) + b_c$, $\mathcal{Z}(k) = [x_k^{\mathrm{T}}, u_k^{\mathrm{T}}]^{\mathrm{T}}$, and Y_c and b_c are the given weight matrix and threshold. Define the error function for the critic network as $e_{ci}^j(k) = \hat{Q}_i^j(x_k, u_k) - Q_i(x_k, u_k)$, where $Q_i(x_k, u_k)$ is the target Q function which satisfies (7). The objective function to be minimized in the critic network training is $E_{ci}^j(k) = \frac{1}{2}(e_{ci}^j(k))^2$. So the gradient-based weight update rule [23] for the critic network is given by

$$W_{ci}^{j+1} = W_{ci}^j + \Delta W_{ci}^j = W_{ci}^j - \alpha_c \left[\frac{\partial E_{ci}^j(k)}{\partial e_{ci}^j(k)} \frac{\partial e_{ci}^j(k)}{\partial \hat{Q}_i^j(x_k, u_k)} \frac{\partial \hat{Q}_i^j(x_k, u_k)}{\partial W_{ci}^j} \right] = W_{ci}^j - \alpha_c e_{ci}^j(k)\sigma(Z_c(k)), \tag{35}$$

where $\alpha_c > 0$ is the learning rate of critic network. If the training precision is achieved, then we say that $Q_i(x_k, u_k)$ can be approximated by the critic network.

4.2 The action network

The principle in adapting the action network is to indirectly back-propagate the error between the desired objective, denoted by U_c and the iterative function $Q_i(x_k, u_k)$. According to the definition of Q function in (3), we know that $U_c \equiv 0$. From Figure 1, according to an array of x_k and u_k, we can obtain x_{k+1}, immediately. Then, the target of the iterative control law $v_i(x_{k+1})$ can be defined as (8). In the action network, the state x_{k+1} is used as input to create the iterative control law as the output of the network. The output can be formulated as $\hat{v}_i^j(x_{k+1}) = W_{ai}^{j\mathrm{T}}\sigma(Z_a(k+1))$, where $Z_a(k+1) = Y_a^{\mathrm{T}}x_{k+1} + b_a$, and Y_a and b_a are the given weight matrix and threshold. Define the output error of the action network as $e_{ai}^j(k+1) = \hat{v}_i^j(x_{k+1}) - v_i(x_{k+1})$.

The weights of the action network are updated to minimize the following performance error measure $E_{ai}^j(k+1) = \frac{1}{2}(e_{ai}^j(k+1))^{\mathrm{T}}(e_{ai}^j(k+1))$. The weights updating algorithm is similar to the one for the critic network. By the gradient descent rule [23], we can obtain

$$\begin{aligned} W_{ai}^{j+1} &= W_{ai}^j + \Delta W_{ai}^j = W_{ai}^j - \beta_a \left[\frac{\partial E_{ai}^j(k+1)}{\partial e_{ai}^j(k+1)} \frac{\partial e_{ai}^j(k+1)}{\partial \hat{v}_i^j(x_{k+1})} \frac{\partial \hat{v}_i^j(x_{k+1})}{\partial W_{ai}^j}\right] \\ &= W_{ai}^j - \beta_a \sigma(Z_a(k+1))(e_{ai}^j(k+1))^{\mathrm{T}}, \end{aligned} \tag{36}$$

where $\beta_a > 0$ is the learning rate of the action network. If the training precision is achieved, then we say that the iterative control law $v_i(x_{k+1})$ can be approximated by the action network.

Finally, inspired by [35], the convergence of NN weights is proven which guarantees that the iterative Q function and iterative control law can be approximated by the critic and action networks, respectively.

Theorem 4. Let the target iterative Q function and the target iterative control law be expressed by $Q_{i+1}(x_k) = W_{ci}^{*\mathrm{T}}\sigma(Z_c(k))$ and $v_i(x_{k+1}) = W_{ai}^{*\mathrm{T}}\sigma(Z_a(k+1))$, respectively. Let the critic and action networks be trained by (35) and (36), respectively. If the learning rates α_c and β_a are both small enough, then the critic network weights W_{ci} and action network weights W_{ai} are asymptotically convergent to the optimal weights W_{ci}^* and W_{ai}^*, respectively.

Proof. Let $\bar{W}_{ci}^j = W_{ci}^j - W_{ci}^*$ and $\bar{W}_{ai}^j = W_{ai}^j - W_{ai}^*$. From (35) and (36), we have

$$\bar{W}_{ci}^{j+1} = \bar{W}_{ci}^j - \alpha_c e_{ci}^j(k)\sigma(Z_c(k)), \quad \bar{W}_{ai}^{j+1} = \bar{W}_{ai}^j - \beta_a e_{ai}^j(k+1)\sigma(Z_a(k+1)).$$

Consider the following Lyapunov function candidate

$$L(\bar{W}_{ci}^j, \bar{W}_{ai}^j) = \mathrm{tr}\left\{\bar{W}_{ci}^{j\mathrm{T}}\bar{W}_{ci}^j + \bar{W}_{ai}^{j\mathrm{T}}\bar{W}_{ai}^j\right\}. \tag{37}$$

Then, the difference of the Lyapunov function candidate (37) is given by

$$\begin{aligned} \Delta L(\bar{W}_{ci}^j, \bar{W}_{ai}^j) &= \mathrm{tr}\left\{\bar{W}_{ci}^{(j+1)\mathrm{T}}\bar{W}_{ci}^{j+1} + \bar{W}_{ai}^{(j+1)\mathrm{T}}\bar{W}_{ai}^{j+1}\right\} - \mathrm{tr}\left\{\bar{W}_{ci}^{j\mathrm{T}}\bar{W}_{ci}^j + \bar{W}_{ai}^{j\mathrm{T}}\bar{W}_{ai}^j\right\} \\ &= \alpha_c\left\|e_{ci}^j(k)\right\|^2\left(-2 + \alpha_c\|\sigma(Z_c(k))\|^2\right) + \beta_a\left\|e_{ai}^j(k+1)\right\|^2\left(-2 + \beta_a\|\sigma(Z_a(k+1))\|^2\right). \end{aligned}$$

According to the definition of $\sigma(\cdot)$, we know that $\|\sigma(Z_c(k))\|^2$ and $\|\sigma(Z_a(k+1))\|^2$ are both finite for $\forall\, Z_c(k), Z_a(k)$. Thus, if α_c and β_a are both small enough that satisfy $\alpha_c \leqslant \frac{2}{\|\sigma(Z_c(k))\|^2}$ and $\beta_a \leqslant \frac{2}{\|\sigma(Z_a(k+1))\|^2}$, then we have $\Delta L(\bar{W}_{ci}^j, \bar{W}_{ai}^j) < 0$. The proof is completed.

5 Simulation study

In this section, we choose two examples for numerical experiments to evaluate the performance of our policy iteration based deterministic Q-learning algorithm.

5.1 Example 1

First, the performance of the developed Q-learning algorithm will be verified by linear system in [59], where the results can be verified for traditional linear optimal control theories. Let us consider the spring-mass-damper system $M\frac{\mathrm{d}^2y}{\mathrm{d}t^2}+b\frac{\mathrm{d}y}{\mathrm{d}t}+\kappa y=u$,where y is the position and u is the control input. Let $M=0.1$ kg denote the mass of object. Let $\kappa=2$ kgf/m be the stiffness coefficient of spring and let $b=0.1$ be the wall friction. Let $x_1=y$ and $x_2=\dfrac{\mathrm{d}y}{\mathrm{d}t}$. Discretizing the system function using Euler method with the sampling interval $\Delta t=0.1$ s leads to

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix}=\begin{bmatrix} 1 & \Delta T \\ -\frac{\kappa}{M}\Delta T & 1-\frac{b}{M}\Delta T \end{bmatrix}\begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix}+\begin{bmatrix} 0 \\ \frac{\Delta T}{M} \end{bmatrix}u_k. \tag{38}$$

Let the initial state be $x_0=[1,1]^{\mathrm{T}}$. Let the performance index function be expressed by (2). The utility function is expressed as $U(x_k,u_k)=x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k$, where $Q=I$, $R=I$ and I denotes the identity matrix with suitable dimensions.

Define the state and control spaces as $\Omega_x=\{x_k\,|\,-1\leqslant x_{1k}\leqslant 1,-1\leqslant x_{2k}\leqslant 1\}$ and $\Omega_u=\{u_k\,|\,-1\leqslant u_k\leqslant 1\}$, respectively. We randomly choose $p=5000$ training data in $\Omega_x\times\Omega_u$ to implement the policy iteration based deterministic Q-learning algorithm to obtain the optimal control law. Neural networks are used to implement the developed Q-learning algorithm. The critic network and the action network are chosen as three-layer back-propagation (BP) neural networks with the structures of 3–8–1 and 2–8–1, respectively. For each iteration step, the critic network and the action network are trained for 200 steps using the learning rate of $\alpha_c=\beta_a=0.02$ so that the neural network training error becomes less than 10^{-5}. Let the iterative function $Q_i(x_k,v_i(x_k))$ be defined as

$$Q_i(x_k,v_i(x_k))=\min_{u_k}Q_i(x_k,u_k). \tag{39}$$

For system (38), we can obtain an admissible control law $v_0(x_k)=Kx_k$, where $K=[0.13,-0.17]^{\mathrm{T}}$. Initialized by the admissible control law $v_0(x_k)$, we implement the developed algorithm for $i=15$ iterations to reach the computation precision $\varepsilon=0.01$. The plots of the iterative function $Q_i(x_k,v_i(k))$ are shown in Figure 2(a), where we let "In" denote "initial iteration" and let "Lm" denote "limiting iteration".

From Figure 2(a) we can see that by the developed policy iteration based deterministic Q-learning algorithm, the iterative Q function is monotonically non-increasing and converges to its optimum after 15 iterations. The iterative trajectories of system states and controls are shown in Figure 2 (b) and (c), respectively. From Figure 2 (b) and (c) we can see that the iterative system states and iterative controls are both convergent to the optimum. Under an arbitrary iterative control law, the system (38) is stable, which justifies the stability properties of the developed policy iteration based deterministic Q-learning algorithm. The optimal states and control trajectories are shown in Figure 2(d).

On the other hand, for the linear system (38), we know that the optimal Q function can be expressed as $Q^*(x_k,u_k)=\mathcal{Z}^{\mathrm{T}}(k)P^*\mathcal{Z}(k)$, $Z_k=[x_k^{\mathrm{T}},u_k^{\mathrm{T}}]^{\mathrm{T}}$. According to the discrete algebraic Riccati equation, we know that $P^*=[27.98\,0.51\,-1.99;\,0.51\,3.13\,1.89;\,-1.99\,1.89\,2.89]$. The optimal control law can be expressed as $u^*(x_k)=Kx_k$, where $K^*=[0.69\;-0.65]$, which can obtain the same trajectories as in Figure 2(d). Hence, the effectiveness of the developed policy iteration based deterministic Q-learning algorithm can be verified for linear systems.

5.2 Example 2

We now examine the performance of the developed Q-learning algorithm in a nonlinear torsional pendulum system [23]. The dynamics of the pendulum is as follows

$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t}=\omega, \\ J\dfrac{\mathrm{d}\omega}{\mathrm{d}t}=u-Mgl\sin\theta-f_d\dfrac{\mathrm{d}\theta}{\mathrm{d}t}, \end{cases} \tag{40}$$

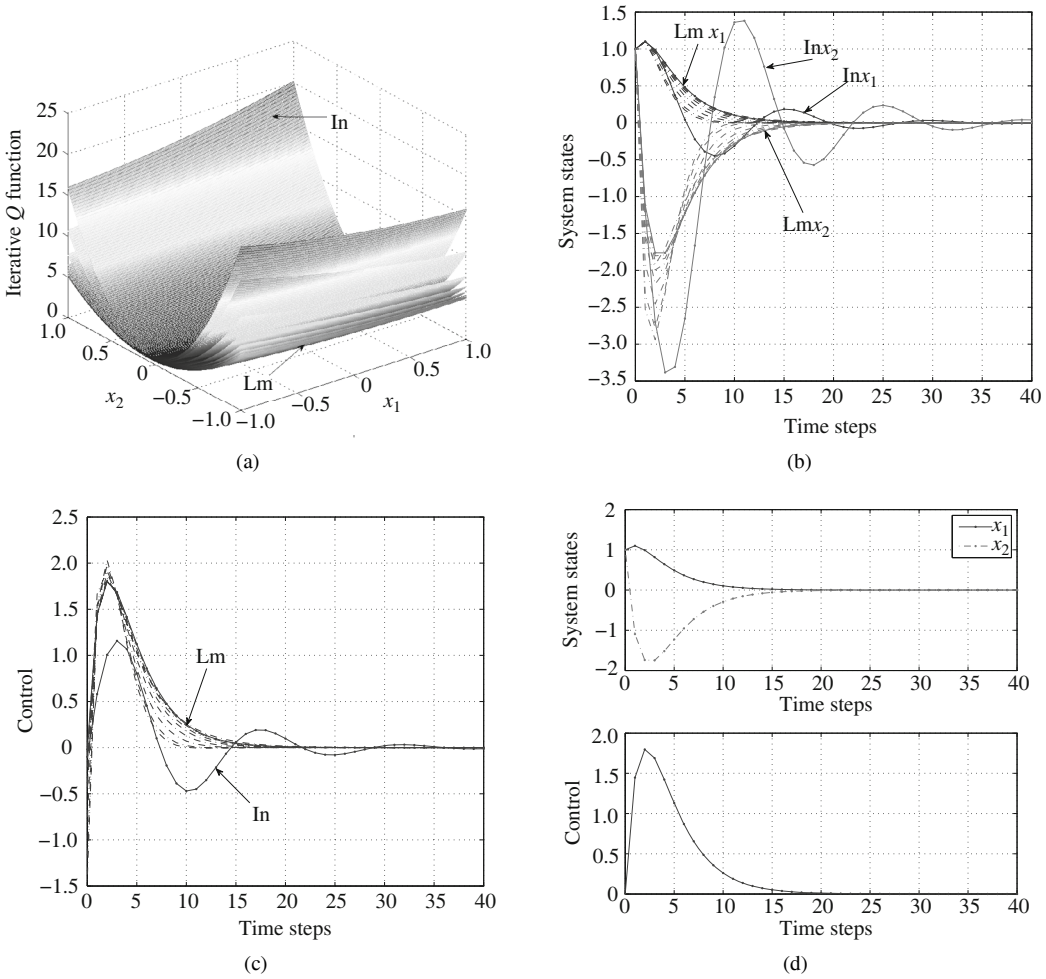

Figure 2 The policy iteration based Q-learning algorithm for linear system. (a) The plots of the iterative Q function; (b) the iterative state trajectories; (c) the iterative control trajectories; (d) the optimal state and control trajectories.

where $M = 1/3$ kg and $l = 2/3$ m are the mass and length of the pendulum bar, respectively. The system states are the current angle θ and the angular velocity ω. Let $J = 4/3Ml^2$ and $f_d = 0.2$ be the rotary inertia and frictional factor, respectively. Let $g = 9.8$ m/s^2 be the gravity. Discretization of the system function using Euler method with the sampling interval $\Delta t = 0.1$ s leads to

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} 0.1x_{2k} + x_{1k} \\ -0.49\sin(x_{1k}) - 0.1f_d x_{2k} + x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_k, \tag{41}$$

where $x_{1k} = \theta_k$ and $x_{2k} = \omega_k$. Let the initial state be $x_0 = [1, -1]^{\mathrm{T}}$. Let the utility function be the quadratic form which is the same as in Example 1. Let the structures of the critic and action networks be 3–12–1 and 2–12–1, respectively. We randomly choose $p = 20000$ training data in $\Omega_x \times \Omega_u$ to implement the policy iteration based deterministic Q-learning algorithm to obtain the optimal control law. For each iteration step, the critic network and the action network are trained for 1000 steps using the learning rate of $\alpha_c = \beta_a = 0.01$ so that the neural network training error becomes less than 10^{-5}. For the nonlinear system (41), we can obtain an admissible control law using action network, i.e., $v_0(x_k) = W_{a,\mathrm{initial}}\sigma(Y_{a,\mathrm{initial}}x_k + b_{a,\mathrm{initial}})$, according to Algorithm 1 in [35] and the detailed method is omitted here. Initialized by the admissible control law $v_0(x_k)$, we implement the developed algorithm for $i = 25$ iterations to reach the computation precision $\varepsilon = 0.01$. The plots of the iterative function $Q_i(x_k, v_i(x_k))$ are shown in Figure 3(a).

For nonlinear system (41), the iterative Q function is monotonically non-increasing and converges to its optimum by the policy iteration based deterministic Q-learning algorithm. The corresponding

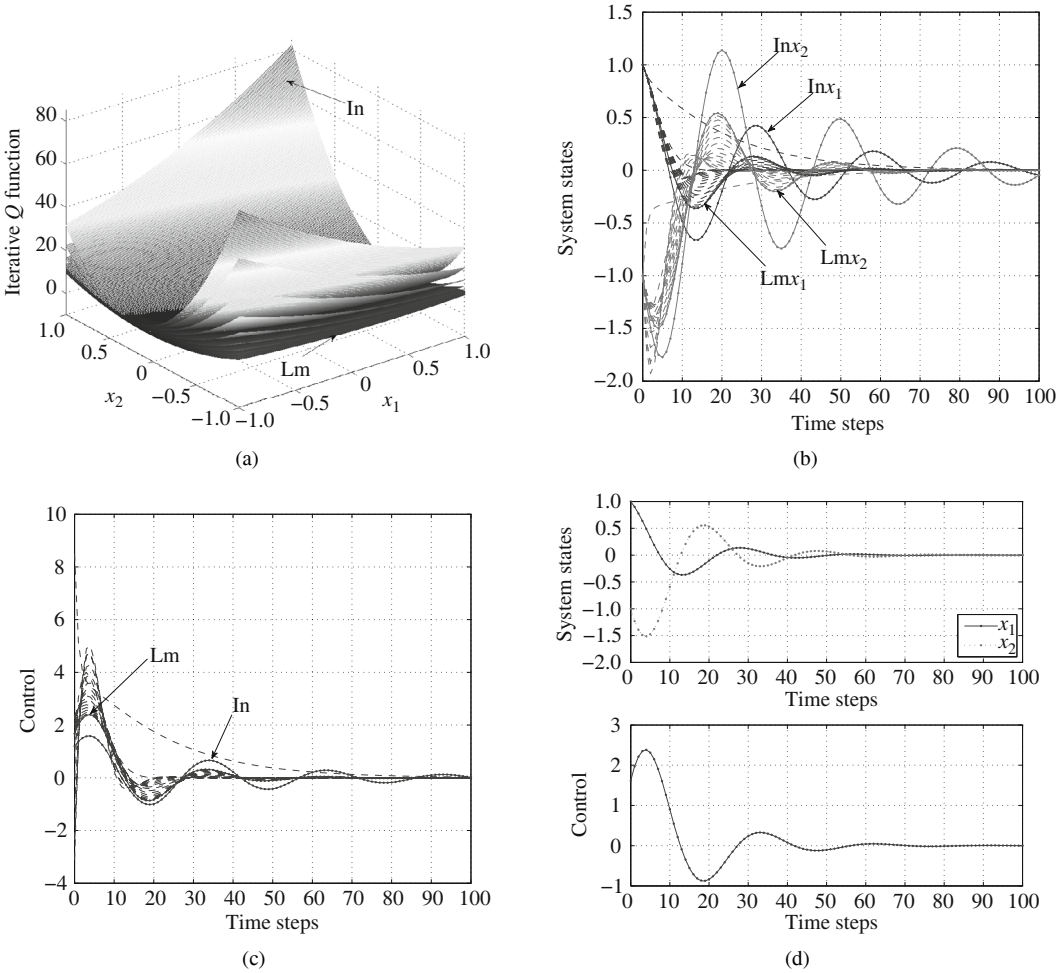

Figure 3 The policy iteration based Q-learning algorithm for nonlinear system. (a) The plots of the iterative Q function; (b) the iterative state trajectories; (c) the iterative control trajectories; (d) the optimal state and control trajectories.

iterative trajectories of system states and controls are shown in Figure 3 (b) and (c), respectively. From Figure 3 (b) and (c), we can see that the iterative system states and iterative controls are both convergent to their optimal ones. The nonlinear system (41) can be stabilized under an arbitrary iterative control law $v_i(x_k)$, where the stability properties of the developed policy iteration based deterministic Q-learning algorithm can be verified for nonlinear systems. The optimal states and control trajectories are shown in Figure 3(d).

To show the effectiveness of the developed Q-learning algorithm, value iteration algorithm [39,40,42] is used for comparisons. Implement the value iteration algorithm for 45 iteration. The plots of the iterative value iteration algorithm are shown in Figure 4(a). The corresponding iterative trajectories of system states and controls are shown in Figure 4 (b) and (c), respectively.

From Figure 4 (a)–(d), we can see that after 45 iterations, the iterative value function converges to the optimal one, where the optimal state and control in Figure 4(d) is the same as the one in Figure 3(d). For the policy iteration based deterministic Q-learning algorithm, the iterative Q function converges to its optimal within 25 iterations, while it takes 45 iterations for value iteration algorithm. It shows the effectiveness of the developed Q-learning algorithm. More importantly, from Figure 4 (b) and (c), we can see that the stability property of system (41) cannot be guaranteed under the iterative control law $u_i(x_k)$ by the value iteration algorithm. On the other hand, from Figure 3 (b) and (c), we can see that system(41) is stable under any of the iterative control law $v_i(x_k)$ by the policy iteration based deterministic Q-learning algorithm. Therefore, according to the simulation comparisons, the effectiveness of the developed policy iteration based deterministic Q-learning algorithm can be justified.

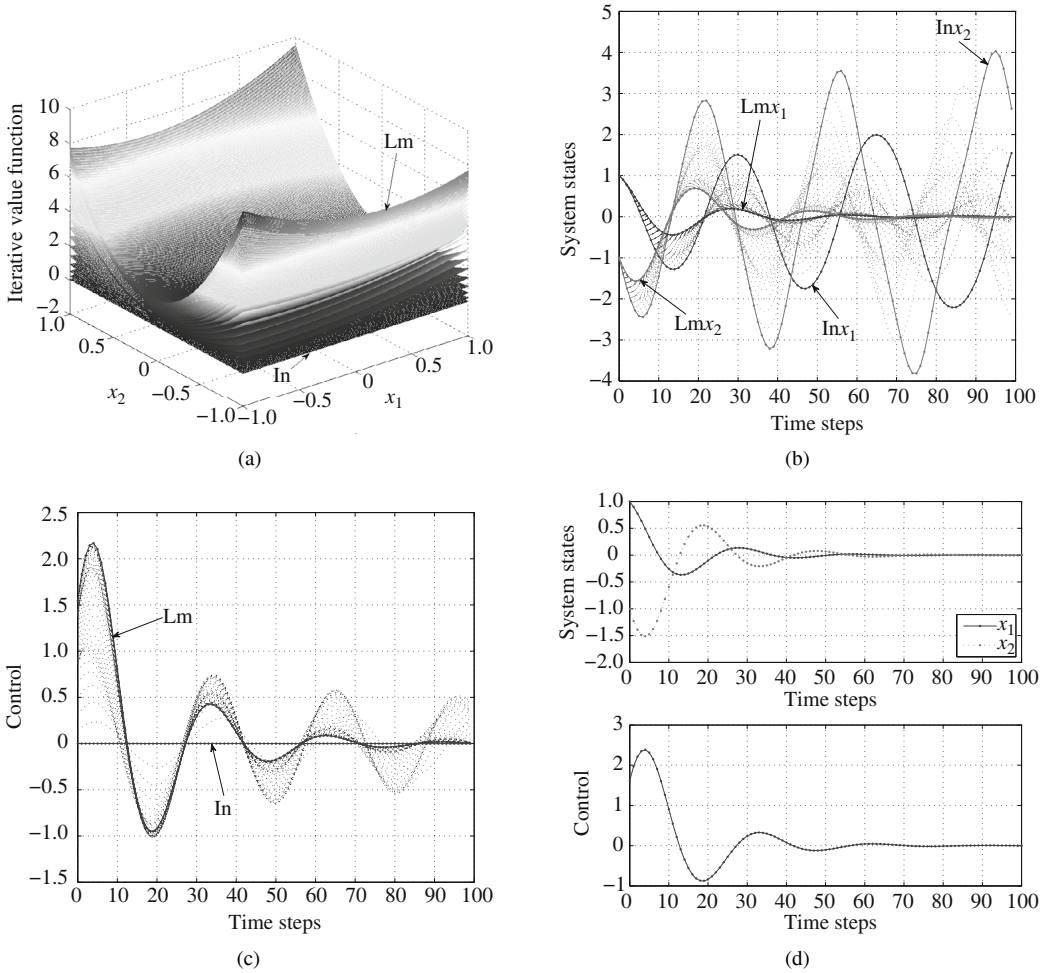

Figure 4 The value iteration algorithm for nonlinear system. (a) The plots of the iterative Q function; (b) the iterative state trajectories; (c) the iterative control trajectories; (d) the optimal state and control trajectories.

6 Conclusion and future work

In this paper, a novel policy iteration based deterministic Q-learning algorithm is developed to solve the optimal control problems for discrete-time nonlinear systems. Initialized by an arbitrary admissible control law, it has been proven that the iterative Q function and iterative control law will converge to their optimum as $i \to \infty$. Stability properties are presented to show that any of the iterative control laws can stabilize the nonlinear system.

Applications of the developed Q-learning algorithm to real engineering problems, such as smart grid or other complex systems, are very important. In [55–57], the Q-learning algorithm applications to smart grid were developed, while the stability of the smart grid system was not analyzed. In [54], the value iteration based Q-learning algorithm was proposed for the optimal energy management in smart residential environments. Although the convergence analysis of the value iteration based Q-learning algorithm in [54] was proposed, the stability of the system under the iterative control law can not be guaranteed. With the present development, we see the possibility of analyzing the stability of these systems. This will be one of our main future research topics.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61374105, 61233001, 61273140), in part by Beijing Natural Science Foundation (Grant No. 4132078).

References

1 Mohler R R, Kolodziej W J. Optimal control of a class of nonlinear stochastic systems. IEEE Trans Automat Contr, 1981, 26: 1048–1054
2 Liu C, Atkeson C G, Su J. Neighboring optimal control for periodic tasks for systems with discontinuous dynamics. Sci China Inf Sci, 2011, 54: 653–663
3 Wang J, Wang T, Yao C, et al. Active tension optimal control for WT wheelchair robot by using a novel control law for holonomic or nonholonomic systems. Sci China Inf Sci, 2014, 57: 112203
4 Liu Z, Wang Y, Li H. Two kinds of optimal controls for probabilistic mix-valued logical dynamic networks. Sci China Inf Sci, 2014, 57: 052201
5 Li X, Wang H, Ding B, et al. MABP: an optimal resource allocation approach in data center networks. Sci China Inf Sci, 2014, 57: 102801
6 Yu H, Tang W, Li S. Joint optimal sensing time and power allocation for multi-channel cognitive radio networks considering sensing-channel selection. Sci China Inf Sci, 2014, 57: 042313
7 Werbos P J. Advanced forecasting methods for global crisis warning and models of intelligence. General Systems Yearbook, 1977, 22: 25–38
8 Werbos P J. A Menu of Designs for Reinforcement Learning Over Time, in Neural Networks for Control. Massachusetts: MIT Press, 1991. 67–95
9 Modares H, Lewis F L. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. Automatica, 2014, 50: 1780–1792
10 Zhang H, Wei Q, Liu D. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica, 2011, 47: 207–214
11 Kumar M, Rajagopal K, Balakrishnan S N, et al. Reinforcement learning based controller synthesis for flexible aircraft wings. IEEE/CAA J Automat Sin, 2014, 1: 435–448
12 Kamalapurkar R, Klotz J R, Dixon W E. Concurrent learning-based approximate feedback-Nash equilibrium solution of N-player nonzero-sum differential games. IEEE/CAA J Automat Sin, 2014, 1: 239–247
13 Zhang Z, Zhao D. Clique-based cooperative multiagent reinforcement learning using factor graphs. IEEE/CAA J Automat Sin, 2015, 1: 248–256
14 Zhong X, He H, Zhang H, et al. Optimal control for unknown discrete-time nonlinear Markov jump systems using adaptive dynamic programming. IEEE Trans Neural Netw Learn Syst, 2014, 25: 2141–2155
15 Wei Q, Liu D, Yang X. Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems. IEEE Trans Neural Netw Learn Syst, 2015, 26: 866–879
16 Prokhorov D V, Wunsch D C. Adaptive critic designs. IEEE Trans Neural Networks, 1997, 8: 997–1007
17 Wei Q, Liu D. Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. IEEE Trans Autom Sci Eng, 2014, 11: 1020–1036
18 Song R, Xiao W, Sun C. A new self-learning optimal control laws for a class of discrete-time nonlinear systems based on ESN architecture. Sci China Inf Sci, 2014, 57: 068202
19 Wei Q, Wang F, Liu D, et al. Finite-approximation-error based discrete-time iterative adaptive dynamic programming. IEEE Trans Cybern, 2014, 44: 2820–2833
20 Ni Z, He B, Zhong X, Prokhorov D V. Model-free dual heuristic dynamic programming. IEEE Trans Neural Netw Learn Syst, 2015, 26: 1834–1839
21 Molina D, Venayagamoothy G K, Liang J, et al. Intelligent local area signals based damping of power system oscillations using virtual generators and approximate dynamic programming. IEEE Trans Smart Grid, 2013, 4: 498–508
22 Bertsekas D P, Tsitsiklis J N. Neuro-Dynamic Programming. Belmont: Athena Scientific, 1996
23 Si J, Wang Y T. On-line learning control by association and reinforcement. IEEE Trans Neural Networks, 2001, 12: 264–276
24 Wei Q, Liu D. Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming. IEEE Trans Ind Electron, 2014, 61: 6399–6408
25 Dierks T, Jagannathan S. Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update. IEEE Trans Neural Networks, 2012, 23: 1118–1129
26 Dierks T, Thumati B, Jagannathan S. Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. Neural Networks, 2009, 22: 851–860
27 Wei Q, Liu D. An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. Neural Networks, 2012, 32: 236–244
28 Wei Q, Song R, Yan P. Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP. IEEE Trans Neural Netw Learn Syst, 2015, PP: 1
29 Lewis F L, Vrabie D, Vamvoudakis K G. Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. IEEE Contr Syst, 2012, 32: 76–105
30 Modares H, Lewis F L, Naghibi-Sistani M B. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. IEEE Trans Neural Netw Learn Syst, 2013, 24: 1513–1525

31 Wei Q, Liu D, Y Xu. Policy iteration optimal tracking control for chaotic systems by adaptive dynamic programming approach. Chin Phys B, 2015, 24: 030502

32 Modares H, Lewis F L, Naghibi-Sistani M B. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. Automatica, 2014, 50: 193–202

33 Murray J J, Cox C J, Lendaris G G, et al. Adaptive dynamic programming. IEEE Trans Syst Man Cybern Part C-Appl Rev, 2002, 32: 140–153

34 Vamvoudakis K G, Lewis F L. Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations. Automatica, 2011, 47: 1556–1569

35 Liu D, Wei Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Trans Neural Netw Learn Syst, 2014, 25: 621–634

36 Song R, Xiao W, Zhang H, et al. Adaptive dynamic programming for a class of complex-valued nonlinear systems. IEEE Trans Neural Netw Learn Syst, 2014, 25: 1733–1739

37 Song R, Lewis F L, Wei Q, et al. Multiple Actor-critic structures for continuous-time optimal control using input-output data. IEEE Trans Neural Netw Learn Syst, 2015, 26: 851–865

38 Song R, Lewis F L, Wei Q, et al. Off-policy actor-critic structure for optimal control of unknown systems with disturbances. IEEE Trans Cybern, 2015, PP: 1

39 Al-Tamimi A, Lewis F L, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans Syst Man Cybern Part B-Cybern, 2008, 38: 943–949

40 Lincoln B, Rantzer A. Relaxing dynamic programming. IEEE Trans Automat Contr, 2006, 51: 1249–1260

41 Wei Q, Wang D, Zhang D. Dual iterative adaptive dynamic programming for a class of discrete-time nonlinear systems with time-delays. Neural Comput Appl, 2013, 23: 1851–1863

42 Zhang H, Wei Q, Luo Y. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE Trans Syst Man Cybern Part B-Cybern, 2008, 38: 937–942

43 Wei Q, Liu D. Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors. Neurocomputing, 2015, 149: 106–115

44 Wei Q, Liu D. Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear systems. Neural Comput Appl, 2014, 24: 1355–1367

45 Wei Q, Liu D. Numerically adaptive learning control scheme for discrete-time nonlinear systems. IET Control Theory Appl, 2013, 7: 1472–1486

46 Kiumarsi B, Lewis F L, Modares H, et al. Reinforcement image-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica, 2014, 50: 1167–1175

47 Liu D, Wei Q. Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. IEEE Trans Cybern, 2013, 43: 779–789

48 Wei Q, Liu D. A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems. IEEE Trans Autom Sci Eng, 2014, 11: 1176–1190

49 Wei Q, Liu D, Shi G, et al. Optimal multi-battery coordination control for home energy management systems via distributed iterative adaptive dynamic programming. IEEE Trans Ind Electron, 2015, 42: 4203–4214

50 Wei Q, Liu D. Nonlinear neuro-optimal tracking control via stable iterative Q-learning algorithm. Neurocomputing, 2015, 168: 520–528

51 Watkins C. Learning from delayed rewards. Dissertation for the Doctoral Degree. Cambridge: Cambridge University, 1989

52 Watkins C, Danyan P. Q-learning. Mach Learn, 1992, 8: 279–292

53 Busoniu L, Babuska R, Schutter B D, et al. Reinforcement Learning and Dynamic Programming Using Function Approximators. Boca Raton: CRC Press, 2010

54 Wei Q, Liu D, Shi G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments. IEEE Trans Ind Electron, 2015, 62: 2509–2518

55 Huang T, Liu D. A self-learning scheme for residential energy system control and management. Neural Comput Appl, 2013, 22: 259–269

56 Boaro M, Fuselli D, Aagelis F D, et al. Adaptive dynamic programming algorithm for renewable energy scheduling and battery management. Cognitive Comput, 2013, 5: 264–277

57 Fuselli D, Angelis F D, Boaro M, et al. Action dependent heuristic dynamic programming for home energy resource scheduling. Int J Elec Power Energ Syst, 2013, 48: 148–160

58 Prashanth L A, Bhatnagar S. Reinforcement learning with function approximation for traffic signal control. IEEE Trans Intell Transp Syst, 2011, 12: 412–421

59 Dorf R C, Bishop R H. Modern Control Systems. 12th ed. New York: Prentice Hall, 2011