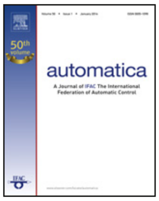
Automatica 50 (2014) 3281–3290

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica



Brief paper

Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design[☆]

Biao Luo[a], Huai-Ning Wu[b,1], Tingwen Huang[c], Derong Liu[a]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] Science and Technology on Aircraft Control Laboratory, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing 100191, PR China
[c] Texas A& M University at Qatar, PO Box 23874, Doha, Qatar

ARTICLE INFO

Article history:
Received 24 October 2013
Received in revised form
2 August 2014
Accepted 3 August 2014
Available online 25 October 2014

Keywords:
Nonlinear optimal control
Reinforcement learning
Off-policy
Data-based approximate policy iteration
Neural network
Hamilton–Jacobi–Bellman equation

ABSTRACT

This paper addresses the model-free nonlinear optimal control problem based on data by introducing the reinforcement learning (RL) technique. It is known that the nonlinear optimal control problem relies on the solution of the Hamilton–Jacobi–Bellman (HJB) equation, which is a nonlinear partial differential equation that is generally impossible to be solved analytically. Even worse, most practical systems are too complicated to establish an accurate mathematical model. To overcome these difficulties, we propose a data-based approximate policy iteration (API) method by using real system data rather than a system model. Firstly, a model-free policy iteration algorithm is derived and its convergence is proved. The implementation of the algorithm is based on the actor–critic structure, where actor and critic neural networks (NNs) are employed to approximate the control policy and cost function, respectively. To update the weights of actor and critic NNs, a least-square approach is developed based on the method of weighted residuals. The data-based API is an off-policy RL method, where the "exploration" is improved by arbitrarily sampling data on the state and input domain. Finally, we test the data-based API control design method on a simple nonlinear system, and further apply it to a rotational/translational actuator system. The simulation results demonstrate the effectiveness of the proposed method.

1. Introduction

The nonlinear optimal control problem has been widely studied in the past few decades, and a large number of theoretical results (Bertsekas, 2005; Hull, 2003; Lewis, Vrabie, & Syrmos, 2013) have been reported. However, the main bottleneck for their practical application is that the so-called Hamilton–Jacobi–Bellman (HJB) equation should be solved. The HJB equation is a first order nonlinear partial differential equation (PDE), which is difficult or impossible to solve, and may not have global analytic solutions even in simple cases. For linear systems, the HJB equation results in an algebraic Riccati equation (ARE). In 1968, Kleinman (1968) proposed a famous iterative scheme for solving the ARE, where it was converted to a sequence of linear Lyapunov matrix equations. In Saridis and Lee (1979), the thought of the iterative scheme was extended to solve the HJB equation, which was successively approximated by a series of generalized HJB (GHJB) equations that are linear PDEs. To solve the GHJB equation, Beard, Saridis, and Wen (1997) proposed a Galerkin approximation approach where a detailed convergence analysis was provided. By using a neural network (NN) for function approximation, the iterative scheme was further extended to constrained input systems (Abu-Khalaf & Lewis, 2005). In Lin, Loxton, and Teo (2014); Wang, Gui, Teo, Loxton, and Yang (2009), the control parameterization method does not require the solution of the HJB equation, and can handle state constraints, which furnishes an open-loop control rather than a feedback control. However, most of these approaches are model-based which require an accurate mathematical model of the system.

With the fast development of science technologies, many industrial systems (such as systems in aeronautics and astronautics, chemical engineering, mechanical engineering, electronics, electric power, traffic and transportation) become more and more complicated due to their large scale and complex manufacturing

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Kok Lay Teo under the direction of Editor Ian R. Petersen.

E-mail addresses: biao.luo@ia.ac.cn (B. Luo), whn@buaa.edu.cn (H.-N. Wu), tingwen.huang@qatar.tamu.edu (T. Huang), derong.liu@ia.ac.cn (D. Liu).

[1] Tel.: +86 10 82317301; fax: +86 10 82317332.

http://dx.doi.org/10.1016/j.automatica.2014.10.056
0005-1098/© 2014 Elsevier Ltd. All rights reserved.

techniques, equipment and procedures. One of the most prominent features for these systems is the presence of vast volume of data accompanied by the lack of an effective physical process model that can support control design. Moreover, the accurate modeling and identification of these systems are extremely costly or impossible to conduct. On the other hand, with the development and extensive applications of digital sensor technologies, and the availability of cheaper measurement and computing equipments, more and more system information could be extracted for direct control design. Thus, the development of data-based control approaches for practical systems is a promising, but still challenging research area.

Over the past few decades, the thought of reinforcement learning (RL) techniques has been introduced to study the optimal control problems (Al-Tamimi, Lewis, & Abu-Khalaf, 2008; Lewis & Liu, 2013; Lewis, Vrabie, & Vamvoudakis, 2012; Luo, Wu, & Huang, in press; Luo, Wu, & Li, in press, 2014; Si & Wang, 2001; Vrabie & Lewis, 2009). RL methods have the ability to find an optimal control policy in an unknown environment, which makes RL a promising method for data-based control design. For discrete-time systems, many RL based optimal control approaches have been developed, such as, heuristic dynamic programming (HDP) (Al-Tamimi et al., 2008), direct HDP (Si & Wang, 2001), dual heuristic programming (Heydari & Balakrishnan, 2013), and globalized DHP algorithm (Wang, Liu, Wei, Zhao, & Jin, 2012). For continuous-time systems, Vrabie and Lewis (2009) proposed a policy iteration algorithm to solve the nonlinear optimal control problem online along a single state trajectory. Vamvoudakis and Lewis (2010) gave an online policy iteration algorithm which tunes synchronously the weights of both actor and critic NNs for the nonlinear optimal control problem. In Liu, Wang, and Li (2014), approximate dynamic programming (ADP) was employed to design a stabilizing control strategy for a class of continuous-time nonlinear interconnected large-scale systems. But those methods are partially model-based (Vrabie & Lewis, 2009) or completely model-based (Liu et al., 2014; Vamvoudakis & Lewis, 2010). Recently, some data-based RL methods have been reported. For example, data-based policy iteration (Jiang & Jiang, 2012) and Q-learning (Lee, Park, & Choi, 2012) algorithms were developed for linear systems. Zhang, Cui, Zhang, and Luo (2011) presented a data-driven robust approximate optimal tracking control scheme for nonlinear systems, but it requires a prior model identification procedure and the ADP method is still model-based. Till present, the development of model-free RL methods and theories for nonlinear continuous-time optimal control problem remains an open issue, which motivates the present study.

In this paper, we consider the optimal control problem of continuous-time nonlinear systems with completely unknown model, and develop a model-free approximate policy iteration (API) method for learning the optimal control policy from real system data. The rest of the paper is arranged as follows. The problem description and some preliminary results are presented in Sections 2 and 3. A data-based API method is developed in Section 4 and its effectiveness is tested in Section 5. Finally, a brief conclusion is given in Section 6.

Notation. $\mathbb{R}, \mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ are the set of real numbers, the n-dimensional Euclidean space and the set of all real matrices, respectively. $\|\cdot\|$ denotes the vector norm or matrix norm in $\mathbb{R}^n$ or $\mathbb{R}^{n\times m}$, respectively. The superscript T is used for the transpose and I denotes the identify matrix of appropriate dimension. $\bigtriangledown \triangleq \partial/\partial x$ denotes a gradient operator notation. For a symmetric matrix $M, M > (\geq) 0$ means that it is a positive (semi-positive) definite matrix. $\|v\|_M^2 \triangleq v^T M v$ for some real vector v and symmetric matrix $M > (\geq) 0$ with appropriate dimensions. $C^1(\mathcal{X})$ is a function space on $\mathcal{X}$ with first derivatives are continuous. Let $\mathcal{X}$ and $\mathcal{U}$ be compact sets, denote $\mathcal{D} \triangleq \{(x, u, x') | x, x' \in \mathcal{X}, u \in \mathcal{U}\}$. For column vector functions $s_1(x, u, x')$ and $s_2(x, u, x')$, where $(x, u, x') \in \mathcal{D}$ define the inner product $\langle s_1(x, u, x'), s_2(x, u, x')\rangle_{\mathcal{D}} \triangleq \int_{\mathcal{D}} s_1^T(x, u, x') s_2(x, u, x') d(x, u, x')$ and the norm $\|s_1(x, u, x')\|_{\mathcal{D}} \triangleq \langle s_1(x, u, x'), s_1(x, u, x')\rangle_{\mathcal{D}}^{1/2}$.

2. Problem description

Let us consider the following continuous-time affine nonlinear system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(0) = x_0 \tag{1}$$

where $[x_1 \ \dots \ x_n]^T \in \mathcal{X} \subset \mathbb{R}^n$ is the state, x_0 is the initial state and $u = [u_1 \ \dots \ u_m]^T \in \mathcal{U} \subset \mathbb{R}^m$ is the control input. Assume that $f(x) + g(x)u(t)$ is Lipschitz continuous on a set $\mathcal{X}$ that contains the origin, $f(0) = 0$, and that the system is stabilizable on $\mathcal{X}$, i.e., there exists a continuous control function such that the system is asymptotically stable on $\mathcal{X}$. In this paper, system dynamics $f(x)$ and $g(x)$ are *unknown* continuous vector or matrix functions of appropriate dimensions.

The optimal control problem under consideration is to find a state feedback control law $u(t) = u^*(x)$ such that the system (1) is closed-loop asymptotically stable, and the following infinite horizon cost function is minimized:

$$V(x_0) \triangleq \int_0^\infty \left(Q(x(t)) + \|u(t)\|_R^2\right) dt \tag{2}$$

where $R > 0$ and $Q(x)$ is a positive definite function, i.e., for $\forall x \neq 0, Q(x) > 0, Q(x) = 0$ only when $x = 0$. Then, the optimal control problem is briefly presented as

$$u(t) \triangleq u^*(x) \triangleq \arg\min_u V(x_0). \tag{3}$$

3. Preliminary works

From the optimal control theory (Bertsekas, 2005; Lewis et al., 2013), if the mathematical model of system (1) is completely known, the optimal control problem (3) with cost function (2) can be converted to solve the following HJB equation:

$$[\nabla V^*(x)]^T f(x) + Q(x) - \frac{1}{4}[\nabla V^*(x)]^T g(x) R^{-1} g^T(x) \nabla V^*(x) = 0 \tag{4}$$

where $V^*(x) \in C^1(\mathcal{X}), V^*(x) \geq 0$ and $V^*(0) = 0$. Then, the optimal controller (3) is given by

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^*(x). \tag{5}$$

It is noted that the optimal control policy (5) depends on the solution $V^*(x)$ of the HJB equation (4). However, the HJB equation is a nonlinear PDE that is impossible to be solved analytically. To obtain its approximate solution, in Saridis and Lee (1979), the HJB equation (4) was successively approximated by a sequence of GHJB equations as follows:

$$[\nabla V^{(i+1)}]^T (f + gu^{(i)}) + Q(x) + \|u^{(i)}\|_R^2 = 0 \tag{6}$$

with

$$u^{(i)} = -\frac{1}{2} R^{-1} g^T(x) \nabla V^{(i)}(x). \tag{7}$$

By providing an initial admissible control policy $u^{(0)}$ (see Definition 1 for admissible control), it has been proven in Saridis and Lee (1979) that the solution of the iterative GHJB equation (6) will converge to the solution of the HJB equation (4), i.e., $\lim_{i\to\infty} V^{(i)} = V^*$ and $\lim_{i\to\infty} u^{(i)} = u^*$.

Definition 1 (*Admissible Control*)**.** For the given system (1), $x \in \mathcal{X}$, a control $u(x)$ is defined to be admissible with respect to cost function (2) on $\mathcal{X}$, denoted by $u(x) \in \mathfrak{U}(\mathcal{X})$, if, (1) u is continuous on $\mathcal{X}$, (2) $u(0) = 0$, (3) $u(x)$ stabilizes the system, and (4) $V(x) < \infty, \forall x \in \mathcal{X}$. □

Remark 1. Note that the GHJB equation (6) is a Lyapunov function equation, which is a linear PDE much simpler than the HJB equation (4). But in Saridis and Lee (1979), no method has been provided to solve the GHJB equation. Thus, Beard et al. (1997) used a Galerkin approximation method to obtain the approximate solution of the GHJB equation (6). However, this approach is completely model-based. □

4. Data-based approximate policy iteration

Since the mathematical model of the system dynamics $f(x)$ and $g(x)$ are completely unknown, the explicit expression of the HJB equation (4) is unavailable. Thus, it is impossible to obtain the solution of HJB equation with model-based approaches. To overcome this problem, a data-based API algorithm is introduced to learn the solution of the HJB equation (4) by using real system data rather than a system model.

4.1. Derivation of data-based policy iteration

To derive the data-based API algorithm, rewrite the system (1) as

$$\dot{x} = f + gu^{(i)} + g[u - u^{(i)}] \tag{8}$$

for $\forall u \in \mathcal{U}$. Let us consider $V^{(i+1)}(x)$, which is the solution of the GHJB equation (6). By using (6) and (7), we take derivative of $V^{(i+1)}(x)$ with respect to time along the state of system (8)

$$\begin{aligned}\frac{dV^{(i+1)}(x)}{dt} &= [\nabla V^{(i+1)}]^T(f + gu^{(i)}) + [\nabla V^{(i+1)}]^T g[u - u^{(i)}] \\ &= -Q(x) - \|u^{(i)}\|_R^2 + 2[u^{(i+1)}]^T R[u^{(i)} - u].\end{aligned} \tag{9}$$

Integrating both sides of (9) on the interval $[t, t + \Delta t]$ and rearranging terms yields

$$\begin{aligned}&V^{(i+1)}(x(t)) - V^{(i+1)}(x(t+\Delta t)) \\ &\quad + 2\int_t^{t+\Delta t} [u^{(i+1)}(x(\tau))]^T R[u^{(i)}(x(\tau)) - u(\tau)]d\tau \\ &= \int_t^{t+\Delta t} [Q(x(\tau)) + \|u^{(i)}(x(\tau))\|_R^2]d\tau.\end{aligned} \tag{10}$$

In (10), $V^{(i+1)}(x)$ and $u^{(i+1)}(x)$ are the unknown function and vector function to be determined, respectively. Given an initial admissible control policy $u^{(0)}$, the problem of solving the GHJB equation (6) for $V^{(i+1)}(x)$ is transformed to the problem of solving Eq. (10) for $V^{(i+1)}(x)$ and $u^{(i+1)}(x)$. Compared with the GHJB equation (6), Eq. (10) does not require the explicit mathematical model of system (1), i.e., $f(x)$ and $g(x)$.

Remark 2. Note that in iterative equation (10), the system dynamics $f(x)$ and $g(x)$ are not required. In fact, their information is embedded in the measurement of the state x and control signal u. Thus, the lack of information about the system model does not have any impact on the model-free policy iteration algorithm for learning the solution of the HJB equation and the optimal control policy. The resulting control policy learns with the real process behavior and thus does not suffer from model inaccuracy or simplifications made in the design process. Furthermore, in contrast to control methods based on the nonparametric identification models, the issue of collecting system data is also incorporated within the learning process and can be concentrated on regions important to the control application. □

The convergence of the data-based policy iteration with (10) is established in Theorem 1.

Theorem 1. *Let $V^{(i+1)}(x) \in C^1(\mathcal{X})$, $V^{(i+1)}(x) \geq 0$, $V^{(i+1)}(0) = 0$ and $u^{(i+1)}(x) \in \mathfrak{U}(\mathcal{X})$. $(V^{(i+1)}(x), u^{(i+1)}(x))$ is the solution of Eq. (10) iff (if and only if) it is the solution of the GHJB equations (6) and (7), i.e., Eq. (10) is equivalent to the GHJB equation (6) with (7).*

Proof. From the derivation of Eq. (10), it is concluded that if $(V^{(i+1)}, u^{(i+1)})$ is the solution of the GHJB equation (6) with (7), then $(V^{(i+1)}, u^{(i+1)})$ also satisfies Eq. (10). To complete the proof, we have to show that $(V^{(i+1)}, u^{(i+1)})$ is the unique solution of Eq. (10). The proof is by contradiction.

Before starting the contradiction proof, we derive a simple fact. Consider any function $\hbar(t)$, then

$$\begin{aligned}&\lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}\hbar(\tau)d\tau \\ &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\left(\int_0^{t+\Delta t}\hbar(\tau)d\tau - \int_0^t \hbar(\tau)d\tau\right) \\ &= \frac{d}{dt}\int_0^t \hbar(\tau)d\tau = \hbar(t).\end{aligned} \tag{11}$$

From (10), we have

$$\begin{aligned}&\frac{dV^{(i+1)}(x)}{dt} \\ &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\left(V^{(i+1)}(x(t+\Delta t)) - V^{(i+1)}(x(t))\right) \\ &= 2\lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}[u^{(i+1)}(x(\tau))]^T R[u^{(i)}(x(\tau)) - u(\tau)]d\tau \\ &\quad - \lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}[Q(x(\tau)) + \|u^{(i)}(x(\tau))\|_R^2]d\tau.\end{aligned} \tag{12}$$

By using (11), Eq. (12) is rewritten as

$$\begin{aligned}\frac{dV^{(i+1)}(x)}{dt} &= 2[u^{(i+1)}(x(t))]^T R[u^{(i)}(x(t)) - u(t)] \\ &\quad - Q(x(t)) - \|u^{(i)}(x(t))\|_R^2.\end{aligned} \tag{13}$$

Suppose that $(W(x), v(x))$ is another solution of Eq. (10), where $W(x) \in C^1(\mathcal{X})$ with boundary condition $W(0) = 0$ and $v(x) \in \mathfrak{U}(\mathcal{X})$. Thus, (W, v) also satisfies Eq. (13), i.e.,

$$\begin{aligned}\frac{dW(x)}{dt} &= 2[v(x(t))]^T R[u^{(i)}(x(t)) - u(t)] \\ &\quad - Q(x(t)) - \|u^{(i)}(x(t))\|_R^2.\end{aligned} \tag{14}$$

Subtracting Eq. (14) from (13) yields,

$$\begin{aligned}&\frac{d}{dt}\left(V^{(i+1)}(x) - W(x)\right) \\ &= 2[u^{(i+1)}(x(t)) - v(x(t))]^T R[u^{(i)}(x(t)) - u(t)].\end{aligned} \tag{15}$$

This means that Eq. (15) holds for $\forall u \in \mathcal{U}$. If letting $u = u^{(i)}$, we have

$$\frac{d}{dt}\left(V^{(i+1)}(x) - W(x)\right) = 0. \tag{16}$$

This implies that $V^{(i+1)}(x) - W(x) = c$ for $\forall x \in \mathcal{X}$, where c is a real constant, and $c = V^{(i+1)}(0) - W(0) = 0$. Then, $V^{(i+1)}(x) - W(x) = 0$, i.e., $V^{(i+1)}(x) = W(x)$ for $\forall x \in \mathcal{X}$. From (15), we have that

$$[u^{(i+1)}(x(t)) - v(x(t))]^T R[u^{(i)}(x(t)) - u(t)] = 0$$

for $\forall u \in \mathcal{U}$, thus $u^{(i+1)}(x) - v(x) = 0$, i.e., $u^{(i+1)}(x) = v(x)$ for $\forall x \in \mathcal{X}$. This completes the proof. □

It follows from Theorem 1 that the data-based policy iteration with Eq. (10) is equivalent to the iteration of Eqs. (6) and (7), which is convergent as proved in Saridis and Lee (1979). Thus, the convergence of the data-based policy iteration with Eq. (10) can be guaranteed.

4.2. Actor–critic neural network structure

To solve Eq. (10) for $V^{(i+1)}(x)$ and $u^{(i+1)}(x)$ based on data instead of a system model, we develop an actor–critic NN-based approach, where critic and actor NNs are used to approximate cost function $V^{(i+1)}(x)$ and control policy $u^{(i+1)}(x)$ respectively. From the well known high-order Weierstrass approximation theorem (Courant & Hilbert, 2004), it follows that a continuous function can be represented by an infinite-dimensional linearly independent basis function set. For real practical applications, it is necessary to approximate the function in a compact set with a finite-dimensional function set. We consider the critic and actor NNs for approximating the cost function and control policy on a compact set $\mathcal{X}$. Let $\varphi(x) \triangleq [\varphi_1(x)\ \dots\ \varphi_{L_V}(x)]^T$ be the vector of linearly independent activation functions for critic NN, where $\varphi_j(x) : \mathcal{X} \mapsto \mathbb{R}$, $j = 1, \dots, L_V, L_V$ is the number of critic NN hide layer neurons. Let $\psi^l(x) \triangleq [\psi_1^l(x)\ \dots\ \psi_{L_u}^l(x)]^T$, be the vector of linearly independent activation functions of the l-th sub-actor NN for approximating control u_l, $l = 1, \dots, m$, where $\psi_k^l(x) : \mathcal{X} \mapsto \mathbb{R}$, $k = 1, \dots, L_u, L_u$ is the number of actor NN hide layer neurons. Then, the outputs of critic and the l-th actor NNs are given by

$$\hat{V}^{(i)}(x) = \sum_{l=1}^{L_V} \theta_{V,j}^{(i)} \varphi_j(x) = \varphi^T(x)\theta_V^{(i)} \tag{17}$$

$$\hat{u}_l^{(i)}(x) = \sum_{k=1}^{L_u} \theta_{u_l,k}^{(i)} \psi_k^l(x) = (\psi^l(x))^T \theta_{u_l}^{(i)} \tag{18}$$

for $\forall i = 0, 1, 2, \dots$, where $\theta_V^{(i)} \triangleq [\theta_{V,1}^{(i)}\ \dots\ \theta_{V,L_V}^{(i)}]^T$ and $\theta_{u_l}^{(i)} \triangleq [\theta_{u_l,1}^{(i)}\ \dots\ \theta_{u_l,L_u}^{(i)}]^T$ are weight vectors of critic and actor NNs respectively. Expression (18) can be rewritten as a compact form

$$\begin{aligned} \hat{u}^{(i)}(x) &= \left[\hat{u}_1^{(i)}(x)\ \dots\ \hat{u}_m^{(i)}(x)\right]^T \\ &= \left[(\psi^1(x))^T\theta_{u_1}^{(i)}\ \dots\ (\psi^m(x))^T\theta_{u_m}^{(i)}\right]^T. \end{aligned} \tag{19}$$

To show the stability of closed-loop system with approximate control policy $\hat{u}^{(i)}(x)$, define estimation error as $\epsilon_u^{(i)}(x) \triangleq \hat{u}^{(i)}(x) - u^{(i)}(x)$. With the approximate control policy (19), the closed-loop system is given by

$$\dot{x} = f(x) + g(x)\hat{u}^{(i)}(x). \tag{20}$$

Select $V^{(i)}(x)$ as the Lyapunov function candidate, where $V^{(i)}(x)$ is the solution of the GHJB equation (6) with index i. By using (6), differentiating $V^{(i)}$ with respect to system (20) yields

$$\begin{aligned} \dot{V}^{(i)}(x) &= [\nabla V^{(i)}]^T(f + g\hat{u}^{(i)}) \\ &= [\nabla V^{(i)}]^T[f + g(u^{(i)} + \epsilon_u^{(i)})] + Q(x) \\ &\quad + \|u^{(i-1)} - u^{(i)}\|_R^2 - Q(x) - \|u^{(i-1)} - u^{(i)}\|_R^2 \\ &= [\nabla V^{(i)}]^T[f + g(u^{(i-1)})] + Q(x) + \|u^{(i-1)}\|_R^2 \\ &\quad + [\nabla V^{(i)}]^T g\epsilon_u^{(i)} - Q(x) - \|u^{(i)}\|_R^2 - \|u^{(i-1)} - u^{(i)}\|_R^2 \\ &= [\nabla V^{(i)}]^T g\epsilon_u^{(i)} - Q(x) - \|u^{(i)}\|_R^2 - \|u^{(i-1)} - u^{(i)}\|_R^2. \end{aligned} \tag{21}$$

Assuming the following condition holds:

$$Q(x) + \|u^{(i)}\|_R^2 + \|u^{(i-1)} - u^{(i)}\|_R^2 \geqslant [\nabla V^{(i)}]^T g\epsilon_u^{(i)} \tag{22}$$

then $\dot{V}^{(i)} \leqslant 0$ according to Eq. (21), i.e., the closed-loop system (20) is asymptotically stable. Based on the uniform approximation property (Courant & Hilbert, 2004) of NN, by choosing appropriate activation function sets and their size L_u for NN, $\epsilon_u^{(i)}$ can be arbitrarily small such that the condition (22) be satisfied.

Considering $R = [r_{l_1,l_2}]_{m\times m}$, we have

$$\|u\|_R^2 = u^T R u = \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2} u_{l_1} u_{l_2}. \tag{23}$$

For notation simplicity, define $x'(t) \triangleq x(t + \Delta t)$ for $\forall t, \ \Delta t$. Due to estimation errors of the critic and actor NNs (17) and (18), the replacement of $V^{(i+1)}$ and $u^{(i+1)}$ in the iterative equation (10) with $\hat{V}^{(i+1)}$ and $\hat{u}^{(i+1)}$ respectively, yields the following residual error:

$$\begin{aligned} &\sigma^{(i)}(x(t), u(t), x'(t)) \triangleq [\varphi(x(t)) - \varphi(x'(t))]^T \theta_V^{(i+1)} \\ &\quad + 2\int_t^{t+\Delta t} [u^{(i)}(x(\tau)) - u(\tau)]^T R u^{(i+1)}(x(\tau)) d\tau \\ &\quad - \int_t^{t+\Delta t} Q(x(\tau)) d\tau - \int_t^{t+\Delta t} \|u^{(i)}(x(\tau))\|_R^2 d\tau. \\ &= [\varphi(x(t)) - \varphi(x'(t))]^T \theta_V^{(i+1)} + 2\sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2} \\ &\quad \times \int_t^{t+\Delta t} [u_{l_1}^{(i)}(x(\tau)) - u_{l_1}(\tau)] u_{l_2}^{(i+1)}(x(\tau)) d\tau \\ &\quad - \int_t^{t+\Delta t} Q(x(\tau)) d\tau \\ &\quad - \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2} \int_t^{t+\Delta t} u_{l_1}^{(i)}(x(\tau)) u_{l_2}^{(i)}(x(\tau)) d\tau \\ &= [\varphi(x(t)) - \varphi(x'(t))]^T \theta_V^{(i+1)} + 2\sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2} \\ &\quad \times \int_t^{t+\Delta t} [(\psi^{l_1}(x(\tau)))^T \theta_{u_{l_1}}^{(i)} - u_{l_1}(\tau)](\psi^{l_2}(x(\tau)))^T \theta_{u_{l_2}}^{(i+1)} d\tau \\ &\quad - \int_t^{t+\Delta t} Q(x(\tau)) d\tau - \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2} \\ &\quad \times \int_t^{t+\Delta t} (\theta_{u_{l_1}}^{(i)})^T \psi^{l_1}(x(\tau))(\psi^{l_2}(x(\tau)))^T \theta_{u_{l_2}}^{(i)} d\tau. \end{aligned} \tag{24}$$

For convenience, define

$$\rho_{\Delta\varphi}(x(t), x'(t)) \triangleq \left[\varphi(x(t)) - \varphi(x'(t))\right] \tag{25}$$

$$\rho_Q(x(t)) \triangleq \int_t^{t+\Delta t} Q(x(\tau)) d\tau \tag{26}$$

$$\rho_\psi^{l_1,l_2}(x(t)) \triangleq \int_t^{t+\Delta t} \psi^{l_1}(x(\tau))(\psi^{l_2}(x(\tau)))^T d\tau \tag{27}$$

$$\rho_{u\psi}^{l_1,l_2}(x(t), u(t)) \triangleq \int_t^{t+\Delta t} u_{l_1}(\tau)(\psi^{l_2}(x(\tau)))^T d\tau \tag{28}$$

where $l_1, l_2 = 1, \dots, m$. Then, expression (24) is rewritten as

$$\begin{aligned} &\sigma^{(i)}(x(t), u(t), x'(t)) = \rho_{\Delta\varphi}^T(x(t), x'(t))\theta_V^{(i+1)} \\ &\quad + 2\sum_{l_2=1}^{m}\left(\sum_{l_1=1}^{m} r_{l_1,l_2}\left[(\theta_{u_{l_1}}^{(i)})^T \rho_\psi^{l_1,l_2}(x(t))\right.\right. \\ &\quad \left.\left. - \rho_{u\psi}^{l_1,l_2}(x(t), u(t))\right]\right)\theta_{u_{l_2}}^{(i+1)} - \rho_Q(x(t)) \end{aligned}$$

$$- \sum_{l_1=1}^{m}\sum_{l_2=1}^{m} r_{l_1,l_2}(\theta_{u_{l_1}}^{(i)})^T \rho_{\psi}^{l_1,l_2}(x(t))\theta_{u_{l_2}}^{(i)}$$

$$= \overline{\rho}^{(i)}(x(t), u(t), x'(t))\theta^{(i+1)} - \pi^{(i)}(x(t)) \tag{29}$$

where $\theta^{(i+1)} \triangleq [(\theta_V^{(i+1)})^T\ (\theta_{u_1}^{(i+1)})^T\ \ldots\ (\theta_{u_m}^{(i+1)})^T]^T$, $\pi^{(i)}(x(t)) \triangleq \rho_Q(x(t)) + \sum_{l_1=1}^m \sum_{l_2=1}^m r_{l_1,l_2}(\theta_{u_{l_1}}^{(i)})^T\ \rho_\psi^{l_1,l_2}(x(t))\theta_{u_{l_2}}^{(i)}$, $\overline{\rho}^{(i)}(x(t), u(t), x'(t)) \triangleq [\rho_{\Delta\varphi}^T(x(t), x'(t))\ 2\rho_{u\psi}^{(i)1}(x(t), u(t))\ \ldots\ 2\rho_{u\psi}^{(i)m}(x(t), u(t))]$, with $\rho_{u\psi}^{(i)l_2}\ (x(t), u(t)) \triangleq \sum_{l_1=1}^m r_{l_1,l_2}[(\theta_{u_{l_1}}^{(i)})^T \rho_\psi^{l_1,l_2}\ (x(t)) - \rho_{u\psi}^{l_1,l_2}\ (x(t), u(t))]$.

For description simplicity, denote $\overline{\rho}^{(i)} = [\overline{\rho}_1^{(i)}\ \ldots\ \overline{\rho}_L^{(i)}]$, where $L \triangleq L_V + mL_u$ is the length of the vector $\overline{\rho}^{(i)}$. Based on the method of weighted residuals (Finlayson, 1972), the unknown critic NN weight vector $\theta^{(i+1)}$ can be computed in such a way that the residual error $\sigma^{(i)}(x, u, x')$ (for $\forall t \geq 0$) of (29) is forced to be zero in some average sense. Thus, we project the residual error $\sigma^{(i)}(x, u, x')$ onto $d\sigma^{(i)}/d\theta^{(i+1)}$ and set the result to zero on domain $\mathscr{D}$ using the inner product, $\langle\cdot,\cdot\rangle_{\mathscr{D}}$, i.e.,

$$\left\langle d\sigma^{(i)}/d\theta^{(i+1)}, \sigma^{(i)}(x, u, x')\right\rangle_{\mathscr{D}} = 0. \tag{30}$$

Then, the substitution of (29) into (30) yields

$$\left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}}\theta^{(i+1)} - \left\langle\overline{\rho}^{(i)}(x, u, x'), \pi^{(i)}(x)\right\rangle_{\mathscr{D}} = 0$$

where the notations $\left\langle\overline{\rho}^{(i)}, \overline{\rho}^{(i)}\right\rangle_{\mathscr{D}}$ and $\left\langle\overline{\rho}^{(i)}, \pi^{(i)}\right\rangle_{\mathscr{D}}$ are given by

$$\left\langle\overline{\rho}^{(i)}, \overline{\rho}^{(i)}\right\rangle_{\mathscr{D}} \triangleq \begin{bmatrix} \left\langle\overline{\rho}_1^{(i)}, \overline{\rho}_1^{(i)}\right\rangle_{\mathscr{D}} & \cdots & \left\langle\overline{\rho}_1^{(i)}, \overline{\rho}_L^{(i)}\right\rangle_{\mathscr{D}} \\ \vdots & \cdots & \vdots \\ \left\langle\overline{\rho}_L^{(i)}, \overline{\rho}_1^{(i)}\right\rangle_{\mathscr{D}} & \cdots & \left\langle\overline{\rho}_L^{(i)}, \overline{\rho}_L^{(i)}\right\rangle_{\mathscr{D}} \end{bmatrix}$$

and $\left\langle\overline{\rho}^{(i)}, \pi^{(i)}\right\rangle_{\mathscr{D}} \triangleq \left[\left\langle\overline{\rho}_1^{(i)}, \pi^{(i)}\right\rangle_{\mathscr{D}}\ \cdots\ \left\langle\overline{\rho}_L^{(i)}, \pi^{(i)}\right\rangle_{\mathscr{D}}\right]^T$.

Thus, $\theta^{(i+1)}$ can be obtained with

$$\theta^{(i+1)} = \left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}}^{-1} \left\langle\overline{\rho}^{(i)}(x, u, x'), \pi^{(i)}(x)\right\rangle_{\mathscr{D}}. \tag{31}$$

Note that the computation of $\left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}}$ and $\left\langle\overline{\rho}^{(i)}(x, u, x'), \pi^{(i)}(x)\right\rangle_{\mathscr{D}}$ involves many numerical integrals on the domain $\mathscr{D}$, which are computationally expensive. Thus, the Monte-Carlo integration method (Peter Lepage, 1978) is introduced, which is especially competitive on multi-dimensional domains. We now illustrate the Monte-Carlo integration for computing $\left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}}$. Let $I_{\mathscr{D}} \triangleq \int_{\mathscr{D}} d(x, u, x')$, and $\mathscr{S}_M \triangleq \{(x_k, u_k, x'_k)\ |(x_k, u_k, x'_k) \in \mathscr{D}, k = 1, 2, \ldots, M\}$ be the set sampled on domain $\mathscr{D}$, where M is the size of the sample set $\mathscr{S}_M$. Then, $\left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}}$ is approximately computed with

$$\begin{aligned} &\left\langle\overline{\rho}^{(i)}(x, u, x'), \overline{\rho}^{(i)}(x, u, x')\right\rangle_{\mathscr{D}} \\ &= \int_{\mathscr{D}} \left(\overline{\rho}^{(i)}(x, u, x')\right)^T \overline{\rho}^{(i)}(x, u, x')d(x, u, x') \\ &= \frac{I_{\mathscr{D}}}{M}\sum_{k=1}^{M}\left(\overline{\rho}^{(i)}(x_k, u_k, x'_k)\right)^T \overline{\rho}^{(i)}(x_k, u_k, x'_k) \\ &= \frac{I_{\mathscr{D}}}{M}\left(Z^{(i)}\right)^T Z^{(i)} \end{aligned} \tag{32}$$

where $Z^{(i)} \triangleq \left[\left(\overline{\rho}^{(i)}(x_1, u_1, x'_1)\right)^T\ \ldots\ \left(\overline{\rho}^{(i)}(x_M, u_M, x'_M)\right)^T\right]^T$. Similarly,

$$\begin{aligned} \left\langle\overline{\rho}^{(i)}(x, u, x'), \pi^{(i)}(x)\right\rangle_{\mathscr{D}} &= \frac{I_{\mathscr{D}}}{M}\sum_{k=1}^{M}\left(\overline{\rho}^{(i)}(x_k, u_k, x'_k)\right)^T \pi^{(i)}(x_k) \\ &= \frac{I_{\mathscr{D}}}{M}\left(Z^{(i)}\right)^T \eta^{(i)} \end{aligned} \tag{33}$$

where $\eta^{(i)} \triangleq \left[\pi^{(i)}(x_1)\ \ldots\ \pi^{(i)}(x_M)\right]^T$. The substitution of (32) and (33) into (31) yields,

$$\theta^{(i+1)} = \left[\left(Z^{(i)}\right)^T Z^{(i)}\right]^{-1}\left(Z^{(i)}\right)^T \eta^{(i)}. \tag{34}$$

The data set $\mathscr{S}_M$ is collected from real system on domain $\mathscr{D}$, which is sampled arbitrary such that the domain $\mathscr{D}$ can be covered adequately. Based on the sample set $\mathscr{S}_M$ and (25)–(28), and using trapezoidal rule for approximating definite integrals, $\rho_{\Delta\varphi}(x_k, x'_k)$, $\rho_Q(x_k)$, $\rho_\psi^{l_1,l_2}(x_k)$ and $\rho_{u\psi}^{l_1,l_2}(x_k, u_k)$ $(k = 1, \ldots, M)$ can be numerically computed with $\rho_{\Delta\varphi}(x_k, x'_k) = \left[\varphi(x_k) - \varphi(x'_k)\right]$, $\rho_Q(x_k) = \frac{\Delta t}{2}[Q(x_k) + Q(x'_k)]$, $\rho_\psi^{l_1,l_2}(x_k) = \frac{\Delta t}{2}[\psi^{l_1}(x_k)(\psi^{l_2}(x_k))^T + \psi^{l_1}(x'_k)(\psi^{l_2}(x'_k))^T]$ and $\rho_{u\psi}^{l_1,l_2}(x_k, u_k) = \frac{\Delta t}{2}[u_{k,l_1}(\psi^{l_2}(x_k))^T + u_{k,l_1}(\psi^{l_2}(x'_k))^T]$. Then, after $Z^{(i)}$ and $\eta^{(i)}$ are computed, $\theta^{(i+1)}$ can be obtained accordingly.

Remark 3. Note that $Z^{(i)} \in \mathbb{R}^{(M\times L)}$, where L represents the number of unknown parameters. The least-square method (34) requires the inverse of matrix $(Z^{(i)})^T Z^{(i)}$, then $Z^{(i)}$ should be full column rank, and its rank would be $\text{rank}(Z^{(i)}) = L$, which is the standard for determining the size M of sample set $\mathscr{S}_M$. Thus, $M \geqslant L$, i.e., the lower bound of M is L. In practical implementation, to achieve $\text{rank}(Z^{(i)}) = L$, two ways could be useful. (1) Increase the size of sample set $\mathscr{S}_M$ such that $M > L$. (2) The input signal is expected to be chosen such that it is persistent exciting (but is not a necessity), which is similar with the issue of "exploration" (will be discussed in Section 4.3) in the machine learning community. □

4.3. Data-based API with off-policy implementation algorithm

In Section 4.2, the developed least-square scheme is designed only for solving one iterative equation (10). Now, we present a complete data-based API algorithm procedure as follows:

Algorithm 1. Data-based API algorithm.

- Step 1: With different input signal u, collect real system data (x_k, u_k, x'_k) for sample set $\mathscr{S}_M$, and compute $\rho_{\Delta\varphi}(x_k, x'_k)$, $\rho_Q(x_k)$, $\rho_\psi^{l_1,l_2}(x_k)$ and $\rho_{u\psi}^{l_1,l_2}(x_k, u_k)$;
- Step 2: Let the initial actor NN weight vector $\theta_{u_l}^{(0)}$ $(l = 1, \ldots, m)$ such that $\hat{u}^{(0)} \in \mathfrak{U}(\mathcal{X})$, and the initial critic NN weight $\theta_V^{(0)} = 0$. Let $i = 0$;
- Step 3: Compute $Z^{(i)}$ and $\eta^{(i)}$, and update $\theta^{(i+1)}$ with (34);
- Step 4: Let $i = i + 1$. If $\|\theta^{(i)} - \theta^{(i-1)}\| \leq \xi$ (ξ is a small positive number), stop iteration and $\theta^{(i)}$ is employed to obtain the final control policy with (19), else go back to Step 3 and continue. □

Remark 4. It is found that the data-based API algorithm uses real system state and input information of the closed-loop system instead of a dynamic model, for learning the optimal control policy (5) and the solution of the HJB equation (4). The procedure of the API algorithm can be divided into a preparation part and an offline control design part. (1) Step 1 is the preparation part for data processing. By collecting system state and input signal, compute $\rho_{\Delta\varphi}(x_k, x'_k)$, $\rho_Q(x_k)$, $\rho_\psi^{l_1,l_2}(x_k)$ and $\rho_{u\psi}^{l_1,l_2}(x_k, u_k)$, and then prepare for iteration. In fact, the information of the system dynamics is embedded in the data measured, and thus explicit system identification is avoided. (2) Steps 2–4 are the offline part for iterative learning the optimal control policy and the solution of HJB equation. After the iterations have converged, the resulting actor NN weight is applied to obtain the optimal control policy for real control. □

Remark 5. In the optimal control community, bang–bang control is a popular method applied in practice. It is noted that the bang–bang control signal is of discontinuous form, while the proposed data-based API algorithm requires a continuous control law. Thus, the data-based API algorithm may not be directly applied to the bang–bang control design. On the other hand, for the optimal control problem of practical systems, many issues are expected to be involved, such as state constraints, input constraints, external disturbance, and uncertainties. Due to the complexity and difficulty of the control design when considering these issues, this paper is concerned only with the optimal control problem of affine nonlinear systems without involving these issues. Moreover, it is still theoretically unclear whether the proposed data-based API algorithm can be extended to solve these problems or not, and thus they are left for future investigation. □

Remark 6. Observe that the data-based API (i.e., Algorithm 1) is an "off-policy" learning method (Luo, Wu, & Huang, in press; Precup, Sutton, & Dasgupta, 2001; Sutton & Barto, 1998), which means that cost function $V^{(i+1)}(x)$ of control policy $u^{(i)}(x)$ can be evaluated by using system data generated with different control policies u. Off-policy learning, the ability for an agent to learn about an optimal policy rather than the one it is following (Luo, Wu, & Huang, in press; Precup et al., 2001; Sutton & Barto, 1998), is a key element of reinforcement learning. The rationality of using the "off-policy" learning method is that it can overcome the difficulty of inadequate exploration, which widely exists in reinforcement learning approaches. Generally, to evaluate the cost function of a control policy μ, it needs to generate system data using policy μ. This biases the learning process by under-representing states that are unlikely to occur under μ. As a result, the estimated cost function of these underrepresented states may be highly inaccurate, and seriously impact the improved policy, which is known as inadequate exploration. For the proposed data-based policy iteration algorithm, the control u and state x can be selected arbitrarily on $\mathcal{U}$ and $\mathcal{X}$, which greatly increases the "exploration" ability during the learning process. □

Remark 7. Most recently, we noted that similar expressions like (10) has been reported in Jiang and Jiang (2014, 2013). Even though, there are several main contributions and differences of this paper compared with the work in Jiang and Jiang (2014, 2013). 1. In Theorem 1, it is proved that the iterative equation (10) is theoretically equivalent to the iterative equations (6) and (7), and the uniqueness of its solution is also demonstrated clearly. Thus, the solution of the iterative equation (10) will converge to the solution of the HJB equation (4). 2. Under the consideration of NN estimation errors, the NN weight vector update rule (34) is derived rigorously with the method of weighted residuals. 3. The methods in Jiang and Jiang (2014, 2013) are online adaptive control approaches, which learn the cost function and control policy online by using system data generated along the neighborhood of a single signal trajectory, and thus will result in deficiency of inadequate exploration as described in Remark 6. The inadequate exploration problem is a particularly difficult issue in RL methods, which is rarely discussed in the existing works using RL techniques for optimal control design. In this paper, the interesting off-policy RL framework is introduced for developing a data-based API algorithm (i.e., Algorithm 1), which learns the cost function and control policy offline, and then the convergent control policy can be employed for real time control. According to the method of weighted residuals described in Section 4.2, the system data for Algorithm 1 can be selected arbitrary on domain $\mathcal{D}$, which implies that the data can be collected from different signal trajectories such that the domain $\mathcal{D}$ be covered adequately, and thus inadequate exploration problem is overcome. □

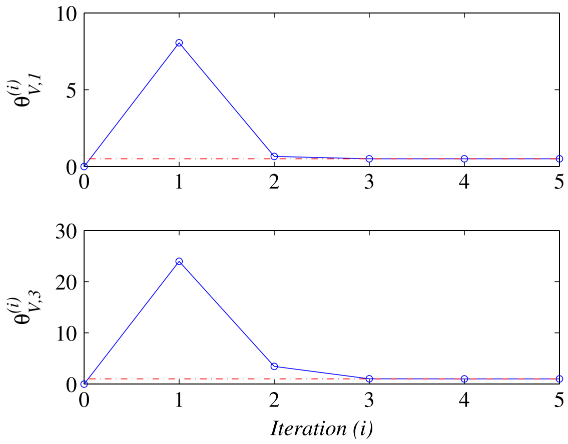


Fig. 1. For example 1, two representative critic NN weights $\theta^{(i)}_{V,1}$ and $\theta^{(i)}_{V,3}$.

5. Simulation studies

In this section, we first test the effectiveness of the developed data-based API algorithm on a simple nonlinear numerical system, and further apply it to the complex RTAC nonlinear benchmark problem.

5.1. Example 1: effectiveness test on a simple nonlinear numerical system

The numerical example is constructed by using the converse HJB approach (Nevistić & Primbs, 1996). The system model is given as follows:

$$\dot{x} = \begin{bmatrix} -x_1 + x_2 \\ -0.5(x_1 + x_2) + 0.5x_1^2 x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ x_1 \end{bmatrix} u, \qquad x_0 = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}. \tag{35}$$

With the choice of $Q(x) = x^T x$ and $R = I$ in the cost function (2), the solution of the associated HJB equation (4) is $V^*(x) = 0.5x_1^2 + x_2^2$, and thus $u^*(x) = -x_1 x_2$.

In the data-based API algorithm, select the critic NN activation function vector as $\varphi(x) = [x_1^2 \; x_1x_2 \; x_2^2]^T$ with the size of $L_V = 3$, actor NN activation function vector as $\psi(x) = [x_1 \; x_2 \; x_1^2 \; x_1x_2 \; x_2^2]^T$ with the size of $L_u = 5$, and the initial actor NN weight vector $\theta_u^{(0)} = [-5 \; -5 \; -5 \; -5 \; -5]^T$. Since $V^*(x) = 0.5x_1^2 + x_2^2$ and $u^*(x) = -x_1x_2$, the optimal critic and actor NN weight vectors are $\theta_V^* = [0.5 \; 0 \; 1]^T$ and $\theta_u^* = [0 \; 0 \; 0 \; -1 \; 0]^T$, respectively.

After collecting sample set $\mathcal{S}_M$ and computing $\rho_{\Delta\varphi}(x_k, x_k')$, $\rho_Q(x_k)$, $\rho_\psi^l(x_k)$, $\rho_{u\psi}^l(x_k, u_k)$, offline iteration (i.e., Steps 2–4) is used to learn the optimal control policy. Setting the value of the convergence criterion $\xi = 10^{-5}$, it is found that the critic and actor NN weight vectors converge respectively to θ_V^* and θ_u^*, at the 5th iteration. Fig. 1 shows two representative critic NN weights $\theta^{(i)}_{V,1}$ and $\theta^{(i)}_{V,3}$, and Fig. 2 shows two representative actor NN weights $\theta^{(i)}_{u,1}$ and $\theta^{(i)}_{u,4}$, wherein the dashed lines are optimal values of the weights. By using the converged actor NN weights $\theta_u^{(5)}$, closed-loop

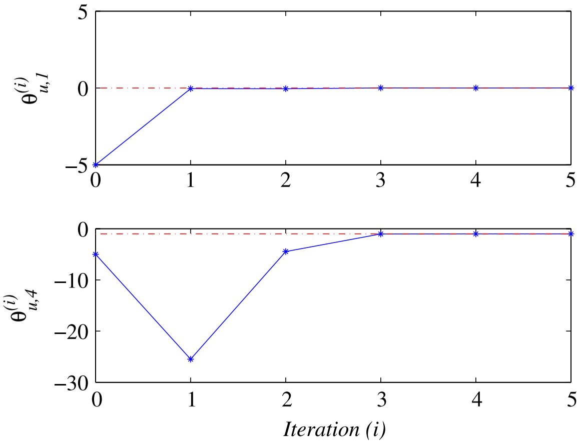


Fig. 2. For example 1, two representative actor NN weights $\theta^{(i)}_{u,1}$ and $\theta^{(i)}_{u,4}$.

simulation is conducted with control policy of (19), and the real cost (2) is 0.0150. Thus, the simulation on this simple nonlinear system demonstrates the effectiveness of the developed data-based API algorithm.

5.2. Example 2: application to the RTAC nonlinear benchmark problem

The rotational/translational actuator (RTAC) nonlinear benchmark problem has been widely used to test the abilities of control methods. The dynamics of this nonlinear plant poses challenges as the rotational and translation motions are coupled. The RTAC system is given as follows:

$$\dot{x} = \begin{bmatrix} x_2 \\ \dfrac{-x_1 + \zeta x_4^2 \sin x_3}{1 - \zeta^2 \cos^2 x_3} \\ x_4 \\ \dfrac{\zeta \cos x_3 (x_1 - \zeta x_4^2 \sin x_3)}{1 - \zeta^2 \cos^2 x_3} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{-\zeta \cos x_3}{1 - \zeta^2 \cos^2 x_3} \\ 0 \\ \dfrac{1}{1 - \zeta^2 \cos^2 x_3} \end{bmatrix} u,$$

$$x_0 = [0.2 \ -0.2 \ 0.2 \ -0.2]^T \tag{36}$$

where $\zeta = 0.2$. Let $Q(x) = x^T x$ and $R = I$ in the cost function (2).

To learn the optimal control policy with the data-based API algorithm, select the critic NN activation function vector as $\varphi(x) = [x_1^2 \ x_1x_2 \ x_1x_3 \ x_1x_4 \ x_2^2 \ x_2x_3 \ x_2x_4 \ x_3^2 \ x_3x_4 \ x_4^2 \ x_1^3 \ x_2^3 \ x_3^3 \ x_4^3 \ x_1^4 \ x_2^4 \ x_3^4 \ x_4^4]^T$ with the size of $L_V = 18$, actor NN activation function vector as $\psi(x) = [x_1 \ x_2 \ x_3 \ x_4 \ x_1^2 \ x_1x_2 \ x_1x_3 \ x_1x_4 \ x_2^2 \ x_2x_3 \ x_2x_4 \ x_3^2 \ x_3x_4 \ x_4^2 \ x_1^3 \ x_2^3 \ x_3^3 \ x_4^3]^T$ with the size of $L_u = 18$, and initial actor NN weight vector as $\theta_u^{(0)} = [6.1302 \ -0.4006 \ -5.0000 \ -9.3413 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$. After the preparation part of Algorithm 1 is completed, offline iteration (i.e., Steps 2–4) is used to learn the optimal control policy. Setting the value of convergence criterion $\xi = 10^{-5}$, it is observed that at the 8th iteration, the critic NN weight vector converges to $\theta_V^{(8)} = [13.2386 \ -0.9901 \ 0.1641 \ -2.5512 \ 13.3557 \ 2.8576 \ 5.4960 \ 1.8723 \ 2.4949 \ 2.3443 \ 0.0108 \ -0.0015 \ 0.0260 \ 0.0019 \ -0.7807 \ 0.8968 \ 0.5084 \ 0.0126]^T$ and the actor NN weight vector converges to $\theta_u^{(8)} = [1.2250 \ -0.0817 \ -1.0011 \ -1.8698 \ 0.0186 \ -0.0037 \ -0.0015 \ -0.0016 \ -0.0091 \ -0.0106 \ -0.0081 \ -0.0047 \ -0.0003 \ -0.0038 \ 0.9607 \ 0.4003 \ -0.4773 \ -0.0162]^T$. Fig. 3 shows six representative critic NN weights $\theta^{(i)}_{V,1}, \theta^{(i)}_{V,4}, \theta^{(i)}_{V,5}, \theta^{(i)}_{V,8}, \theta^{(i)}_{V,10}$ and $\theta^{(i)}_{V,18}$, and Fig. 4 gives six representative actor NN weights $\theta^{(i)}_{u,1}, \theta^{(i)}_{u,2}, \theta^{(i)}_{u,3}, \theta^{(i)}_{u,4}, \theta^{(i)}_{u,10}$ and $\theta^{(i)}_{u,17}$. By using the convergent actor NN weights, closed-loop simulation is conducted with control policy (19), and the real cost (2) is 1.3862. Figs. 5 and 6 demonstrate the state trajectories and control action, respectively.

6. Conclusions

The model-free optimal control problem of nonlinear continuous-time systems is addressed by proposing a data-based API algorithm, and its convergence is proved. The data-based API method learns the solution of the HJB equation and the optimal control policy from real system data instead of a mathematical model. Based on the actor–critic-NN structure, the algorithm implementation procedure is developed under the off-policy RL framework, which contains a preparation part for data processing, and an offline part

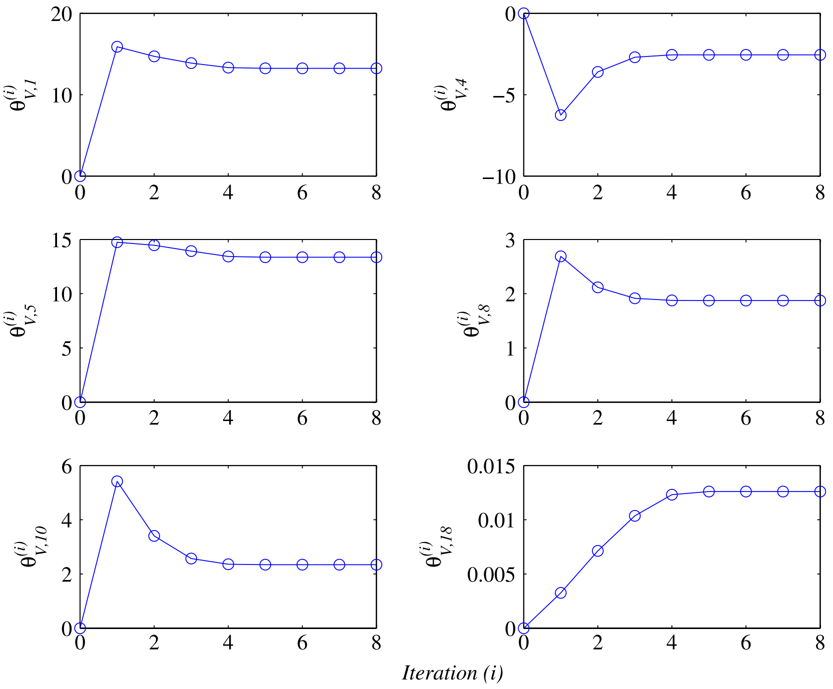


Fig. 3. For example 2, six representative critic NN weights $\theta^{(i)}_{V,1}, \theta^{(i)}_{V,4}, \theta^{(i)}_{V,5}, \theta^{(i)}_{V,8}, \theta^{(i)}_{V,10}$ and $\theta^{(i)}_{V,18}$.

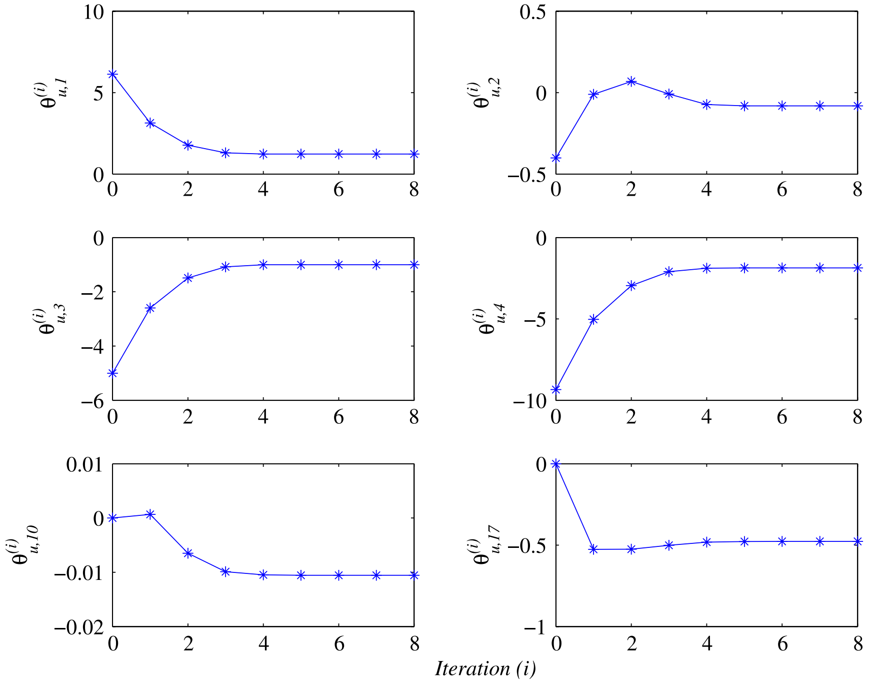


Fig. 4. For example 2, six representative actor NN weights $\theta^{(i)}_{u,1}$, $\theta^{(i)}_{u,2}$, $\theta^{(i)}_{u,3}$, $\theta^{(i)}_{u,4}$, $\theta^{(i)}_{u,10}$ and $\theta^{(i)}_{u,17}$.

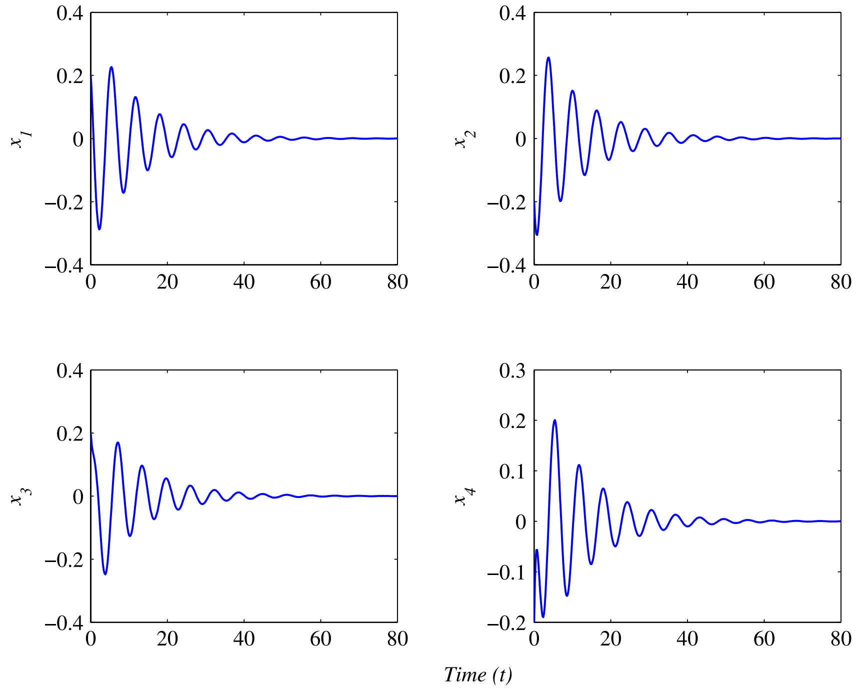


Fig. 5. For example 2, the system state trajectories.

for iterative learning the optimal critic and actor NN weight vectors. The applications on a simple nonlinear numerical system and a RTAC benchmark system demonstrate the effectiveness of the developed data-based API optimal control design method.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) (2012CB720003), the National Natural Science Foundation of China under Grant (61121003), the General Research Fund project from Science and Technology on Aircraft Control Laboratory of Beihang University under Grant 9140C480301130C48001. This publication was also made possible by NPRP grant # NPRP 4-1162-1-181 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the author[s]. The authors would like to thank anonymous reviewers for their valuable comments.

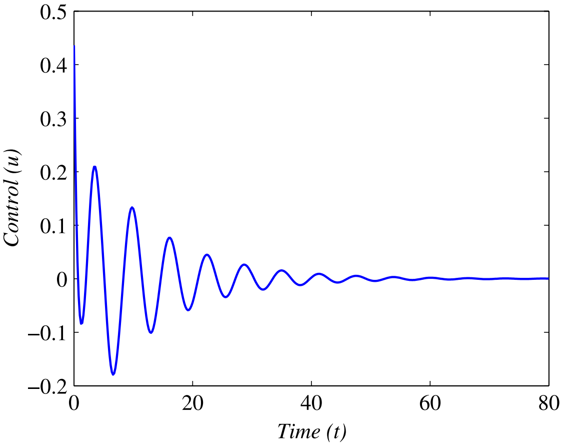


Fig. 6. For example 2, the trajectory of control action.

References

Abu-Khalaf, Murad, & Lewis, Frank L. (2005). Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica*, *41*(5), 779–791.

Al-Tamimi, Asma, Lewis, Frank L., & Abu-Khalaf, Murad (2008). Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *38*(4), 943–949.

Beard, Randal W., Saridis, George N., & Wen, John T. (1997). Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation. *Automatica*, *33*(12), 2159–2177.

Bertsekas, Dimitri P. (2005). *Dynamic programming and optimal control, Vol. 1*. Nashua: Athena Scientific.

Courant, Richard, & Hilbert, David (2004). *Methods of mathematical physics, Vol. 1*. Wiley.

Finlayson, Bruce A. (1972). *The method of weighted residuals and variational principles: with applications in fluid mechanics, heat and mass transfer, Vol. 87*. New York: Academic Press, Inc..

Heydari, Ali, & Balakrishnan, Sivasubramanya N. (2013). Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(1), 147–157.

Hull, David G. (2003). *Optimal control theory for applications*. Troy, NY: Springer.

Jiang, Yu, & Jiang, Z.-P. (2014). Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 882–893.

Jiang, Yu, & Jiang, Zhong-Ping (2012). Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, *48*(10), 2699–2704.

Jiang, Zhong-Ping, & Jiang, Yu (2013). Robust adaptive dynamic programming for linear and nonlinear systems: an overview. *European Journal of Control*, *19*(5), 417–425.

Kleinman, David L. (1968). On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, *13*(1), 114–115.

Lee, Jae Young, Park, Jin Bae, & Choi, Yoon Ho (2012). Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. *Automatica*, *48*(11), 2850–2859.

Lewis, Frank L., & Liu, Derong (2013). *Reinforcement learning and approximate dynamic programming for feedback control, Vol. 17*. Hoboken, New Jersey: John Wiley & Sons, Inc..

Lewis, Frank L., Vrabie, Draguna, & Syrmos, Vassilis L. (2013). *Optimal control*. Hoboken, New Jersey: John Wiley & Sons, Inc..

Lewis, Frank L., Vrabie, Draguna, & Vamvoudakis, Kyriakos G. (2012). Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems*, *32*(6), 76–105.

Lin, Qun, Loxton, Ryan, & Teo, Kok Lay (2014). The control parameterization method for nonlinear optimal control: a survey. *Journal of Industrial and Management Optimization*, *10*(1), 275–309.

Liu, Derong, Wang, Ding, & Li, Hongliang (2014). Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(2), 418–428.

Luo, Biao, Wu, Huai-Ning, & Huang, Tingwen (2014). Off-policy reinforcement learning for H_∞ control design. *IEEE Transactions on Cybernetics*, http://dx.doi.org/10.1109/TCYB. 2014.2319577. in press.

Luo, Biao, Wu, H.-N., & Li, H.-X. (2014). Adaptive optimal control of highly dissipative nonlinear spatially distributed processes with neuro-dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2014.2320744. in press.

Luo, Biao, Wu, Huai-Ning, & Li, Han-Xiong (2014). Data-based suboptimal neuro-control design with reinforcement learning for dissipative spatially distributed processes. *Industrial & Engineering Chemistry Research*, *53*(29), 8106–8119.

Nevistić, Vesna, & Primbs, James A. (1996). *Optimality of nonlinear design techniques: a converse HJB approach. Technical report TR96-022*. California Institute of Technology.

Peter Lepage, G. (1978). A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, *27*(2), 192–203.

Precup, Doina, Sutton, Richard S., & Dasgupta, Sanjoy (2001). Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th international conference on machine learning* (pp. 417–424).

Saridis, George N., & Lee, Chun-Sing G. (1979). An approximation theory of optimal control for trainable manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, *9*(3), 152–159.

Si, Jennie, & Wang, Yu-Tsung (2001). Online learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, *12*(2), 264–276.

Sutton, Richard S., & Barto, Andrew G. (1998). *Reinforcement learning: an introduction*. Massachusetts London, England: Cambridge Univ Press.

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

Vrabie, Draguna, & Lewis, Frank L. (2009). Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Networks*, *22*(3), 237–246.

Wang, Ling Yun, Gui, Wei Hua, Teo, Kok Lay, Loxton, Ryan C., & Yang, Chun Hua (2009). Time delayed optimal control problems with multiple characteristic time points: computation and industrial applications. *Journal of Industrial and Management Optimization*, *5*(4), 705–718.

Wang, Ding, Liu, Derong, Wei, Qinglai, Zhao, Dongbin, & Jin, Ning (2012). Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica*, *48*(8), 1825–1832.

Zhang, Huaguang, Cui, Lili, Zhang, Xin, & Luo, Yanhong (2011). Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Transactions on Neural Networks*, *22*(12), 2226–2236.

Biao Luo is an Assistant Professor at Institute of Automation, Chinese Academy of Sciences, Beijing, China. He received his B.E. degree and his M.E. degree from Xiangtan University, Xiangtan, China, 2006 and 2009, and his Ph.D. degree from Beihang University, Beijing, China, 2014, respectively.

From February 2013 to August 2013, he was a Research Assistant with the Department of System Engineering and Engineering Management (SEEM), City University of Hong Kong, Kowloon, Hong Kong. From September 2013 to December 2013 and from June 2014 to August 2014, he was a Research Assistant with Department of Mathematics and Science, Texas A&M University at Qatar, Doha, Qatar. His current research interests include distributed parameter systems, optimal control, data-based control, fuzzy/neural modeling and control, hypersonic entry/reentry guidance, learning and control from big data, reinforcement learning, approximate dynamic programming, and evolutionary computation.

He was a recipient of the Excellent Master Dissertation Award of Hunan Province in 2011.

Huai-Ning Wu was born in Anhui, China, on November 15, 1972. He received the B.E. degree in Automation from Shandong Institute of Building Materials Industry, Jinan, China and the Ph.D. degree in Control Theory and Control Engineering from Xi'an Jiaotong University, Xian, China, in 1992 and 1997, respectively.

From August 1997 to July 1999, he was a Postdoctoral Researcher in the Department of Electronic Engineering at Beijing Institute of Technology, Beijing, China. In August 1999, he joined the School of Automation Science and Electrical Engineering, Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing. From December 2005 to May 2006, he was a Senior Research Associate with the Department of Manufacturing Engineering and Engineering Management (MEEM), City University of Hong Kong, Kowloon, Hong Kong. From October to December during 2006–2008 and from July to August in 2010, he was a Research Fellow with the Department of MEEM, City University of Hong Kong. From July to August in 2011 and 2013, he was a Research Fellow with the Department of Systems Engineering and Engineering Management, City University of Hong Kong. He is currently a Professor with Beihang University. His current research interests include robust control, fault-tolerant control, distributed parameter systems, and fuzzy/neural modeling and control.

He serves as Associate Editor of the IEEE Transactions on Systems, Man & Cybernetics: Systems. He is a member of the Committee of Technical Process Failure Diagnosis and Safety, Chinese Association of Automation.

Tingwen Huang is a Professor at Texas A&M University at Qatar. He received his B.S. degree from Southwest Normal University (now Southwest University), China, 1990, his M.S. degree from Sichuan University, China, 1993, and his Ph.D. degree from Texas A&M University, College Station, Texas, 2002. After graduated from Texas A&M University, he worked as a Visiting Assistant Professor there. Then he joined Texas A&M University at Qatar (TAMUQ) as an Assistant Professor in August 2003, then he was promoted to Professor in 2013. His focus areas for research interests include neural networks, chaotic dynamical systems, complex networks, optimization and control. He has authored and co-authored more than 100 refereed journal papers.

Derong Liu received the Ph.D. degree in Electrical Engineering from the University of Notre Dame in 1994. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, and became a Full Professor of Electrical and Computer Engineering and of Computer Science in 2006. He was selected for the ''100 Talents Program'' by the Chinese Academy of Sciences in 2008. He has published 15 books (six research monographs and nine edited volumes). Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems. He received the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999), the University Scholar Award from University of Illinois (2006–2009), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008). He is a Fellow of the IEEE and a Fellow of the INNS.