Cogn Comput (2015) 7:763–771
DOI 10.1007/s12559-015-9350-z

Convergence Proof of Approximate Policy Iteration for Undiscounted Optimal Control of Discrete-Time Systems

Yuanheng Zhu[1] · Dongbin Zhao[1] · Haibo He[2] · Junhong Ji[3]

Received: 8 April 2015 / Accepted: 4 August 2015 / Published online: 25 August 2015

Abstract Approximate policy iteration (API) is studied to solve undiscounted optimal control problems in this paper. A discrete-time system with the continuous-state space and the finite-action set is considered. As approximation technique is used for the continuous-state space, approximation errors exist in the calculation and disturb the convergence of the original policy iteration. In our research, we analyze and prove the convergence of API for undiscounted optimal control. We use an iterative method to implement approximate policy evaluation and demonstrate that the error between approximate and exact value functions is bounded. Then, with the finite-action set, the greedy policy in policy improvement is generated directly. Our main theorem proves that if a sufficiently accurate approximator is used, API converges to the optimal policy. For implementation, we introduce a fuzzy approximator and verify the performance on the puddle world problem.

✉ Dongbin Zhao
dongbin.zhao@gmail.com

Yuanheng Zhu
yuanheng.zhu@gmail.com

Haibo He
he@ele.uri.edu

Junhong Ji
junhong.ji@hit.edu.cn

[1] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[2] Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA

[3] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

Keywords Approximate policy iteration · Approximation error · Optimal control · Fuzzy approximator

Introduction

Currently, a lot of complex control technique has been proposed to solve some difficult tasks, such as mobile robots [16, 25], complex network control [14], Markov chains [15], and so on [29, 33]. When considering optimal control, the target is to produce the optimal control policies according to the optimal value functions, which are mostly unknown beforehand. Reinforcement learning (RL) [9, 17, 22, 26, 32] and adaptive dynamic programming (ADP) [7, 30, 31] are among the most efficient methods to obtain the optimal performance. Their calculation of the optimal value functions can be seen as a cognitive process which is established through interacting with the system dynamics and combining with the long-term rewards.

Policy iteration (PI) [3, 13, 28] as an approach of RL and ADP has been developed and become an efficient way to solve optimal control problems. Generally, PI includes a two-step iteration: policy evaluation and policy improvement. The value function of a policy is computed in the first step, and a greedy policy is extracted in the second step. Another similar iterative method is value iteration (VI) [4, 12, 34], and detailed comparisons between PI and VI are available in the literatures [9, 17].

Many works have studied the convergence of PI. For finite Markov decision problems (MDPs), Bertsekas and Tsitsiklis [6] proved that the solution of PI converged to the optimal policy through analyzing the monotonicity of the value function sequence. Other researchers [1, 2] considered continuous-time systems and studied the optimal problem with a saturating controller using PI. They analyzed the

convergence and gave a rigorous proof. Liu and Wei [19] studied discrete-time systems and obtained similar theorems.

However, few of the above works considered the impact of approximation errors. When the state space is large-scale or continuous, approximators have to be used to approach value functions and policies. In this way, approximation errors inevitably occur in PI and disturb the convergence. So the convergence of approximate policy iteration (API) needs to be reconsidered.

For discounted optimal control, Bertsekas and Tsitsiklis [6] gave a generic error analysis between the result of API and the optimal solution. They proved that the error was related to the approximation error and the discounted factor. Munos [21] obtained a stricter bound. However, when optimal problems are undiscounted, their analysis cannot be applied as the iterative calculation is no longer a contraction. Liu and Wei [18] developed a novel numerically adaptive learning control scheme based on ADP and gave rigorous analysis of convergence and stability. However, the convergence of API for undiscounted problems has not been studied.

Here, a discrete-time system with a continuous-state space and a finite-action set is considered, and API is studied for undiscounted optimal control. First, in approximate policy evaluation, it is demonstrated that if the approximator satisfies certain conditions, errors of approximate value functions are bounded. Then, the corresponding greedy policy is the same policy of the exact value function. The convergence theorem is concluded. Our contribution emphasizes that it is the first time to prove the convergence of API for undiscounted optimal control. To verify our results, we use a fuzzy approximator in the implementation and apply to the puddle world problem to observe the performance.

The whole paper is organized as follows. First, brief introductions of PI and API are given in Sects. "Policy Iteration" and "Approximate Policy Iteration." The theoretical analysis is presented in Sect. "Convergence of Approximate Policy Iteration." A fuzzy approximator is combined with API, and an example is simulated in Sect. "A Fuzzy Implementation and an Example". In the end are our discussion and conclusion.

Policy Iteration

Consider a deterministic discrete-time system with a compact continuous-state space $\Omega \subseteq \mathbb{R}^n$ and a finite-action set $U = \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m\}$. We use the following function to specify its system dynamics

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where $x_k, x_{k+1} \in \Omega$ and $u_k \in U$. k is the time index. Suppose zero point is an equilibrium, i.e., $f(0,0) = 0$. Given a policy $h : \Omega \to U$, its value function is defined by

$$V^h(x_k) = \sum_{t=k}^{\infty} r(x_t, u_t) \Big| u_t = h(x_t)$$

where r is a positive definite penalty or cost function. The above equation can be transformed to a backward form

$$V^h(x_k) = r(x_k, h(x_k)) + V^h(f(x_k, h(x_k))). \tag{2}$$

The undiscounted optimal control is to find a policy that achieves the minimum value function for all $x_k \in \Omega$

$$V^*(x_k) = \min_h V^h(x_k)$$

and the corresponding policy h^* is called the optimal policy.

As the value function is defined in the infinite horizon manner and by the undiscounted factor, an admissible definition is needed.

Definition 1 (*Admissible*) [1, 19] A control policy h is defined to be admissible w.r.t. (1) on Ω, if $h(0) = 0$, h stabilizes (1) on Ω, and $\forall x_k \in \Omega$, $V^h(x_k)$ is finite.

As it is mentioned, PI includes a two-step iteration. Given an initial admissible policy $h^{(0)}$, its corresponding value function $V^{h^{(0)}}$ is calculated based on (2) (policy evaluation). Then a greedy policy (policy improvement) is computed using

$$h^{(i+1)}(x_k) = \arg\min_{u_k \in U} \left[r(x_k, u_k) + V^{h^{(i)}}(f(x_k, u_k)) \right]. \tag{3}$$

With the new policy $h^{(1)}$, the calculation starts again and the process keeps iterating. Liu and Wei [19] have proved that the new policy has a better performance than the previous one, i.e., $V^{h^{(i+1)}} \leq V^{h^{(i)}}$, and finally converges to the optimal solution, i.e., $h^{(i)} \to h^*$, $V^{h^{(i)}} \to V^*$ as $i \to \infty$.

However, value functions and policies can be exactly approached only if the system has a small and finite state-action set. For large or continuous systems, approximators are required and PI becomes API.

Approximate Policy Iteration

Any kind of approximators can be used to approach value functions but all bring in approximation errors. For brevity, the approximation of a value function V^h is denoted by $\hat{V}^h$, and a projection operator P is defined to map target functions to approximate functions. In this way, policy evaluation in (2) turns to calculating the following equation

$$\hat{V}^h(x_k) = P\Big(r(x_k, h(x_k)) + \hat{V}^h(f(x_k, h(x_k)))\Big). \tag{4}$$

P is generally known, and a common projection is based on least-square principle. Even if P is known, it is still difficult to solve $\hat{V}^h$ directly from (4) as the equation is implicit. Based on linear parametrizations and least-squares principle, LSTD or LSPE [23] can solve it directly, but the error between $\hat{V}^h$ and V^h is hardly analyzed.

To overcome this problem, we introduce an iterative method [9] for approximate policy evaluation. Given an initial approximation $\hat{V}_0^h$ (usually $\hat{V}_0^h = 0$), implement iterative calculation as

$$\hat{V}_{j+1}^h(x_k) = P\Big(r(x_k, h(x_k)) + \hat{V}_j^h(f(x_k, h(x_k)))\Big). \tag{5}$$

All the elements on the right side of (5) are available during the iterations, so it is easier to compute (5) than to solve (4). Furthermore, it will be demonstrated in the following section that under certain conditions, the difference between $\hat{V}_j^h$ and V^h will reduce into a small bound which is related to the approximation error. The result of (5) after the iterative calculation is denoted as the approximate policy evaluation, i.e., $\hat{V}_\infty^h \to \hat{V}^h$. It is further used in policy improvement

$$\hat{h}'(x_k) = \arg\min_{u_k \in U}\Big[r(x_k, u_k) + \hat{V}^h(f(x_k, u_k))\Big].$$

As the action set U is finite and $\hat{V}^h$ is available, $\hat{h}'$ is computed exactly using enumeration to extract the minimum value. Here, we use $\hat{h}'$ to specify the greedy policy w.r.t. approximate policy evaluation $\hat{V}^h$, compared to the greedy policy h' w.r.t. exact policy evaluation V^h. Note that as the difference between $\hat{V}^h$ and V^h is caused by approximation errors, the relationship between $\hat{h}'$ and h' needs to be analyzed.

Algorithm 1 presents the whole process of API. Because of approximation errors, the convergence of the original PI is disturbed and a new analysis of API is required.

Algorithm 1 Approximate policy iteration

Input: projection operator P; admissible policy $\hat{h}^{(0)}$; threshold parameter ε_{APE}
Output: policy $\hat{h}^{(i)}$

1: **for** $i = 0, 1, 2, \cdots$ **do**
2: initialize approximation $\hat{V}_0^{\hat{h}^{(i)}} = 0$
3: **for** $j = 0, 1, 2, \cdots$ **do**
4: $\hat{V}_{j+1}^{\hat{h}^{(i)}}(x_k) = P\left(r(x_k, \hat{h}^{(i)}(x_k)) + \hat{V}_j^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\right)$
5: **end for** $\left|\hat{V}_{j+1}^{\hat{h}^{(i)}} - \hat{V}_j^{\hat{h}^{(i)}}\right| \le \varepsilon_{APE}$
6: $\hat{V}_{j+1}^{\hat{h}^{(i)}} \to \hat{V}^{\hat{h}^{(i)}}$
7: $\hat{h}^{(i+1)}(x_k) = \arg\min_{u_k \in U}\left[r(x_k, u_k) + \hat{V}^{\hat{h}^{(i)}}(f(x_k, u_k))\right]$
8: **end for** $\hat{h}^{(i+1)} = \hat{h}^{(i)}$

Convergence of Approximate Policy Iteration

Now, let us study the convergence of API. First, consider the i-th iteration of API and use the notations in Table 1 for our analysis. The following assumptions are required.

Assumption 1 $\forall x_k \in \Omega$, given an arbitrary admissible policy h, we have $r(x_k, h(x_k)) \ge \delta V^h(x_k)$ and $r(x_k, h(x_k)) \ge \gamma V^h(f(x_k, h(x_k)))$, where $\delta > 0$ and $\gamma > 0$.

Assumption 2 Given a function g and its approximation $\hat{g} = P(g)$, the approximation error ε ($|\hat{g} - g| \le \varepsilon$) can be rewritten using a parameter σ in the form $(1-\sigma)g \le \hat{g} \le (1+\sigma)g$ with $0 < \sigma < 1$.

The first assumption is about the system, and the second one is about the approximation. Assumption 1 defines some relations between reward and value functions. Assumption 2 transforms the approximation error into a proportional version which will benefit our next analysis. With both sides of $|\hat{g} - g| \le \varepsilon$ being divided by $|g|$, we have $|\hat{g}/g - 1| \le \varepsilon/|g|$. Defining $\sigma = \varepsilon/|g|$, the form of Assumption 2 is obtained. Similar assumptions were used by Liu and Wei [18] to prove the convergence of their ADP method.

Theorem 1 At the i-th iteration of API, admissible policy $\hat{h}^{(i)}$, exact value function $V^{\hat{h}^{(i)}}$ and approximate value functions $\hat{V}_0^{\hat{h}^{(i)}}, \hat{V}_1^{\hat{h}^{(i)}}, \ldots, \hat{V}_j^{\hat{h}^{(i)}}, \ldots$ are defined as above. Under Assumptions 1 and 2, from an initial approximation $\hat{V}_0^{\hat{h}^{(i)}} = 0$, $\forall x_k \in \Omega$ we have

$$\begin{aligned}(1-\sigma)&\left[\tfrac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\tfrac{1-\sigma}{1+\gamma}\right)^l + \left(\tfrac{1-\sigma}{1+\gamma}\right)^{j-1}\delta\right]V^{\hat{h}^{(i)}}(x_k)\\ &\le \hat{V}_j^{\hat{h}^{(i)}}(x_k)\\ &\le (1+\sigma)\left[\tfrac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\tfrac{1+\sigma}{1+\gamma}\right)^l + \left(\tfrac{1+\sigma}{1+\gamma}\right)^{j-1}\right]V^{\hat{h}^{(i)}}(x_k).\end{aligned} \tag{6}$$

where $j = 1, 2, \ldots$.

Table 1 Notations for the i-th iteration of API

Symbol	Meaning
$\hat{h}^{(i)}$	Initial policy at the i-th iteration
$V^{\hat{h}^{(i)}}$	Exact value function of $\hat{h}^{(i)}$ from (2)
$\hat{V}_0^{\hat{h}^{(i)}}, \hat{V}_1^{\hat{h}^{(i)}}, \cdots \to \hat{V}^{\hat{h}^{(i)}}$	Approximate policy evaluation from (5)
$h^{(i+1)}$	Greedy policy from $V^{\hat{h}^{(i)}}$
$\hat{h}^{(i+1)}$	Greedy policy from $\hat{V}^{\hat{h}^{(i)}}$

Proof (6) can be proved using the inductive method. At the beginning with $\hat{V}_0^{\hat{h}^{(i)}} = 0$, compute $\hat{V}_1^{\hat{h}^{(i)}}(x_k) = P(r(x_k, \hat{h}^{(i)}(x_k)))$ by (5). Under Assumptions 1 and 2, we have

$$\begin{aligned}(1-\sigma)\delta V^{\hat{h}^{(i)}}(x_k) &\le (1-\sigma) r(x_k, \hat{h}^{(i)}(x_k)) \\ &\le \hat{V}_1^{\hat{h}^{(i)}}(x_k) \\ &\le (1+\sigma) r(x_k, \hat{h}^{(i)}(x_k)) \\ &\le (1+\sigma) V^{\hat{h}^{(i)}}(x_k).\end{aligned}$$

After j iterations, we suppose the following relationship holds

$$\begin{aligned}&(1-\sigma)\left[\tfrac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\tfrac{1-\sigma}{1+\gamma}\right)^l + \left(\tfrac{1-\sigma}{1+\gamma}\right)^{j-1}\delta\right] V^{\hat{h}^{(i)}}(x_k) \\ &\le \hat{V}_j^{\hat{h}^{(i)}}(x_k) \\ &\le (1+\sigma)\left[\tfrac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\tfrac{1+\sigma}{1+\gamma}\right)^l + \left(\tfrac{1+\sigma}{1+\gamma}\right)^{j-1}\right] V^{\hat{h}^{(i)}}(x_k).\end{aligned}$$

Then for the $(j+1)$-th iteration, it is easy to deduce (7) and (8)

$$\begin{aligned}&\hat{V}_{j+1}^{\hat{h}^{(i)}}(x_k) \\ &= P\Big(r(x_k, \hat{h}^{(i)}(x_k)) + \hat{V}_j^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big) \\ &\le (1+\sigma)\Big[r(x_k, \hat{h}^{(i)}(x_k)) + \hat{V}_j^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &\le (1+\sigma)\Big[r(x_k, \hat{h}^{(i)}(x_k)) + (1+\sigma)\mathbb{A} V^{\hat{h}^{(i)}} \\ &\Big(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &\le (1+\sigma) \quad \left\{\left[1 + \frac{1}{1+\gamma}[(1+\sigma)\mathbb{A} - 1]\right] r(x_k, \hat{h}^{(i)}(x_k)) \right. \\ &\qquad + \left[(1+\sigma)\mathbb{A} - \frac{\gamma}{1+\gamma}[(1+\sigma)\mathbb{A} - 1]\right] \\ &\qquad \left. \times V^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\right\} \\ &= (1+\sigma)\left(\frac{\gamma}{1+\gamma} + \frac{1+\sigma}{1+\gamma}\mathbb{A}\right)\Big[r(x_k, \hat{h}^{(i)}(x_k)) \\ &\quad + V^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &= (1+\sigma)\left[\frac{\gamma}{1+\gamma}\sum_{l=0}^{j-1}\left(\frac{1+\sigma}{1+\gamma}\right)^l + \left(\frac{1+\sigma}{1+\gamma}\right)^j\right] V^{\hat{h}^{(i)}}(x_k). \\ &\left(\mathbb{A} = \left[\frac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\frac{1+\sigma}{1+\gamma}\right)^l + \left(\frac{1+\sigma}{1+\gamma}\right)^{j-1}\right]\right)\end{aligned} \tag{7}$$

$$\begin{aligned}&\hat{V}_{j+1}^{\hat{h}^{(i)}}(x_k) \\ &= P\Big(r(x_k, \hat{h}^{(i)}(x_k)) + \hat{V}_j^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big) \\ &\ge (1-\sigma)\Big[r(x_k, \hat{h}^{(i)}(x_k)) + \hat{V}_j^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &\ge (1-\sigma)\Big[r(x_k, \hat{h}^{(i)}(x_k)) + (1-\sigma)\mathbb{B} V^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &\ge (1-\sigma) \quad \left\{\left[1 - \frac{1}{1+\gamma}[1-(1-\sigma)\mathbb{B}]\right] r(x_k, \hat{h}^{(i)}(x_k)) \right. \\ &\qquad + \left[(1-\sigma)\mathbb{B} + \frac{\gamma}{1+\gamma}[1-(1-\sigma)\mathbb{B}]\right] \\ &\qquad \left. \times V^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\right\} \\ &= (1-\sigma)\left(\frac{\gamma}{1+\gamma} + \frac{1-\sigma}{1+\gamma}\mathbb{B}\right)\Big[r(x_k, \hat{h}^{(i)}(x_k)) \\ &\quad + V^{\hat{h}^{(i)}}(f(x_k, \hat{h}^{(i)}(x_k)))\Big] \\ &= (1-\sigma)\left[\frac{\gamma}{1+\gamma}\sum_{l=0}^{j-1}\left(\frac{1-\sigma}{1+\gamma}\right)^l + \left(\frac{1-\sigma}{1+\gamma}\right)^j\delta\right] V^{\hat{h}^{(i)}}(x_k). \\ &\left(\mathbb{B} = \left[\frac{\gamma}{1+\gamma}\sum_{l=0}^{j-2}\left(\frac{1-\sigma}{1+\gamma}\right)^l + \left(\frac{1-\sigma}{1+\gamma}\right)^{j-1}\delta\right]\right)\end{aligned} \tag{8}$$

where the first inequations come from Assumption 2 and the second ones are based on the above premise about $\hat{V}_j^{\hat{h}^{(i)}}$ and $V^{\hat{h}^{(i)}}$. In the third inequations, Assumption 1 is utilized. It is obvious that the forms of (7) and (8) are the same as the premise but with $(j+1)$.

By the induction, the proof is complete. □

Corollary 1 Hold the results of Theorem 1 and suppose approximation error σ satisfies $0 < \sigma < \gamma$. Then $\forall x_k \in \Omega$,

$$\begin{aligned}\left[1 - \frac{\sigma(1+\gamma)}{\gamma+\sigma}\right] V^{\hat{h}^{(i)}}(x_k) &\le \lim_{j\to\infty} \hat{V}_j^{\hat{h}^{(i)}}(x_k) \\ &= \hat{V}^{\hat{h}^{(i)}}(x_k) \\ &\le \left[1 + \frac{\sigma(1+\gamma)}{\gamma-\sigma}\right] V^{\hat{h}^{(i)}}(x_k).\end{aligned} \tag{9}$$

Proof (9) can be proved directly from (6) under the condition $0 < \sigma < \gamma$. □

From the conclusions of Theorem 1 and Corollary 1, it is demonstrated that if approximation error σ satisfies $0 < \sigma < \gamma$, the result of approximate policy evaluation $(\hat{V}^{\hat{h}^{(i)}})$ is constrained around the exact policy evaluation $(V^{\hat{h}^{(i)}})$.

The smaller σ is, the more closely $\hat{V}^{\hat{h}^{(i)}}$ approaches $V^{\hat{h}^{(i)}}$. Next, a theorem about the policy improvement of API is presented. Before that, a new assumption is defined.

Assumption 3 Given an arbitrary admissible policy h and its corresponding value function V^h, the greedy policy h' w.r.t. V^h based on (3) is always ρ better than any other actions for all $x_k \in \Omega$, i.e.,

$$\begin{aligned} &r(x_k, h'(x_k)) + V^h(f(x_k, h'(x_k))) \\ &\quad \le \min_{u_k \in U \setminus \{h'(x_k)\}} \left[r(x_k, u_k) + V^h(f(x_k, u_k)) \right] - \rho, \end{aligned}$$

where ρ is a positive constant.

Assumption 3 defines the superiority of greedy actions to the others during the policy improvement. Then, the following theorem is deduced.

Theorem 2 At the i-th iteration of API, $\hat{h}^{(i)}$ denotes the given policy, $V^{\hat{h}^{(i)}}$ denotes the corresponding exact value function, and $\hat{V}^{\hat{h}^{(i)}}$ denotes the result of approximate policy evaluation under $0 < \sigma < \gamma$. Suppose $V^{\hat{h}^{(i)}}$ satisfies $|V^{\hat{h}^{(i)}}| < C$ on Ω. Under Assumptions 1–3, the greedy policy $h^{(i+1)}$ w.r.t. $V^{\hat{h}^{(i)}}$ is the same greedy policy $\hat{h}^{(i+1)}$ w.r.t. $\hat{V}^{\hat{h}^{(i)}}$ on Ω, if

$$\sigma < \min\left\{ \gamma, \frac{1}{2}\left[(1+\gamma)\left(1+\tfrac{2C}{\rho}\right) - \sqrt{(1+\gamma)^2\left(1+\tfrac{2C}{\rho}\right)^2 - 4\gamma} \right] \right\}. \tag{10}$$

That means $\forall x_k \in \Omega$

$$h^{(i+1)}(x_k) = \hat{h}^{(i+1)}(x_k).$$

Proof Let $\bar{u}$ denote arbitrary actions in action set except $h^{(i+1)}(x_k)$, i.e., $\forall \bar{u} \in U \setminus \{h^{(i+1)}(x_k)\}$. Based on Assumption 3, we have

$$\begin{aligned} &r(x_k, h^{(i+1)}(x_k)) + V^{\hat{h}^{(i)}}(f(x_k, h^{(i+1)}(x_k))) \\ &\quad \le r(x_k, \bar{u}) + V^{\hat{h}^{(i)}}(f(x_k, \bar{u})) - \rho. \end{aligned} \tag{11}$$

Substitute the result of Corollary 1 into (11),

$$\begin{aligned} &r(x_k, h^{(i+1)}(x_k)) + \frac{1}{1 + \frac{\sigma(1+\gamma)}{\gamma - \sigma}} \hat{V}^{\hat{h}^{(i)}}(f(x_k, h^{(i+1)}(x_k))) \\ &\quad \le r(x_k, \bar{u}) + \frac{1}{1 - \frac{\sigma(1+\gamma)}{\gamma + \sigma}} \hat{V}^{\hat{h}^{(i)}}(f(x_k, \bar{u})) - \rho. \end{aligned} \tag{12}$$

Rewrite (12)

$$\begin{aligned} &r(x_k, h^{(i+1)}(x_k)) + \hat{V}^{\hat{h}^{(i)}}(f(x_k, h^{(i+1)}(x_k))) \\ &\quad - r(x_k, \bar{u}) - \hat{V}^{\hat{h}^{(i)}}(f(x_k, \bar{u})) \\ &\quad \le \frac{\sigma(1+\gamma)}{\gamma(1+\sigma)} \hat{V}^{\hat{h}^{(i)}}(f(x_k, h^{(i+1)}(x_k))) \\ &\quad + \frac{\sigma(1+\gamma)}{\gamma(1-\sigma)} \hat{V}^{\hat{h}^{(i)}}(f(x_k, \bar{u})) - \rho. \end{aligned} \tag{13}$$

To have $\hat{h}^{(i+1)}(x_k) = h^{(i+1)}(x_k)$, we let the right side of (13) less than zero, i.e.,

$$\begin{aligned} &\frac{\sigma(1+\gamma)}{\gamma(1+\sigma)} \hat{V}^{\hat{h}^{(i)}}(f(x_k, h^{(i+1)}(x_k))) \\ &\quad + \frac{\sigma(1+\gamma)}{\gamma(1-\sigma)} \hat{V}^{\hat{h}^{(i)}}(f(x_k, \bar{u})) - \rho < 0. \end{aligned} \tag{14}$$

Besides, for all $x_k \in \Omega$,

$$\left| \hat{V}^{\hat{h}^{(i)}} \right| \le \left[1 + \frac{\sigma(1+\gamma)}{\gamma - \sigma} \right] \left| V^{\hat{h}^{(i)}} \right| \le \frac{\gamma(1+\sigma)}{\gamma - \sigma} C.$$

So we define the following inequation to support (14)

$$\left[\frac{\sigma(1+\gamma)}{\gamma(1+\sigma)} + \frac{\sigma(1+\gamma)}{\gamma(1-\sigma)} \right] \frac{\gamma(1+\sigma)}{\gamma - \sigma} C < \rho. \tag{15}$$

Rewrite (15)

$$\sigma^2 - (1+\gamma)(1 + \frac{2C}{\rho})\sigma + \gamma > 0$$

, and the solution is

$$\sigma < \frac{1}{2}\left[(1+\gamma)\left(1 + \frac{2C}{\rho}\right) - \sqrt{(1+\gamma)^2 (1 + \frac{2C}{\rho})^2 - 4\gamma} \right].$$

In this way, combined with $0 < \sigma < \gamma$, $h^{(i+1)}$ is greedy to $\hat{V}^{\hat{h}^{(i)}}$, namely the same as $\hat{h}^{(i+1)}$. □

After the above analysis, our main theorem is concluded.

Theorem 3 Given an initial admissible policy $\hat{h}^{(0)}$, compute the policy sequence $\{\hat{h}^{(1)}, \hat{h}^{(2)}, \cdots\}$ using API. Suppose Assumptions 1–3 hold and $\hat{h}^{(0)}$ has $|V^{\hat{h}^{(0)}}| < C$ on Ω. If the approximation error σ satisfies (10), then $\{\hat{h}^{(i)}\}$ is convergent to the optimal policy h^* on Ω.

Proof From Theorem 2, it is inferred that with $\hat{h}^{(0)}$, $\hat{h}^{(1)}$ selects the same actions as $h^{(1)}$ if σ satisfies (10). Based on the convergence analysis of Liu and Wei [19], $\hat{h}^{(1)}$ is also

an admissible policy and $V^{\hat{h}^{(1)}} < V^{\hat{h}^{(0)}} < C$. Proceeding iteratively, $\hat{h}^{(i)} = h^{(i)}$ holds for any i. As the policy sequence $\{h^{(i)}\}$ of the original PI converges to h^*, $\{\hat{h}^{(i)}\}$ of API also converges to h^* on Ω. □

Through our analysis, it is proved that API can converge to the optimal policy under some conditions. The convergence is guaranteed only if the approximation error is constrained to a small value, indicating a sufficiently accurate approximator is necessary. Besides, we use a generic form to represent the approximator and the approximation error. So the analytic results do not rely on any specific structure. Arbitrary approximators that satisfy the conditions can conclude the same convergence theorem.

A Fuzzy Implementation and an Example

Fuzzy approximator is commonly used in RL because of its quantified approximate property (e.g., [10, 11, 20]). To verify our convergence theorem, a fuzzy approximator is combined with API, more concretely, with approximate policy evaluation. An example is simulated to observe the performance.

Fuzzy-API

Here, we use the same fuzzy approximator of Busoniu et al. [8] which considered the implementation of VI. A triangular fuzzy partition is defined with the state space into N sets. Each set corresponds to a triangular membership function $\mu_l : \Omega \to \mathbb{R}$. The membership of a state x belonging to set l is equal to $\mu_l(x)$. Each triangular membership function has a core c_l and satisfies the following properties

1. $\mu_l(c_l) = 1$
2. $\sum_{l=1}^{N} \mu_l(x) = 1, \forall x \in \Omega$

Suppose the target function is F and it has the following continuity assumption.

Assumption 4 For any $x, y \in \Omega$,

$$|F(x) - F(y)| \le L_F \|x - y\|$$

where L_F is the Lipschitz continuity of F.

Given the value $F(c_l)$ at each c_l, the approximation is formulated by

$$\hat{F}(x) = P(F(x)) = \sum_{l=1}^{N} \mu_l(x) F(c_l).$$

Now, let us define a resolution which is helpful to estimate the precision of a fuzzy approximator.

Definition 2 (*Resolution*) Resolution is the largest distance between any state and its closest core,

$$\delta = \max_{x \in \Omega} \min_{l=1,\ldots,N} \|x - c_l\|.$$

Based on the above properties, for arbitrary x only the surrounding memberships have values. So the following inequation holds

$$\sum_{l=1}^{N} \mu_l(x) \|x - c_l\| \le \delta.$$

Then, about the triangular fuzzy approximator we have the following theorem.

Theorem 4 The approximation error between F and $\hat{F}$ is bounded by

$$\left|\hat{F}(x) - F(x)\right| \le L_F \delta.$$

Proof

$$\begin{aligned} \left|\hat{F}(x) - F(x)\right| &\le \textstyle\sum_{l=1}^{N} \mu_l(x) |F(c_l) - F(x)| \\ &\le \textstyle\sum_{l=1}^{N} L_F \mu_l(x) \|c_l - x\| \\ &\le L_F \delta \end{aligned}$$

□

From Theorem 4, it is revealed that the approximation error of the triangular fuzzy approximator is related to the resolution. The smaller value resolution chooses, the smaller approximation error will be. So we can design a fine fuzzy approximator in API so that the requirement in Theorem 2 is satisfied. The approximate policy evaluation with the triangular fuzzy approximator is presented in Algorithm 2.

Algorithm 2 Approximate policy evaluation of Fuzzy-API

Input: triangular membership functions $\{\mu_l\}$ and cores $\{c_l\}$; given policy $\hat{h}$; threshold parameter ε_{APE}; initial core values $\hat{V}_0^{\hat{h}}(c_l) = 0$;

Output: approximate value function $\hat{V}_{j+1}^{\hat{h}} \to \hat{V}^{\hat{h}}$

1: calculate approximation $\hat{V}_0^{\hat{h}}(x) = \sum_{l=1}^{N} \mu_l(x) \hat{V}_0^{\hat{h}}(c_l)$
2: **for** $j = 0, 1, 2, \cdots$ **do**
3: $\quad \hat{V}_{j+1}^{\hat{h}}(c_l) = r(c_l, \hat{h}(c_l)) + \hat{V}_j^{\hat{h}}(f(c_l, \hat{h}(c_l)))$
4: $\quad \hat{V}_{j+1}^{\hat{h}}(x) = \sum_{l=1}^{N} \mu_l(x) \hat{V}_{j+1}^{\hat{h}}(c_l)$
5: **end for** $\left|\hat{V}_{j+1}^{\hat{h}} - \hat{V}_j^{\hat{h}}\right| \le \varepsilon_{APE}$

Puddle World Problem

Now, we apply the Fuzzy-API to a commonly used problem–puddle world [5, 27]. Puddle world is a two-dimensional path problem [24] with the goal in the upper-right

corner and two oval puddles. Each dimension is continuous in [0, 1]. The two puddles extend with a radius 0.1 from two line segments: one from (0.2, 0.65) to (0.55, 0.65) and the other from (0.55, 0.3) to (0.55, 0.7). The state variables are the x and y coordinates, and there are four actions—up, down, right and left. At each action, the agent moves 0.05 distance. The cost is 1 for each step, plus a penalty if the agent is in any puddle, equal to 400 times the distance into the puddle (distance to the nearest edge). The goal region satisfies $x + y \geq 0.95 + 0.95$. An initial admissible policy is moving the agent directly up and right to the goal.

In Fuzzy-API, we choose the triangular membership functions with M equidistant cores for each state variable, leading to M^2 fuzzy sets. To study the impact of M, we change it from 4 to 41. A total of 100 randomly selected samples are used to evaluate the convergent policies. The performance of a policy is defined by the average of accumulated costs starting from the selected samples. In addition, if in an episode the sum of costs has exceeded a large value (here we use 10^3) before reaching the goal, we truncate the accumulated sum with the large value without further calculating.

After the simulation, the relationship between M and the convergent policies is revealed in Fig. 1. When M is small (<7), the learned policies are bad. Not all samples reach the goal because of large approximation errors. When M reaches 7, the policy is improved obviously. With M increasing continually, the performance is improved gradually and stabilizes in the end. This result is consistent with our analysis that if the approximation error is small, API is convergent to the optimal policy.

Next, we select $M = 21$ to observe the details in the implementation of Fuzzy-API. The costs of policies at different iterations are presented in Fig. 2. It is viewed in the figure that policies are improved monotonically with

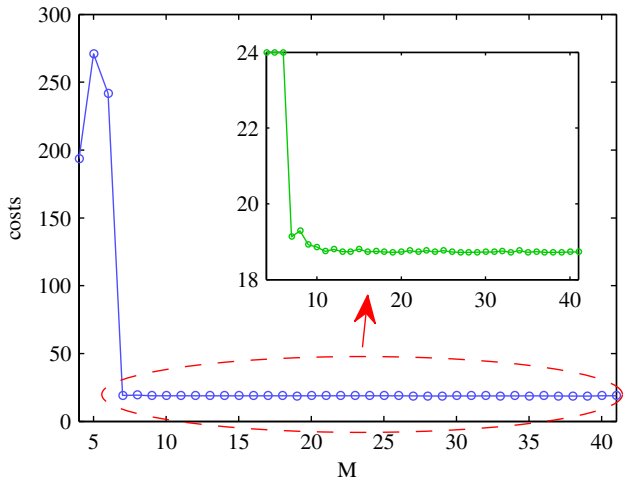

Fig. 1 Costs of convergent policies by Fuzzy-API at different M

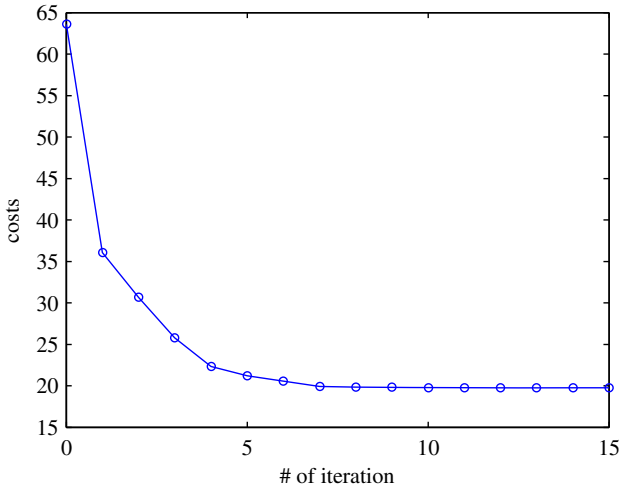

Fig. 2 Costs of policies at different iterations in Fuzzy-API when $M = 21$

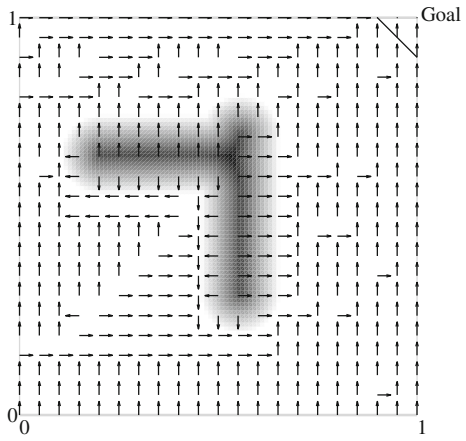

Fig. 3 The strategy of the convergent policy when $M = 21$

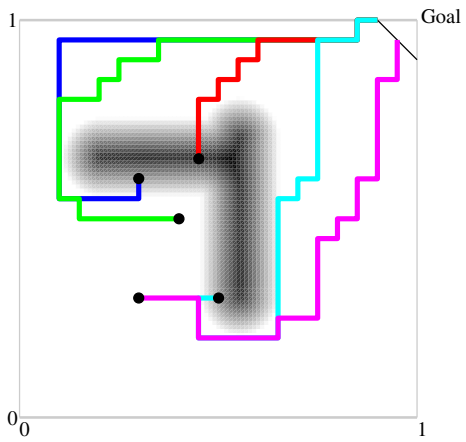

Fig. 4 Trajectories from different positions by the convergent policy when $M = 21$

the iteration increasing. After ten iterations, policies are convergent. The strategy of the convergent policy is illustrated in Fig. 3, and trajectories from different positions are depicted in Fig. 4.

Discussion and Conclusion

The convergence of API for undiscounted optimal control is proved for the first time in this paper. By the iterative method, errors in approximate policy evaluation are bounded. With the finite-action set, the same improved policy is extracted in policy improvement. The convergence theorem is concluded that API converges to the optimal policy if approximations have small approximation errors. Note that our theoretical results do not rely on any specific approximators.

For the implementation of API, we choose a triangular fuzzy structure to verify our analysis. It is demonstrated that the approximator can satisfy the requirement of API. A puddle world is simulated, and the results are consistent with our analysis.

However, we only consider the finite-action set in this paper. It avoids bringing in errors in policy improvement. But the more general case is the continuous-action system. In these systems, approximations have to be used to approach continuous policies. Due to this, additional errors occur in API in addition to the errors of approximate policy evaluation. Therefore, the analysis of API for continuous-action systems is more difficult and needs further research.

Acknowledgments This work was supported in part by National Natural Science Foundation of China (No. 61273136), State Key Laboratory of Robotics and System (SKLRS-2015-ZD-04), and National Science Foundation (NSF) under grant ECCS 1053717.

References

1. Abu-Khalaf M, Lewis FL. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica. 2005;41(5):779–91.
2. Abu-Khalaf M, Lewis F, Huang J. Policy iterations on the Hamilton–Jacobi–Isaacs equation for H_∞ state feedback control with input saturation. IEEE Trans Autom Control. 2006;51(12):1989–95.
3. Al-Tamimi A, Abu-Khalaf M, Lewis F. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. IEEE Trans Syst Man Cybern B. 2007;37(1):240–7.
4. Al-Tamimi A, Lewis F, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. IEEE Trans Syst Man Cybern B. 2008;38(4):943–9.
5. Barty K, Girardeau P, Roy JS, Strugarek C. Q-learning with continuous state spaces and finite decision set. In: Proceedings of the 2007 IEEE international symposium on approximate dynamic programming and reinforcement learning (ADPRL 2007); 2007. pp. 346–351.
6. Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Belmont, MA: Athena Scientific; 1996.
7. Boaro M, Fuselli D, Angelis F, Liu D, Wei Q, Piazza F. Adaptive dynamic programming algorithm for renewable energy scheduling and battery management. Cogn Comput. 2013;5(2):264–77.
8. Busoniu L, Ernst D, De Schutter B, Babuska R. Fuzzy approximation for convergent model-based reinforcement learning. In: Proceedings of the 2007 IEEE international conference on Fuzzy systems (FUZZ-IEEE-07), London, UK; 2007. pp. 968–973.
9. Busoniu L, Babuska R, De Schutter B, Ernst D. Reinforcement learning and dynamic programming using function approximators. New York: CRC Press; 2010.
10. Chen F, Jiang B, Tao G. Fault self-repairing flight control of a small helicopter via fuzzy feedforward and quantum control techniques. Cogn Comput. 2012;4(4):543–8.
11. Derhami V, Majd VJ, Nili Ahmadabadi M. Exploration and exploitation balance management in fuzzy reinforcement learning. Fuzzy Sets Syst. 2010;161(4):578–95.
12. Heydari A. Revisiting approximate dynamic programming and its convergence. IEEE Trans Cybern. 2014;44(12):2733–43.
13. Howard R. Dynamic programming and Markov processes. Cambridge, MA: MIT Press; 1960.
14. Hui G, Huang B, Wang Y, Meng X. Quantized control design for coupled dynamic networks with communication constraints. Cogn Comput. 2013;5(2):200–6.
15. Ikonen E, Najim K. Multiple model-based control using finite controlled markov chains. Cogn Comput. 2009;1(3):234–43.
16. Jia Z, Song Y, Cai W. Bio-inspired approach for smooth motion control of wheeled mobile robots. Cogn Comput. 2013;5(2):252–63.
17. Lewis F, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst Mag. 2009;9(3):32–50.
18. Liu D, Wei Q. Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. IEEE Trans Cybern. 2013;43(2):779–89.
19. Liu D, Wei Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Trans Neural Netw Learn Syst. 2014;25(3):621–34.
20. Meng F, Chen X. Correlation coefficients of hesitant fuzzy sets and their application based on fuzzy measures. Cogn Comput. 2015;7(4):445–63.
21. Munos R. Error bounds for approximate policy iteration. In: Proceedings of the 20th international conference on machine learning, Washington, Columbia; 2003. pp. 560–576.
22. Muse D, Wermter S. Actor-critic learning for platform-independent robot navigation. Cogn Comput. 2009;1(3):203–20.
23. Nedić A, Bertsekas DP. Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dyn Syst. 2003;13(1–2):79–110.
24. Samar R, Kamal W. Optimal path computation for autonomous aerial vehicles. Cogn Comput. 2012;4(4):515–25.
25. Song Y, Li Q, Kang Y. Conjugate unscented fastslam for autonomous mobile robots in large-scale environments. Cogn Comput. 2014;6(3):496–509.
26. Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge, MA: MIT Press; 1998.
27. Vieira D, Adeodato P, Goncalves P. A temporal difference GNG-based algorithm that can learn to control in reinforcement learning environments. In: Proceedings of the 12th international conference on machine learning and applications (ICMLA 2013), 2013; vol 1, pp. 329–332.

28. Wang D, Liu D, Li H. Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems. IEEE Trans Autom Sci Eng. 2014;11(2):627–32.
29. Wang Y, Feng G. On finite-time stability and stabilization of nonlinear port-controlled Hamiltonian systems. Sci China Inf Sci. 2013;56(10):1–14.
30. Wei Q, Liu D. A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems. IEEE Trans Autom Sci Eng. 2014;11(4):1176–90.
31. Zhang H, Liu D, Luo Y, Wang D. Adaptive dynamic programming for control: algorithms and stability. London: Springer; 2013.
32. Zhao D, Zhu Y. MEC-a near-optimal online reinforcement learning algorithm for continuous deterministic systems. IEEE Trans Neural Netw Learn Syst. 2015;26(2):346–56.
33. Zhao Y, Cheng D. On controllability and stabilizability of probabilistic Boolean control networks. Sci China Inf Sci. 2014;57(1):1–14.
34. Zhu Y, Zhao D, Liu D. Convergence analysis and application of fuzzy-HDP for nonlinear discrete-time HJB systems. Neurocomputing. 2015;149:124–31.