2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Data-Driven Iterative Adaptive Dynamic Programming Algorithm for Approximate Optimal Control of Unknown Nonlinear Systems

Hongliang Li, Derong Liu, *Fellow, IEEE*, Ding Wang, and Chao Li

***Abstract*—In this paper, we develop a data-driven iterative adaptive dynamic programming algorithm to learn offline the approximate optimal control of unknown discrete-time nonlinear systems. We do not use a model network to identify the unknown system, but utilize the available offline data to learn the approximate optimal control directly. First, the data-driven iterative adaptive dynamic programming algorithm is presented with a convergence analysis. Then, the error bounds for this algorithm are provided considering the approximation errors of function approximation structures. To implement the developed algorithm, two neural networks are used to approximate the state-action value function and the control policy. Finally, two simulation examples are given to demonstrate the effectiveness of the developed algorithm.**

I. Introduction

DYNAMIC programming [1] is a very effective method in solving the optimal control problem of nonlinear systems which relies on solving the Hamilton-Jacobi-Bellman (HJB) equation. However, it is computationally untenable for dynamic programming to obtain the optimal solution due to the well-known "curse of dimensionality" [2]. Adaptive dynamic programming (ADP) [3]–[5], also known as approximate dynamic programming [6]–[8] or neuro-dynamic programming [9], has received significantly increasing attention, which has been applied in many practical areas, such as call admission control [10], engine control [11], and energy system control [12], etc. Existing ADP approaches can be classified into several main schemes [13], [14]: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), globalized dual heuristic dynamic programming (GDHP), and their action-dependent versions, ADHDP, ADDHP, and ADGDHP. Fairbank et al. [15] presented a simple and fast calculation of the second-order gradients for GDHP. Dierks et al. [16] proposed a time-based ADP algorithm to solve the HJB equation forward-in-time without using value iteration and policy iteration.

Al-Tamimi et al. [17] proved the convergence of the value-iteration-based HDP algorithm for solving the discrete-time HJB equation. Dierks et al. [18] relaxed the need of partial knowledge of the system dynamics by online system identification. Zhang et al. [19] derived an iterative DHP algorithm to solve the approximate optimal control problem of discrete-time affine nonlinear systems with control constraints. Liu et al. [20]–[22] presented an iterative GDHP algorithm to solve the optimal control of unknown nonaffine nonlinear discrete-time systems with discount factor in the cost function. Wang et al. [23] solved the finite-horizon optimal control problem for discrete-time nonlinear systems with unspecified terminal time. Heydari and Balakrishnan [24] derived a value-iteration-based ADP algorithm to solve the fixed-final-time finite-horizon optimal control problem. In [25] and [26], a greedy HDP algorithm was presented to solve the optimal tracking control problem for a class of discrete-time nonlinear systems. Zhang et al. [27] proposed an iterative HDP algorithm to solve the optimal tracking control problem for nonlinear discrete-time systems with time delays. It should be mentioned that all the algorithms above assume that the update equations of both value function and control policy can be exactly solved at each iteration.

Leake and Liu [28] derived an inequality version of the HJB equation to derive bounds on the optimal cost function. Rantzer [29] introduced a relaxed value iteration scheme to simplify computation based on upper and lower bounds of the optimal cost function. In [30], the relaxed value iteration scheme was used to solve the optimal switching between linear systems and the optimal control of a linear system with piecewise linear cost. Liu and Wei [31] presented a convergence analysis for the approximate value iteration algorithm by using a novel expression of approximation errors.

Existing iterative ADP algorithms [17]–[27] either require the exact knowledge of the system dynamics or need a model network to identify the unknown dynamical system. In this paper, we develop a data-driven iterative ADP algorithm to learn offline the approximate optimal control of unknown discrete-time nonlinear systems. Our proposed algorithm in this paper is closely related to fitted Q iteration [32], [33]. One major difference is that we consider the undiscounted optimal control problems of nonlinear systems with continuous state space and action space. Another is that we analyze the convergence and provide the error bounds considering the approximation errors by a novel method. We do not use a model network to identify the unknown system, but utilize the available offline data to learn the approximate optimal control directly. The advantage is that it can avoid the modeling errors of the model network in the HDP and DHP. We use a model-

The authors are with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (phone: +86-10-82544761; fax: +86-10-82544799; e-mail: hongliang.li@ia.ac.cn; derong.liu@ia.ac.cn; ding.wang@ia.ac.cn; chao.li@ia.ac.cn).

This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, and 61304086.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

free ADHDP structure with two neural networks to implement the developed algorithm. Two simulation examples are given to demonstrate the effectiveness of the developed algorithm.

The remainder of this paper is organized as follows. Section II provides the problem statement of undiscounted infinite-horizon optimal control problems of discrete-time nonlinear systems. Section III presents the data-driven iterative ADP algorithm, establishes the error bounds, and gives the neural network implementation. Section IV presents two simulation examples to demonstrate the effectiveness of the developed algorithm and is followed by conclusions in Section V.

II. Problem Statement

We consider the following deterministic discrete-time nonlinear dynamical system given by

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

where $x_k \in \mathbb{R}^n$ is the system state, and $u_k \in \mathbb{R}^m$ is the control signal input. We assume that the system (1) is controllable in the sense that there exists a continuous control policy on a compact set $\Omega \subseteq \mathbb{R}^n$ that asymptotically stabilizes the system, and assume that $x_k = 0$ is an equilibrium state of the system (1). The system function $f(x_k, u_k)$ is Lipschitz continuous on Ω containing the origin, and $f(0, 0) = 0$.

Our goal is to find a state feedback control policy $u(x_k)$ which can minimize the following undiscounted infinite-horizon cost function for any initial state x_0

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

where U is a positive definite utility function, $U(0, 0) = 0$ and $U(x_k, u_k) \geq 0$, $\forall x_k$, u_k. For any admissible control policy $\mu(x)$, the map from any state x to the value of (2) is called a state value function $V^{\mu}(x)$. Then, we define the optimal state value function as

$$V^*(x) = \min_{\mu} \big\{V^{\mu}(x)\big\}.$$

According to Bellman's principle of optimality [2], the optimal state value function $V^*(x)$ satisfies the discrete-time HJB equation

$$V^*(x) = \min_{u} \big\{U(x, u) + V^*\big(f(x, u)\big)\big\}.$$

If it can be solved for V^*, the optimal control policy $\mu^*(x)$ can be obtained by

$$\mu^*(x) = \arg\min_{u} \big\{U(x, u) + V^*\big(f(x, u)\big)\big\}.$$

Similar to the state value function, the state-action value function (also known Q-function) is defined as

$$Q^{\mu}(x, u) = U(x, u) + V^{\mu}\big(f(x, u)\big). \tag{3}$$

The connection between the state value function and the state-action value function is

$$V^{\mu}(x) = Q^{\mu}\big(x, \mu(x)\big).$$

The optimal state-action value function is defined as

$$Q^*(x, u) = \min_{\mu} Q^{\mu}(x, u).$$

The optimal control policy $\mu^*(x)$ can be obtained by

$$\mu^*(x) = \arg\min_{u} Q^*(x, u). \tag{4}$$

The optimal state-action value function satisfies the following Bellman optimality equation

$$Q^*(x, u) = U(x, u) + \min_{u'} Q^*\big(f(x, u), u'\big). \tag{5}$$

The connection between the optimal state value function and the optimal state-action value function is

$$V^*(x) = \min_{u} Q^*(x, u).$$

Since the state-action value function depends on the state and action, we can develop a data-driven iterative ADP algorithm by using the state-action value function.

The contraction assumption is often required for the discounted optimal control problem. However, in the undiscounted case, we utilize the following assumption instead of the contraction assumption.

Assumption 1: There exists finite positive constant λ that makes the condition $0 \leq \min_{u'} Q^*\big(x', u'\big) \leq \lambda U(x, u)$ holds uniformly on Ω, where $x' = f(x, u)$.

For the nonlinear systems with continuous state space and action space, the optimal control problems cannot be solved exactly. Most ADP methods use function approximation structures to approximate the value function and the control policy. However, iterating on these approximate structures will inevitably give rise to approximation errors. Therefore, it is necessary to establish the error bounds considering the function approximation errors.

III. Data-Driven Iterative ADP

In this section, we first present a data-driven iterative ADP algorithm with a convergence analysis, then establish the error bounds for this algorithm considering the approximation errors, and finally give the neural network implementation.

A. Derivation of the Data-Driven Iterative ADP

To develop a data-driven iterative ADP algorithm, we use the state-action value function (3), which is different from the state value function used in the previous iterative ADP algorithms [18]–[23]. We assume that the system dynamics (1) is unknown, and only an offline data set $\{x_l, u_l, x'_l\}_N$ is available, where x'_l is the next state of x_l and u_l, and N is the number of samples in the data set. These samples may be recorded from a single trajectory or from different trajectories, and they must reflect the system sufficiently.

For the data-driven iterative ADP algorithm, it starts with any initial positive definite state-action value function Q_0 or $Q_0(\cdot, \cdot) = 0$. For $i = 1, 2, \dots$, the algorithm iterates between the control policy update

$$\mu_{i-1}(x'_l) = \arg\min_{u} Q_{i-1}(x'_l, u), \tag{6}$$

and the value function update

$$Q_i(x_l, u_l) = U\big(x_l, u_l\big) + Q_{i-1}\big(x'_l, \mu_{i-1}(x'_l)\big). \tag{7}$$

Note that i is the iteration index and l is the sample index in the data set. Combing (6) and (7), we can obtain

$$Q_i(x_l,u_l)=U(x_l,u_l)+\min_u Q_{i-1}(x_l',u). \tag{8}$$

The data-driven iterative ADP is a value iteration algorithm which can solve the optimal control problems without requiring an initial stabilizing control policy. It iterates between the control policy update (6) and the state-action value function update (7) until the iterative state-action value function converges to the optimal one. We can show that the state-action value function sequence $\{Q_i\}$ asymptotically converges to the optimal one Q^* by the following theorem.

Theorem 1: Let Assumption 1 hold. Suppose that $0 \leq \alpha Q^* \leq Q_0 \leq \beta Q^*$, $0 \leq \alpha \leq 1$ and $1 \leq \beta < \infty$. The control policy μ_i and the state-action value function Q_i are iteratively updated by (6) and (7). Then, the state-action value function sequence $\{Q_i\}$ approaches Q^* according to the inequalities

$$\left[1-\frac{1-\alpha}{(1+\lambda^{-1})^i}\right]Q^* \leq Q_i \leq \left[1+\frac{\beta-1}{(1+\lambda^{-1})^i}\right]Q^*,\ \forall i \geq 1. \tag{9}$$

Moreover, Q_i and μ_i converge to Q^* and μ^* uniformly on Ω as $i \to \infty$.

Proof: When $i=1$, according to Assumption 1, we can obtain

$$\begin{aligned}
Q_1(x_l,u_l) &= U(x_l,u_l)+\min_u Q_0(x_l',u)\\
&\geq U(x_l,u_l)+\alpha\min_u Q^*(x_l',u)\\
&\geq \left(1-\lambda\frac{1-\alpha}{\lambda+1}\right)U(x_l,u_l)\\
&\quad+\left(\alpha+\frac{1-\alpha}{\lambda+1}\right)\min_u Q^*(x_l',u)\\
&= \left(1-\frac{1-\alpha}{1+\lambda^{-1}}\right)Q^*(x_l,u_l).
\end{aligned}$$

Thus, the lower bound of Q_i holds for $i=1$. When $i=2$, according to Assumption 1, we can get

$$\begin{aligned}
Q_2(x_l,u_l) &= U(x_l,u_l)+\min_u Q_1(x_l',u)\\
&\geq U(x_l,u_l)+\left(1-\frac{1-\alpha}{1+\lambda^{-1}}\right)\min_u Q^*(x_l',u)\\
&\geq \left[1-\frac{1-\alpha}{(1+\lambda^{-1})^2}\right]U(x_l,u_l)\\
&\quad+\left[1-\frac{1-\alpha}{1+\lambda^{-1}}+\frac{\lambda(1-\alpha)}{(1+\lambda)^2}\right]\min_u Q^*(x_l',u)\\
&= \left[1-\frac{1-\alpha}{(1+\lambda^{-1})^2}\right]Q^*(x_l,u_l).
\end{aligned}$$

Thus, the lower bound of Q_i holds for $i=2$. Then, we can prove the left hand side of the inequality (9) by repeating the argument i times. The right hand side can be shown by the same way.

According to the inequalities (9), for $0<\lambda<\infty$, the state-action value function Q_i converges to Q^* uniformly as $i \to \infty$. The control policy μ_i also converges to μ^* according to (4). ■

B. Error Bounds for the Data-Driven Iterative ADP

In general, the control policy update (6) and state-action value function update (7) cannot be solved accurately. Function approximation structures like neural networks are usually used to approximate the state-action value function Q_i and the control policy μ_i. Here, we use $\hat{Q}_i$ and $\hat{\mu}_i$ to stand for the approximate expressions of Q_i and μ_i, respectively. According to (8), the approximation errors in the control policy update (6) and state-action value function update (7) are expressed as

$$\begin{aligned}
\bar{\epsilon}\big[U(x_l,u_l)+\min_u \hat{Q}_{i-1}(x_l',u)\big] &\geq \hat{Q}_i(x_l,u_l)\\
&\geq \underline{\epsilon}\big[U(x_l,u_l)+\min_u \hat{Q}_{i-1}(x_l',u)\big]
\end{aligned} \tag{10}$$

where $\bar{\epsilon} \geq 1$ and $\underline{\epsilon} \leq 1$ are finite positive constants.

Based on Assumption 1, we can establish the error bounds for the data-driven iterative ADP algorithm by the following theorem.

Theorem 2: Let Assumption 1 hold. Suppose that $0 \leq \alpha Q^* \leq Q_0 \leq \beta Q^*$, $0 \leq \alpha \leq 1$ and $1 \leq \beta < \infty$. The approximate state-action value function $\hat{Q}_i$ satisfy the iterative error condition (10). Then, the approximate state-action value function sequence $\{\hat{Q}_i\}$ approaches Q^* according to the following inequalities

$$\begin{aligned}
\underline{\epsilon}\left[1-\sum_{j=1}^{i}\frac{\underline{\epsilon}^{j-1}(1-\underline{\epsilon})}{(1+\lambda^{-1})^j}-\frac{\underline{\epsilon}^i(1-\alpha)}{(1+\lambda^{-1})^{i+1}}\right]Q^* &\leq \hat{Q}_{i+1}\\
\leq \bar{\epsilon}\left[1+\sum_{j=1}^{i}\frac{\bar{\epsilon}^{j-1}(\bar{\epsilon}-1)}{(1+\lambda^{-1})^j}+\frac{\bar{\epsilon}^i(\beta-1)}{(1+\lambda^{-1})^{i+1}}\right]Q^*, &\forall i \geq 0.
\end{aligned} \tag{11}$$

Moreover, the approximate state-action value function sequence $\{\hat{Q}_i\}$ converges to a finite neighborhood of Q^* uniformly on Ω as $i \to \infty$, i.e.,

$$\frac{\underline{\epsilon}}{1+\lambda-\underline{\epsilon}\lambda}Q^* \leq \lim_{i\to\infty}\hat{Q}_i \leq \frac{\bar{\epsilon}}{1+\lambda-\bar{\epsilon}\lambda}Q^*, \tag{12}$$

under the condition $\bar{\epsilon} < \lambda^{-1}+1$.

Proof: First, we prove the lower bound of the approximate state-action value function $\hat{Q}_{i+1}$ by mathematical induction. When $i=0$, according to Assumption 1, we can obtain

$$\begin{aligned}
\hat{Q}_1(x_l,u_l) &\geq \underline{\epsilon}\big[U(x_l,u_l)+\min_u \hat{Q}_0(x_l',u)\big]\\
&\geq \underline{\epsilon}\big[U(x_l,u_l)+\alpha\min_u Q^*(x_l',u)\big]\\
&\geq \underline{\epsilon}\left(1-\frac{1-\alpha}{1+\lambda^{-1}}\right)Q^*(x_l,u_l).
\end{aligned}$$

Thus, the lower bound of $\hat{Q}_{i+1}$ holds for $i=0$. When $i=1$,

according to Assumption 1, we can get

$$\begin{aligned}
&\hat{Q}_2(x_l,u_l)\\
&\geq \underline{\epsilon}\big[U(x_l,u_l)+\min_u \hat{Q}_1(x_l',u)\big]\\
&\geq \underline{\epsilon}\left[U(x_l,u_l)+\underline{\epsilon}\left(1-\frac{1-\alpha}{1+\lambda^{-1}}\right)\min_u Q^*(x_l',u)\right]\\
&\geq \underline{\epsilon}\left\{\left[1-\frac{1-\underline{\epsilon}}{1+\lambda^{-1}}-\frac{\underline{\epsilon}(1-\alpha)}{(1+\lambda^{-1})^2}\right]U(x_l,u_l)\right.\\
&\left.+\left[\underline{\epsilon}\left(1-\frac{1-\alpha}{1+\lambda^{-1}}\right)+\frac{1-\underline{\epsilon}}{1+\lambda}+\frac{\lambda\underline{\epsilon}(1-\alpha)}{(1+\lambda)^2}\right]\min_u Q^*(x_l',u)\right\}\\
&\geq \underline{\epsilon}\left[1-\frac{1-\underline{\epsilon}}{1+\lambda^{-1}}-\frac{\underline{\epsilon}(1-\alpha)}{(1+\lambda^{-1})^2}\right]Q^*(x_l,u_l).
\end{aligned}$$

Hence, the lower bound of $\hat{Q}_{i+1}$ holds for $i=1$. The lower bound of $\hat{Q}_{i+1}$ in (11) can be proved by repeating the argument $i+1$ times.

Also, the upper bound of $\hat{Q}_{i+1}$ can be proved similarly. Therefore, the lower and upper bounds of $\hat{Q}_{i+1}$ in (11) have been proved.

At last, we prove that the approximate state-action value function sequence $\{\hat{Q}_i\}$ converges to a finite neighborhood of Q^* uniformly on Ω as $i\to\infty$. Since the sequence $\{\underline{\epsilon}^{j-1}(1-\underline{\epsilon})/(1+\lambda^{-1})^j\}$ is a geometric series, we have

$$\sum_{j=1}^{i}\frac{\underline{\epsilon}^{j-1}(1-\underline{\epsilon})}{(1+\lambda^{-1})^j}=\frac{\frac{(1-\underline{\epsilon})}{1+\lambda^{-1}}\left(1-\left(\frac{\underline{\epsilon}}{1+\lambda^{-1}}\right)^i\right)}{1-\frac{\underline{\epsilon}}{1+\lambda^{-1}}}.$$

Considering $\underline{\epsilon}/(1+\lambda^{-1})<1$, we have

$$\lim_{i\to\infty}\hat{Q}_i\geq\frac{\underline{\epsilon}}{1+\lambda-\underline{\epsilon}\lambda}Q^*.$$

For the other part, if $\overline{\epsilon}/(1+\lambda^{-1})<1$, i.e., $\overline{\epsilon}<\lambda^{-1}+1$, we can show that

$$\lim_{i\to\infty}\hat{Q}_i\leq\frac{\overline{\epsilon}}{1+\lambda-\overline{\epsilon}\lambda}Q^*.$$

Thus, we complete the proof. ■

Remark 1: Inequalities (12) gives the suboptimality bound of the approximate optimal state-action value function. The condition $\overline{\epsilon}<1/\lambda+1$ should satisfy to make the upper bound in (12) be finite and positive. The lower bound in (12) is always positive for $\underline{\epsilon}\leq 1$. A larger λ will lead to a slower convergence rate and a larger error bound. When $\underline{\epsilon}=\overline{\epsilon}=1$, the inequalities (11) are the same as the inequalities (9), and the state-action value function sequence $\{\hat{Q}_i\}$ converges to Q^* uniformly on Ω as $i\to\infty$.

C. Neural Network Implementation for Approximate Optimal Control

We have shown that the approximate state-action value iteration can converge to a finite neighborhood of the optimal one. This makes it feasible to use neural networks as function approximation structures. We present a detailed implementation of this algorithm using neural networks in this subsection.

Our proposed data-driven iterative ADP algorithm is most relevant to the model-free ADHDP structure. The whole structure diagram is shown in Fig. 1, where the critic and action neural networks are used to approximate the state-action value function and the control policy, respectively.

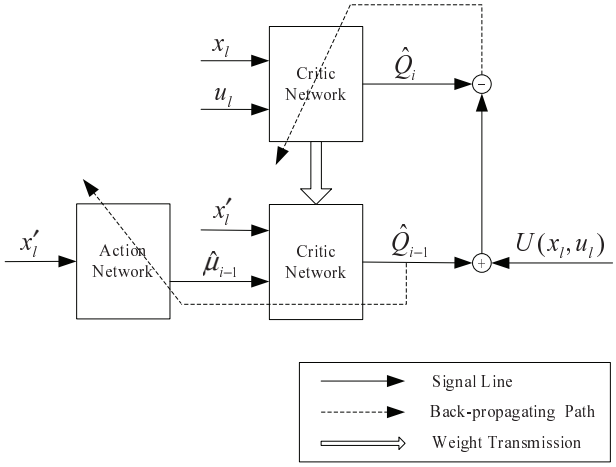

Fig. 1. Structure diagram of data-driven iterative ADP

A neural network can be used to approximate some smooth function on a prescribed compact set. The state-action value function $\hat{Q}_i(x_l,u_l)$ is expressed by the critic neural network

$$\hat{Q}_i(x_l,u_l)=\big(W_{c(i)}\big)^{\mathrm{T}}\phi\big(\big(Y_{c(i)}\big)^{\mathrm{T}}[x_l;u_l]\big). \tag{13}$$

where the activation functions are selected as tansig$(\cdot)$. The target function of the critic neural network is given by

$$\hat{Q}_i^*(x_l,u_l)=U(x_l,u_l)+\hat{Q}_{i-1}\big(x_l',\hat{\mu}_{i-1}(x_l')\big).$$

Then, the error function for training the critic neural network is defined by

$$e_{c(i)}=\hat{Q}_i(x_l,u_l)-\hat{Q}_i^*(x_l,u_l),$$

and the performance function to be minimized is defined by

$$E_{c(i)}=\frac{1}{2}(e_{c(i)})^{\mathrm{T}}e_{c(i)}. \tag{14}$$

The control policy $\hat{\mu}_{i-1}(x_l')$ is expressed by the action neural network

$$\hat{\mu}_{i-1}(x_l')=W_{a(i-1)}^{\mathrm{T}}\phi\big(Y_{a(i-1)}^{\mathrm{T}}x_l'\big). \tag{15}$$

The target function of the action neural network is defined by

$$\hat{\mu}_{i-1}^*(x_l')=\arg\min_u\hat{Q}_{i-1}(x_l',u).$$

Then, the error function for training the action neural network is given by

$$e_{a(i-1)}=\hat{\mu}_{i-1}(x_l')-\hat{\mu}_{i-1}^*(x_l').$$

The weights of the action neural network are updated to minimize the following performance function

$$E_{a(i-1)}=\frac{1}{2}(e_{a(i-1)})^{\mathrm{T}}e_{a(i-1)}. \tag{16}$$

We use the gradient descent method to tune the weights of critic and neural networks on a data set sampled from different trajectories.

A detailed process of the data-driven iterative ADP algorithm is given in Algorithm 1. It should be mentioned that Algorithm 1 runs in an offline manner.

Algorithm 1 Data-Driven Iterative ADP

Step 1. Collect samples to construct a data set $\{x_l, u_l, x_l'\}_N$. Initialize critic and action neural networks. Set the maximum number of iteration steps $i_{\max}$, and set $i = 0$.

Step 2. Set $i \leftarrow i + 1$.

Step 3. Update the control policy $\hat{\mu}_{i-1}(x_l')$ by minimizing (16) on the data set $\{x_l, u_l, x_l'\}_N$.

Step 4. Update the state-action value function $\hat{Q}_i(x_l, u_l)$ by minimizing (14) on the data set $\{x_l, u_l, x_l'\}_N$.

Step 5. Repeat Steps 2–4 until the convergence conditions are met.

Step 6. Obtain the approximate optimal control policy $\hat{\mu}_{i-1}$.

IV. Simulation Study

In this section, two simulation examples are given to demonstrate the effectiveness of the data-driven iterative ADP algorithm.

Example 1: (Discrete-Time Linear System) Consider the following discrete-time linear system $x_{k+1} = Ax_k + Bu_k$, where

$$A = \begin{bmatrix} 0 & 0.4 \\ 0.3 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{17}$$

$x_k = [x_{1k} \;\; x_{2k}]^{\mathrm{T}} \in \mathbb{R}^2$, and $u_k \in \mathbb{R}$.
Define the cost function as

$$J(x_0, u) = \sum_{k=0}^{\infty} \left(x_{1k}^2 + x_{2k}^2 + u_k^2\right).$$

The structures of the critic and action neural networks are chosen as 3–6–1 and 2–6–1, respectively. The initial weights of the action neural network are chosen randomly in $[-0.1, 0.1]$, and the initial weights of the critic neural network are all chosen as zero. The maximum number of iteration steps is selected as $i_{\max} = 10$. The data set is constructed by collecting 1000 samples from different trajectories of the system (17).

After running the algorithm for 10 iteration steps, we apply the obtained approximate optimal control policy $\hat{\mu}_9$ to the system (17) for the initial state $x_0 = [1, -1]^{\mathrm{T}}$. From the state trajectories in Fig. 2 and the control input in Fig. 3, we can find that the obtained approximate optimal control policy is quite near to the optimal one.

Example 2: (Discrete-Time Nonlinear System) Consider the following discrete-time nonlinear system $x_{k+1} = h(x_k) + g(x_k)u_k$, where

$$h(x_k) = \begin{bmatrix} 0.9x_{1k} + 0.1x_{2k} \\ -0.05(x_{1k} + x_{2k}(1 - (\cos(2x_{1k}) + 2)^2)) + x_{2k} \end{bmatrix} \tag{18}$$

$$g(x_k) = \begin{bmatrix} 0 \\ 0.1\cos(2x_{1k}) + 0.2 \end{bmatrix},$$

$x_k = [x_{1k} \;\; x_{2k}]^{\mathrm{T}} \in \mathbb{R}^2$, and $u_k \in \mathbb{R}$.
Define the cost function as

$$J(x_0, u) = \sum_{k=0}^{\infty} \left(0.1x_{1k}^2 + 0.1x_{2k}^2 + 0.1u_k^2\right).$$

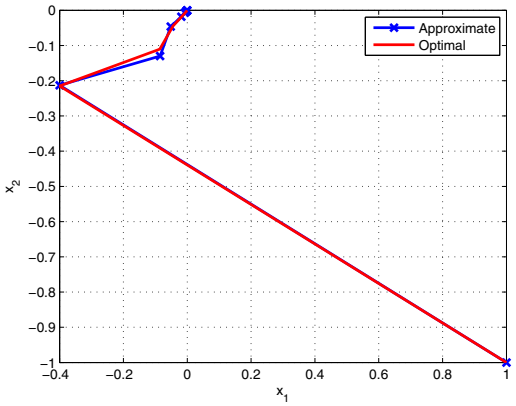

Fig. 2. The state trajectories of Example 1

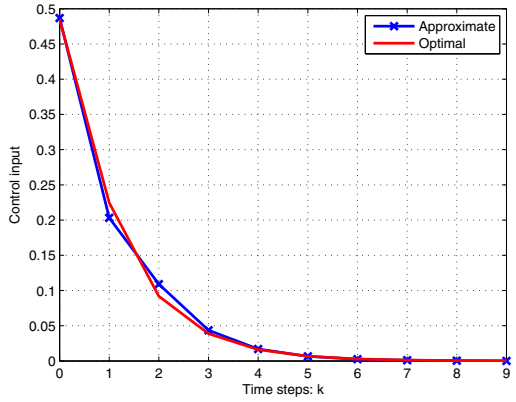

Fig. 3. The control input of Example 1

The structures and initialization methods of the critic and action neural networks are the same as those in Example 1. The maximum number of iteration steps is selected as $i_{\max} = 20$. The data set is constructed by collecting 1000 samples from different trajectories of the system (18).

The convergence curve of $\hat{Q}_i(-0.6533, 0.4715, -0.9875)$ is given in Fig. 4. It can be seen that $\hat{Q}_i$ has converged after 20 iteration steps. Then, we apply the obtained approximate optimal control policy $\hat{\mu}_{19}$ to the system (18) for 100 time steps. The state trajectories and the control inputs are displayed in Figs. 5 and 6, respectively. It is shown that the obtained control obtains very good performance.

V. Conclusions

In this paper, a data-driven iterative ADP algorithm was developed to learn the approximate optimal control by utilizing the available offline data directly. The error bounds for this algorithm were provided considering the approximation errors. Two neural networks were used to approximate the state-action value function and the control policy. The simulation examples demonstrated the effectiveness of the developed algorithm.

References

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmonth, MA: Athena Scientific, 2012.

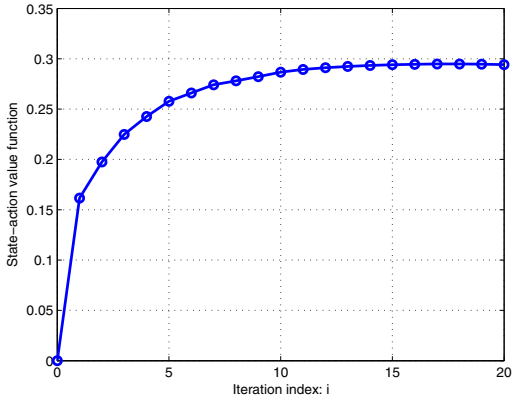

Fig. 4. The convergence curve of $\hat{Q}_i(-0.6533, 0.4715, -0.9875)$

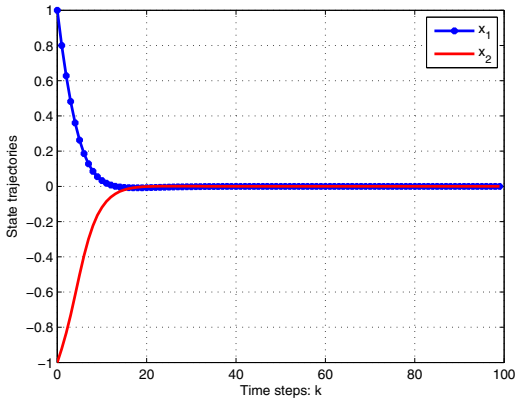

Fig. 5. The state trajectories of Example 2

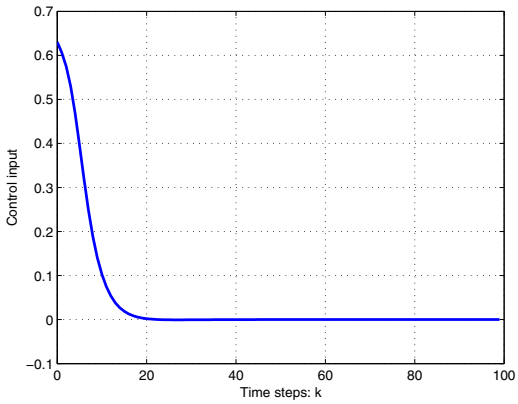

Fig. 6. The control input of Example 2

[2] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[3] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[4] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, July 2009.

[5] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[6] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, ch. 13.

[7] J. Si, A. G. Barto, W. B. Powell, and D. C. Wunsch, Eds., *Handbook of Learning and Approximate Dynamic Programming*. New York: IEEE Press/Wiley, 2004.

[8] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Hoboken, NJ: Wiley, 2013.

[9] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[10] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1219–1228, Sept. 2005.

[11] D. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air–fuel ratio control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.

[12] T. Huang and D. Liu, "A self-learning scheme for residential energy system control and management," *Neural Computing and Applications*, vol. 22, no. 2, pp. 259–269, Feb. 2013.

[13] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sept. 1997.

[14] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[15] M. Fairbank, E. Alonso, and D. Prokhorov, "Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks," *IEEE Trans. Neural Netw. and Learning Systems*, vol. 23, no. 7, pp. 1671–1676, Oct. 2012.

[16] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. and Learning Systems*, vol. 23, no. 7, pp. 1118–1129, July 2012.

[17] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[18] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Netw.*, vol. 22, no. 5–6, pp. 851–860, July–Aug. 2009.

[19] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sept. 2009.

[20] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, July 2012.

[21] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[22] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Information Sciences*, vol. 220, pp. 331–342, Jan. 2013.

[23] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear sytems with ε-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1854–1862, Dec. 2011.

[24] A. Heydari and S. N. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE*

Trans. Neural Netw. and Learn. Syst., vol. 24, no. 1, pp. 145–157, Jan. 2013.

[25] H. Zhang, Q. Wei, and Y. Luo, "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. B, Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[26] D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, Feb. 2012.

[27] H. Zhang, R. Song, Q. Wei, and T. Zhang, "Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[28] R. J. Leake and R. W. Liu, "Construction of suboptimal control sequences," *SIAM J. Control*, vol. 5, no. 1, pp. 54–63, 1967.

[29] A. Rantzer, "Relaxed dynamic programming in switching systems," *Proc. Inst. Elect. Eng*, vol. 153, no. 5, pp. 567–574, 2006.

[30] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1249–1260, Aug. 2006.

[31] D. Liu and Q. Wei, "Finite-approximation-error based optimal control approach for discrete-time nonlinear systems," *IEEE Trans. on Cybern.*, vol. 43, no. 2, pp. 779–789, Apr. 2013.

[32] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *J. Mach. Learn. Res.*, vol. 6, pp. 503–556, 2005.

[33] M. Riedmiller, "Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method," in *Proc. 16th Eur. Conf. Mach. Learn.*, Porto, Portugal, Oct. 3–7, 2005, pp. 317–328.