Model-Free Adaptive Algorithm for Optimal Control of Continuous-Time Nonlinear System

Yuanheng Zhu and Dongbin Zhao
The State Key Laboratory of Management and Control for Complex Systems
Institution of Automation, Chinese Academy of Sciences, Beijing 100190, China
Email: yuanheng.zhu@ia.ac.cn, dongbin.zhao@ia.ac.cn

***Abstract*—Reinforcement learning has provided an efficient approach to solve the optimal control of some complicated systems. In this paper we resort to the idea of off-policy scheme and propose a complete-model-free algorithm to the continuous-time nonlinear optimal control problems. The novel algorithm consists of two neural networks to approximate the value and policy and adapts them with continuous tuning laws. In addition, experience replay technique is employed so that both the instantaneous observations and the past data are utilized. The convergence to the optimal solutions are guaranteed under the Lyapunov analysis. A nonlinear system is simulated to test the learning performance.**

Keywords*—Reinforcement learning; off-policy; model-free; experience replay.

I. Introduction

During the last few decades, reinforcement learning (RL) and adaptive dynamic programming (ADP) have provided efficient model-free solutions to the control problems in an optimal manner. Through the interaction with the environment, a value function is calculated to evaluate the control effect in regarding to the current policy, which is further adjusted under the guidance of the value function to improve the performance. In the early research, considerable effort was made upon the optimal control of discrete-time systems [1], [2], [3], but in the real applications, dynamics is mostly continuous-time. Due to that, extension of RL to continuous-time systems has gained a lot of interest.

In [4], [5], an online algorithm was developed to solve the optimal control of continuous-time nonlinear systems. Combination of the integral reinforcement learning (IRL) technique and policy iteration (PI), makes the algorithm independent of the internal dynamics. The algorithm learns in an *on-policy* manner, i.e. the actions producing system solutions are drawn from the current evaluated policy. As policies remain admissible, the system has to be reset to unstable points in order to avoid the states being stuck at the origin.

To make the RL-based algorithms completely model-free, [6], [7], [8], [9], [10] put forward various algorithms for different cases to solve the optimal control problem. The methodology that these works have in common is the system trajectory, which forms the calculation of these algorithms, are generated following a different control input in contrast to the evaluated policy. After putting the data into a novel policy iteration equation, which we term as *off-policy* PI, a sequence of value functions and policies are yield, which converge to the optimal solutions.

Different to the above algorithms that the critic (value function) and actor (policy) are updated at discrete moments in time, [11] devised a continuous adaptive algorithm with known dynamics. The parameters of the critic and actor are adapted in an online real-time manner and the closed-loop system stability and the approximation errors are uniformly ultimately bounded. Inspired by this work, a series of algorithms [12], [13], [14], [15] are proposed. It is interesting to point out that these algorithms are all on-policy. But during online learning, exploratory input signals are necessary to excite the system, so the executed policy differs to the learned one. Such variance is harmful to the on-policy adaptation. Motivated by that, we resort to off-policy method and propose a novel adaptive optimal algorithms for the continuous-time nonlinear system. The convergence to the optimal solution is established using Lyapunov analysis. Relying on the off-policy property, the algorithm is completely model-free in contrast to the aforementioned on-policy algorithms that require the knowledge of dynamics or use a model identifier.

In the context of RL, experience replay (ER) is a useful technique for online learning. Most RL-based algorithms are conducted only based on instantaneous observation, but history data also contain the complete information about the system and are useful for the tuning. In this light, experience replay is devised to repeatedly utilize the past data and concurrently form the adaptation together with the instantaneous data. We bring experience replay into our continuous adaptation law to learn the optimal solutions and it is demonstrated that the utilization of past data significantly improve the convergence rate.

II. Preliminary

The system considered here is a continuous-time nonlinear input-affine system described by dynamics

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

where the state $x(t) \in \mathbb{R}^n$, control $u(t) \in \mathbb{R}^m$, internal dynamics $f(x(t)) \in \mathbb{R}^n$, and input gain matrix $g(x(t)) \in \mathbb{R}^{n\times m}$. We assume $f(0) = 0$ and f, g are Lipschitz continuous on a compact set $\Omega \in \mathbb{R}^n$ that contains the origin. In addition, we assume (1) is stabilizable on Ω.

This work is supported by National Natural Science Foundation of China (NSFC) under Grants No. 61273136.

For a policy $u \equiv u(x(t))$, the performance index is defined by an infinite horizon integral cost known as *value function*

$$V(x(0)) \equiv \int_0^\infty \big(Q(x) + u^T R u\big) d\tau \tag{2}$$

where $Q(\cdot)$ is a positive definite function with $Q(x) > 0$, $\forall x \neq 0$ and $Q(0) = 0$, and R is a symmetric positive definite matrix.

Definition 1 (Admissible): As for a control policy u, if u is continuous on Ω, $u(0) = 0$, $u(x(t))$ stabilizes (1) on Ω, and its value $V(x)$ defined by (2) is finite $\forall x \in \Omega$, then u is said to be admissible on Ω, denoted by $u \in \Psi(\Omega)$.

The optimal control is to find the optimal admissible policy $u^* \in \Psi(\Omega)$ that has the lowest performance index for every state, called the *optimal policy*. The corresponding value function is called the *optimal value function*, denoted by $V^*(x) = \min_{u \in \Psi(\Omega)} V(x)$. We assume there exists a unique solution to the optimal control problem and V^* is continuously differentiable on Ω, i.e. $V^* \in \mathcal{C}^1(\Omega)$.

An infinitesimal equivalent to the value function definition (2) is a Bellman equation

$$\nabla V^T (f + gu) + Q + u^T R u = 0, V(0) = 0 \tag{3}$$

Define the Hamiltonian function

$$H(x, \nabla V, u) \equiv \nabla V^T (f + gu) + Q + u^T R u$$

where ∇ denotes the partial derivative operator, i.e. $\nabla V = \partial V / \partial x$. The optimal control is equivalent to solving V^* w.r.t the Hamilton-Jacobi-Bellman (HJB) equation

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

and the optimal policy is formulated by

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^*(x) \tag{4}$$

In general, it is difficult or impossible to give an analytical solution to the HJB equation even for simple cases. An efficient approach is iteratively solving the problem based on policy iteration (PI), which involves a *policy evaluation* step

$$\begin{aligned} &[\nabla V^{(i)}]^T (f + g u^{(i)}) + Q + [u^{(i)}]^T R u^{(i)} = 0, \\ &V^{(i)}(0) = 0 \end{aligned} \tag{5}$$

and a *policy improvement* step

$$u^{(i+1)}(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^{(i)}(x) \tag{6}$$

According to the literature, keeping iterating the two steps, the value and policy sequences will converge to V^* and u^* in the end.

The PI method gives a feasible solution to the HJB equation, but it requires the complete knowledge of the dynamics. Even the integral reinforcement learning (IRL) technique can eliminate the dependence on the internal dynamics f, but the input gain matrix g is still necessary.

III. Off-Policy Approach to the HJB Equation

In this part, a model-free approaches to the HJB equation are presented based on off-policy scheme. Off-policy indicates the system is conducted by a policy that is different to the evaluated one.

Consider an arbitrary control input u_s and apply it on (1) to produce system solutions $(f + gu_s)$. Given an admissible policy $u_0 \in \Psi(\Omega)$, its value function V_0 satisfies (5). By differentiating V_0 along the system solutions we yield

$$\begin{aligned} \dot{V}_0 &= \nabla V_0^T \big(f + g u_0 + g(u_s - u_0)\big) \\ &= \nabla V_0^T g(u_s - u_0) - Q - u_0^T R u_0 \end{aligned} \tag{7}$$

Based on the policy improvement step in (6), the improved policy u_1 has $\nabla V_0^T g = -2 u_1^T R$. After inserting into (7) and employing the integral reinforcement learning, we get

$$\begin{aligned} &V_0(x(t)) - V_0(x(t-T)) \\ &\quad + \int_{t-T}^{t} \big(2 u_1^T R (u_s - u_0) + Q + u_0^T R u_0\big) d\tau = 0 \end{aligned} \tag{8}$$

Now the original two-step policy iteration which requires the system dynamics is converted to a single-step model-free equation. [10] pointed out that if control input u_s satisfies certain PE condition as they provided, the solution to (8) is uniquely determined by u_0's value V_0 and its improved policy u_1 defined by (5) and (6).

Similarly, as for the optimal value V^* and the optimal policy u^*, we plug them into (8) to replace V_0 and u_0, u_1, and the following *complete-model-free HJB* equation is formed

$$\begin{aligned} &V^*(x(t)) - V^*(x(t-T)) \\ &\quad + \int_{t-T}^{t} \big(2 u^{*T} R u_s - u^{*T} R u^* + Q\big) d\tau = 0 \end{aligned} \tag{9}$$

Compared to the original HJB equation, the novel equation is completely independent of any dynamics, neither f nor g.

IV. Complete-Model-Free Off-Policy Learning

In order to approximate value functions and policies, universal approximation property of neural network (NN) is used here. The closed-loop stability of the system is ensured by the following assumption about the control input u_s.

Assumption 1: Suppose the control u_s is a stabilizable input such that the closed-loop system remains in the compact set Ω for any starting state $x(0) \in \Omega_0$, where $\Omega_0 \subseteq \Omega$.

According to the Weirstrass high-order approximation theorem, the smooth value function V is uniformly approximated on Ω by

$$V(x) = W_c^T \phi_c(x) + \varepsilon_c(x) \tag{10}$$

where $W_c \in \mathbb{R}^{K_c}$ represent the ideal coefficients and K_c denotes the hidden neuron number. $\phi_c(x) \in \mathbb{R}^{K_c}$ is the basis function vector. $\varepsilon_c(x) \in \mathbb{R}$ is the approximation error. Similarly, a smooth control policy u is uniformly approximated on Ω by

$$u(x) = W_a^T \phi_a(x) + \varepsilon_a(x) \tag{11}$$

where $W_a \in \mathbb{R}^{K_a \times m}$, $\phi_a(x) \in \mathbb{R}^{K_a}$, and $\varepsilon_a(x) \in \mathbb{R}^m$. K_a denotes the hidden neuron number in the actor. We make the following assumptions about the critic and actor NNs.

Assumption 2: a. The critic and actor NN approximation errors are bounded on the compact Ω so that

$$\|\varepsilon_c\| < \|\varepsilon_c\|_{\max}, \quad \|\varepsilon_a\| < \|\varepsilon_a\|_{\max}$$

b. The critic and actor NN basis functions are bounded so that

$$\|\phi_c(x)\| < \|\phi_c\|_{\max}, \quad \|\phi_a(x)\| < \|\phi_a\|_{\max}$$

A. Complete-model-free policy evaluation

Given an admissible policy u_0, define its value V_0 in the form of (10) and its improved policy u_1 in the form of (11). After inserting the value and action NNs into the complete-model-free equation (8), we get

$$\begin{aligned}\varepsilon_{PE}(t) = {} & W_c^T\big[\phi_c(t) - \phi_c(t-T)\big] \\ & + \int_{t-T}^{t}\Big(2\phi_a^T W_a R(u_s - u_0) + Q + u_0^T R u_0\Big)d\tau \end{aligned} \quad (12)$$

where ε_{PE} is the residual error caused by NN approximation errors, defined by

$$\varepsilon_{PE}(t) \equiv -\varepsilon_c(t) + \varepsilon_c(t-T) - \int_{t-T}^{t} 2\varepsilon_a^T R(u_s - u_0)d\tau$$

A critic and an actor NNs are built to approximate V_0 and u_1. Let $\hat{W}_c$ be the estimate of W_c and $\hat{W}_a$ be the estimate of W_a. A policy evaluation error is formulated

$$\begin{aligned} e_1(t) = {} & \hat{W}_c^T\big[\phi_c(t) - \phi_c(t-T)\big] \\ & + \int_{t-T}^{t}\Big(2\phi_a^T \hat{W}_a R(u_s - u_0) + Q + u_0^T R u_0\Big)d\tau \end{aligned}$$

Using Kronecker product $\otimes$, the above equation is rewritten as

$$\begin{aligned} e_1(t) = {} & \hat{W}_c^T\big[\phi_c(t) - \phi_c(t-T)\big] \\ & + \mathbf{vec}(\hat{W}_a)^T \int_{t-T}^{t} 2\big[R(u_s - u_0)\big] \otimes \phi_a d\tau \\ & + \int_{t-T}^{t}(Q + u_0^T R u_0)d\tau \end{aligned}$$

where $\mathbf{vec}(\cdot)$ indicates the vectorizing operator that transforms a matrix into a vector by stacking all its elements along the column direction. If define

$$\begin{aligned} \sigma_1(t) &= \phi_c(t) - \phi_c(t-T) \\ \eta_1(t) &= \int_{t-T}^{t} 2\big[R(u_s - u_0)\big] \otimes \phi_a d\tau \\ p_1(t) &= \int_{t-T}^{t} Q d\tau \\ s_1(t) &= \int_{t-T}^{t} u_0^T R u_0 d\tau \end{aligned}$$

then we have

$$e_1(t) = \hat{W}_c^T\sigma_1(t) + \mathbf{vec}(\hat{W}_a)^T\eta_1(t) + p_1(t) + s_1(t) \quad (13)$$

Error e_1 represents the reinforcement signal, a continuous-time counterpart of temporal difference (TD) error in RL.

The gradient descent method is applicable to form a continuous adaptation of the estimations $\hat{W}_c$ and $\hat{W}_a$ with the target of minimizing the square error $\frac{1}{2}e_1^2$. However, reviewing the definition of the TD error, it consists of two parts: the current estimations $\hat{W}_c$, $\hat{W}_a$, and the system data $(\sigma_1, \eta_1, p_1, s_1)$. These data, following $(f + gu_s)$, is totally independent of the estimations, making them repeatedly utilizable to form the TD errors and the adaptation. As per the above, we resort to the ideal of experience replay and employ the past data to form the tuning of $\hat{W}_c$ and $\hat{W}_a$. Suppose a set of history data $\{(\sigma_1(t_j), \eta_1(t_j), p_1(t_j), s_1(t_j))\}_{j=1}^{N}$ are stored, where

$$\begin{aligned} \sigma_1(t_j) &= \phi_c(t_j) - \phi_c(t_j - T) \\ \eta_1(t_j) &= \int_{t_j-T}^{t_j} 2\big[R(u_s - u_0)\big] \otimes \phi_a d\tau \\ p_1(t_j) &= \int_{t_j-T}^{t_j} Q d\tau \\ s_1(t_j) &= \int_{t_j-T}^{t_j} u_0^T R u_0 d\tau \end{aligned}$$

and N is the data set size. The experience-replay adaptive law for model-free policy evaluation has

Algorithm 1:

$$\begin{aligned} \dot{\hat{W}}_c = -\frac{\alpha}{N+1}\Bigg\{ & \frac{\sigma_1(t)}{m_1^2(t)}\Big[\hat{W}_c^T\sigma_1(t) + \mathbf{vec}(\hat{W}_a)^T\eta_1(t) \\ & + p_1(t) + s_1(t)\Big] \\ & + \sum_{j=1}^{N}\frac{\sigma_1(t_j)}{m_1^2(t_j)}\Big[\hat{W}_c^T\sigma_1(t_j) + \mathbf{vec}(\hat{W}_a)^T\eta_1(t_j) \\ & + p_1(t_j) + s_1(t_j)\Big]\Bigg\} \end{aligned} \quad (14)$$

$$\begin{aligned} & \mathbf{vec}(\dot{\hat{W}}_a) \\ & = -\frac{\alpha}{N+1}\Bigg\{ \frac{\eta_1(t)}{m_1^2(t)}\Big[\hat{W}_c^T\sigma_1(t) + \mathbf{vec}(\hat{W}_a)^T\eta_1(t) \\ & \qquad + p_1(t) + s_1(t)\Big] \\ & \quad + \sum_{j=1}^{N}\frac{\eta_1(t_j)}{m_1^2(t_j)}\Big[\hat{W}_c^T\sigma_1(t_j) + \mathbf{vec}(\hat{W}_a)^T\eta_1(t_j) \\ & \qquad + p_1(t_j) + s_1(t_j)\Big]\Bigg\} \end{aligned} \quad (15)$$

where m_1 is defined as $m_1(t) = \big(\sigma_1^T(t)\sigma_1(t) + \eta_1^T(t)\eta_1(t) + 1\big)^{\frac{1}{2}}$ for normalization.

Theorem 1: Given an admissible policy u_0, let W_c be the ideal critic coefficients of V_0, W_a be the ideal actor coefficients of u_1, and $\hat{W}_c$ and $\hat{W}_a$ be the estimations of W_c and W_a. Let $\rho_1 \equiv [\sigma_1^T, \eta_1^T]^T$ and assume signal $\bar{\rho}_1 \equiv \rho_1/m_1$ is persistently exciting. Let Assumptions 1 and 2 hold. Under the tuning laws provided by (14) and (15), the errors $\tilde{W}_c = W_c - \hat{W}_c$ and

$\tilde{W}_a = W_a - \hat{W}_a$ converge exponentially to a residual set around zero.

Proof: Define a Lyapunov candidate

$$L_1 = \frac{1}{2}\tilde{W}_c^T \alpha^{-1} \tilde{W}_c + \frac{1}{2}\mathbf{vec}(\tilde{W}_a)^T \alpha^{-1} \mathbf{vec}(\tilde{W}_a)$$

Its time derivative has

$$\dot{L}_1 = \tilde{W}_c^T \alpha^{-1} \dot{\tilde{W}}_c + \mathbf{vec}(\tilde{W}_a)^T \alpha^{-1} \mathbf{vec}(\dot{\tilde{W}}_a)$$

After substituting (12) into (13), we have

$$e_1(t) = -\tilde{W}_c^T \sigma_1(t) - \mathbf{vec}(\tilde{W}_a)^T \eta_1(t) + \varepsilon_{PE}(t)$$

Note that the above relationship holds for any time index t_j. Let $\tilde{Z}_1 = \left[\tilde{W}_c^T, \mathbf{vec}(\tilde{W}_a)^T\right]^T$ and then

$$\dot{\tilde{W}}_c = \frac{\alpha}{N+1}\left\{ \frac{\sigma_1(t)}{m_1^2(t)}\left[-\tilde{Z}_1^T \rho_1(t) + \varepsilon_{PE}(t)\right] + \sum_{j=1}^{N} \frac{\sigma_1(t_j)}{m_1^2(t_j)}\left[-\tilde{Z}_1^T \rho_1(t_j) + \varepsilon_{PE}(t_j)\right] \right\}$$

$$\mathbf{vec}(\dot{\tilde{W}}_a) = \frac{\alpha}{N+1}\left\{ \frac{\eta_1(t)}{m_1^2(t)}\left[-\tilde{Z}_1^T \rho_1(t) + \varepsilon_{PE}(t)\right] + \sum_{j=1}^{N} \frac{\eta_1(t_j)}{m_1^2(t_j)}\left[-\tilde{Z}_1^T \rho_1(t_j) + \varepsilon_{PE}(t_j)\right] \right\}$$

After inserting into $\dot{L}_1$, we yield

$$\dot{L}_1 \leq -\frac{1}{N+1}\lambda_{\min}(\bar{H}_1)\|\tilde{Z}_1\|^2 + \|\varepsilon_{PE}\|_{\max}\|\tilde{Z}_1\|$$

where

$$\bar{H}_1 \equiv \bar{\rho}_1(t)\bar{\rho}_1^T(t) + \sum_{j=1}^{N} \bar{\rho}_1(t_j)\bar{\rho}_1^T(t_j)$$

Whenever $\|\tilde{Z}_1\| > \frac{(N+1)\|\varepsilon_{PE}\|_{\max}}{\lambda_{\min}(\bar{H}_1)}$, $\dot{L}_1$ is negative. The proof is complete. ■

From the analysis in Theorem 1, the convergence rate is determined by $\lambda_{\min}(\bar{H}_1)$ which can be improved by maximizing the minimum eigenvalue of $H_1 \equiv \sum_{j=1}^{N} \bar{\rho}_1(t_j)\bar{\rho}_1^T(t_j)$. So during the learning process, we can continually update the history set through replacing some old data by the latest one when such replacement results in an increase in the minimum eigenvalue of H_1.

B. Model-free adaptive optimal learning

Now back to the major interest in this paper, an adaptive optimal algorithm is presented here. Let the optimal value V^* and the optimal policy u^* be expressed by NNs in the form of (10) and (11). Combining with the complete-model-free HJB equation (9) we yield

$$\varepsilon_{HJB}(t) = W_c^T[\phi_c(t) - \phi_c(t-T)] + \int_{t-T}^{t} \left(2\phi_a^T W_a R u_s - \phi_a W_a R W_a^T \phi_a + Q\right) d\tau \quad (16)$$

where

$$\varepsilon_{HJB}(t) \equiv -\varepsilon_c(t) + \varepsilon_c(t-T) + \int_{t-T}^{t} \left(-2\varepsilon_a^T R u_s + 2\varepsilon_a^T R W_a^T \phi_a + \varepsilon_a^T R \varepsilon_a\right) d\tau$$

and ε_{HJB} is bounded as the NN basis functions and approximation errors are bounded.

To identify the unknown W_c and W_a, define the critic and actor NNs to output the approximation

$$\hat{V}^*(x) = \hat{W}_c \phi_c(x)$$
$$\hat{u}^*(x) = \hat{W}_a \phi_a(x)$$

Consequently an HJB error is formulated

$$e_2(t) = \hat{W}_c^T[\phi_c(t) - \phi_c(t-T)] + \int_{t-T}^{t} \left(2\phi_a^T \hat{W}_a R u_s - \phi_a \hat{W}_a R \hat{W}_a^T \phi_a + Q\right) d\tau \quad (17)$$

Under the representation of Kronecker product, define

$$\begin{aligned} \sigma_2(t) &= \phi_c(t) - \phi_c(t-T) \\ \mu_2(t) &= \int_{t-T}^{t} (R u_s) \otimes \phi_a d\tau \\ D_2(t) &= \int_{t-T}^{t} (\phi_a^T \otimes R) \otimes \phi_a d\tau \\ p_2(t) &= \int_{t-T}^{t} Q d\tau \end{aligned} \quad (18)$$

and let

$$\eta_2(t) = 2\mu_2(t) - 2D_2(t)\mathbf{vec}(\hat{W}_a)$$

Then e_2 is rewritten as

$$e_2(t) = \hat{W}_c^T \sigma_2(t) + \mathbf{vec}(\hat{W}_a)^T \eta_2(t) + \mathbf{vec}(\hat{W}_a)^T D_2(t) \mathbf{vec}(\hat{W}_a) + p_2(t)$$

Besides, under the current estimation $\hat{W}_c$ and $\hat{W}_a$, for arbitrary moment t_j the past data form the errors

$$e_2(t_j) = \hat{W}_c^T \sigma_2(t_j) + \mathbf{vec}(\hat{W}_a)^T \eta_2(t_j) + \mathbf{vec}(\hat{W}_a)^T D_2(t_j) \mathbf{vec}(\hat{W}_a) + p_2(t_j)$$

where

$$\eta_2(t_j) = 2\mu_2(t_j) - 2D_2(t_j)\mathbf{vec}(\hat{W}_a)$$

and $\sigma_2(t_j)$, $\mu_2(t_j)$, $D_2(t_j)$, $p_2(t_j)$ are defined in the same way as (18) but with time index t_j.

The complete-model-free adaptive optimal algorithm with experience replay is designed on the basis of a history data set $\{(\sigma_2(t_j), \mu_2(t_j), D_2(t_j), p_2(t_j))\}_{j=1}^{N}$ with the following

tuning

Algorithm 2:

$$\dot{\hat{W}}_c = -\frac{\alpha}{N+1}\left\{\frac{\sigma_2(t)}{m_2^2(t)}\Big[\hat{W}_c^T\sigma_2(t)+\mathbf{vec}(\hat{W}_a)^T\eta_2(t) + \mathbf{vec}(\hat{W}_a)^T D_2(t)\mathbf{vec}(\hat{W}_a)+p_2(t)\Big] + \sum_{j=1}^{N}\frac{\sigma_2(t_j)}{m_2^2(t_j)}\Big[\hat{W}_c^T\sigma_2(t_j)+\mathbf{vec}(\hat{W}_a)^T\eta_2(t_j) + \mathbf{vec}(\hat{W}_a)^T D_2(t_j)\mathbf{vec}(\hat{W}_a)+p_2(t_j)\Big]\right\} \tag{19}$$

$$\mathbf{vec}(\dot{\hat{W}}_a) = -\frac{\alpha}{N+1}\left\{\frac{\eta_2(t)}{m_2^2(t)}\Big[\hat{W}_c^T\sigma_2(t)+\mathbf{vec}(\hat{W}_a)^T\eta_2(t) + \mathbf{vec}(\hat{W}_a)^T D_2(t)\mathbf{vec}(\hat{W}_a)+p_2(t)\Big] + \sum_{j=1}^{N}\frac{\eta_2(t_j)}{m_2^2(t_j)}\Big[\hat{W}_c^T\sigma_2(t_j)+\mathbf{vec}(\hat{W}_a)^T\eta_2(t_j) + \mathbf{vec}(\hat{W}_a)^T D_2(t_j)\mathbf{vec}(\hat{W}_a)+p_2(t_j)\Big]\right\} \tag{20}$$

In this case we have $m_2(t) = \left(\sigma_2^T(t)\sigma_2(t)+\eta_2^T(t)\eta_2(t)+1\right)^{\frac{1}{2}}$.

Theorem 2: Let W_c be the ideal critic coefficients of V^*, W_a be the ideal actor coefficients of u^*, and $\hat{W}_c$ and $\hat{W}_a$ be the estimations of W_c and W_a. Let $\rho_2 \equiv [\sigma_2^T, \eta_2^T]^T$ and assume signal $\bar{\rho}_2 \equiv \rho_2/m_2$ is persistently exciting. Let Assumptions 1 and 2 hold. If the integral interval T satisfies the requirement as described in the sequel and NN weights are appropriately initialized, the errors $\tilde{W}_c = W_c - \hat{W}_c$ and $\tilde{W}_a = W_a - \hat{W}_a$ converge exponentially to a residual set in the neighbor of zero under the tuning laws provided by Algorithm 2.

Proof: The proof is similar to the Theorem 1 so we omit the detailed proving process. Define the Lyapunov candidate $L_2 = \frac{1}{2}\tilde{W}_c^T\alpha^{-1}\tilde{W}_c + \frac{1}{2}\mathbf{vec}(\tilde{W}_a)^T\alpha^{-1}\mathbf{vec}(\tilde{W}_a)$ and let $\tilde{Z}_2 = \left[\tilde{W}_c^T, \mathbf{vec}(\tilde{W}_a)^T\right]^T$. Then the Lyapunov time derivative has

$$\dot{L}_2 \le -\frac{1-\varepsilon_T}{N+1}\lambda_{\min}(\bar{H}_2)\left\|\tilde{Z}_2\right\|^2 + \left\|\tilde{Z}_2\right\|\left\|\varepsilon_{HJB}\right\|_{\max}$$

where ε_T is a small constant and

$$\bar{H}_2 \equiv \bar{\rho}_2(t)\bar{\rho}_2^T(t) + \sum_{j=1}^{N}\bar{\rho}_2(t_j)\bar{\rho}_2^T(t_j)$$

So the estimation error $\tilde{Z}_2$ converges exponentially to the residual set $\|\tilde{Z}_2\| < \frac{(N+1)\|\varepsilon_{HJB}\|_{\max}}{\lambda_{\min}(\bar{H}_2)}$. The proof is complete. ■

During the adaptive optimal learning under (19) and (20), we can accelerate the process by letting one of the old data in the history set be replaced by the latest $(\sigma_2(t), \mu_2(t), D(t), p_2(t))$ if the minimum eigenvalue of $H_2 = \sum_{j=1}^{N}\bar{\rho}_2(t_j)\bar{\rho}_2^T(t_j)$ is maximized.

In order to identify the unknown parameters, some signals are required to remain persistently exciting. An effective approach to such condition is adding small exploratory sinusoids consisting of various frequencies into the stabilizable control input u_s.

The convergence in Theorem 2 declares that an appropriate initialization of $\hat{W}_c$ and $\hat{W}_a$ is necessary as the proof requires $\tilde{Z}_2$ to be bounded. A policy evaluation process by Algorithm 1 prior to Algorithm 2 is reasonable so that the critic and actor NNs are learned to a set of satisfying weights. After that, the adaptive optimal learning is applied. As for some simply cases, like self-stable problems, such prior policy evaluation is dispensable as long as the NN weights are initialized close to zero.

V. Experiment

The nonlinear system for simulation is the rotational/translational actuator (RTAC) nonlinear bench-mark problem [8] whose dynamics can be written as

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{-x_1+\xi\frac{x_4^2}{100}\sin\frac{x_3}{100}}{1-\xi^2\cos^2\frac{x_3}{100}} \\ x_4 \\ \frac{\xi\cos\frac{x_3}{100}(x_1-\xi\frac{x_4^2}{100}\sin\frac{x_3}{100})}{1-\xi^2\cos^2\frac{x_3}{100}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-\xi\cos\frac{x_3}{100}}{1-\xi^2\cos^2\frac{x_3}{100}} \\ 0 \\ \frac{1}{1-\xi^2\cos^2\frac{x_3}{100}} \end{bmatrix} u$$

where $\xi = 0.2$. The cost function is selected as $Q(x) = x^Tx$ and $R = I_{4\times4}$. The critic and actor basis functions are selected as

$$\phi_c(x) = \big[x_1^2, x_1x_2, x_1x_3, x_1x_4, x_2^2, x_2x_3, x_2x_4, x_3^2, x_3x_4, x_4^2, x_1^4/100, x_2^4/100, x_3^4/100, x_4^4/100\big]^T$$

$$\phi_a(x) = \big[x_1, x_2, x_3, x_4, x_1^3/100, x_2^3/100, x_3^3/100, x_4^3/100\big]^T$$

which is sufficient for the problem. As the system is unstable and the problem is complicated, a policy evaluation phase is necessary. The given admissible policy is $u_0(x) = 0.2x_1 - 0.3x_3 - 0.8x_4$, and the online control input is $u_s(t) = u_0(t) + e(t)$ where e is the exploratory signal. After the convergence of the policy evaluation phase, the adaptive optimal tuning is applied sequentially. The learning rate α is set to 100 and the initial state is $[20, -20, 20, -20]^T$. The size of the experience set selects 25.

The result of the algorithm is presented in Figs. 1 and 2. The vertical red lines in Fig. 2 represent the moments when the replacement in history data set happens. Note that the minimum eigenvalue is significantly improved with our update criterion, which is conducive to the convergence of the critic and actor. The final NNs weights are

$$\hat{W}_c = [13.2231, -0.9938, 0.1575, -2.5530, 13.3389, 2.8515, 5.4878, 1.8679, 2.4937, 2.3408, 0.0025, -0.0004, -0.0004, 0.0106]^T$$

$$\hat{W}_a = [1.2278, -0.0796, -0.9997, -1.8687, -0.0049, -0.0029, -0.0002, 0.0054]^T$$

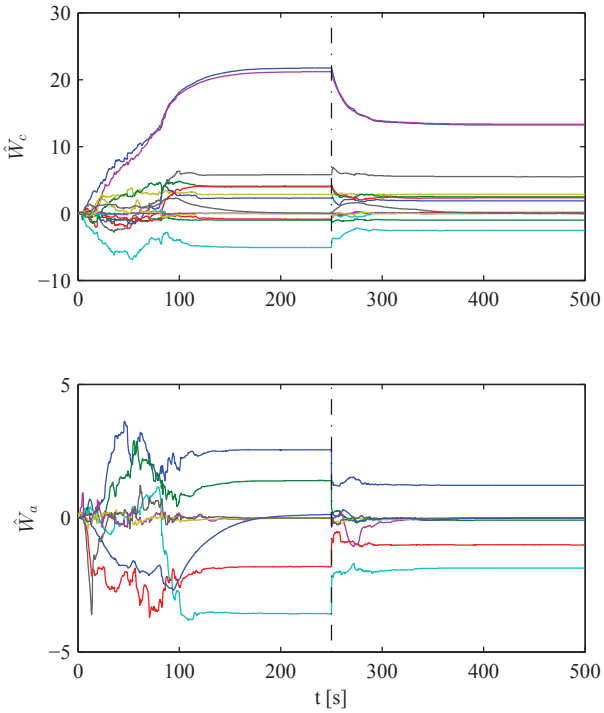

Fig. 1. NN weights in the critic and actor.

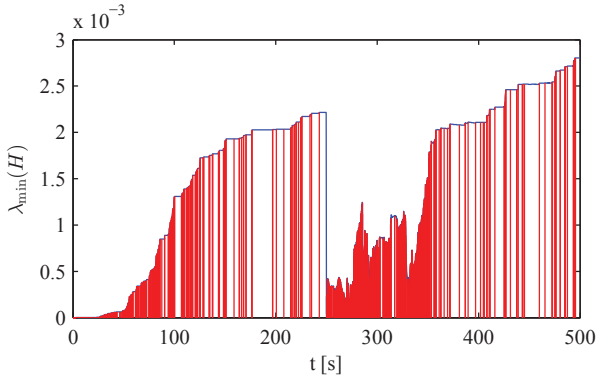

Fig. 2. Trajectory of minimum eigenvalue of history data set.

The state trajectories by the converged controller are depicted in Fig. 3.

VI. Conclusion

The continuous-time nonlinear optimal control problem is considered in this paper. Based on off-policy policy iteration, a novel online real-time adaptive optimal algorithms are devised. It is a completely model-free algorithm in contrast to the traditional model-based or partial-model-based ones. In order to make high utilization of online data, experience replay is employed to integrate the past data into the tuning. Experimental results verify the effectiveness of experience replay in the fact that the convergence rate is significantly improved.

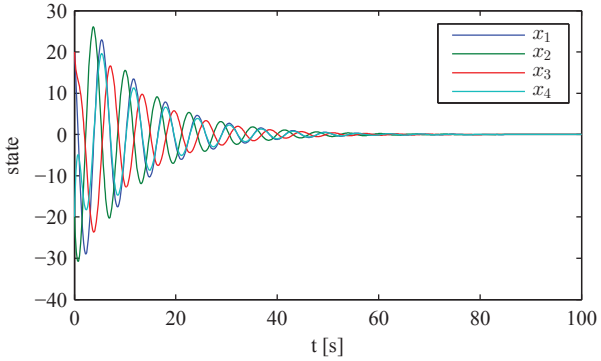

Fig. 3. State trajectories by the convergent controller.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[2] D. Zhao and Y. Zhu, "MEC–a near-optimal online reinforcement learning algorithm for continuous deterministic systems," *IEEE Trans. on Neural Netw. Learning Syst.*, vol. 26, no. 2, pp. 346–356, Feb 2015.

[3] Y. Zhu and D. Zhao, "A data-based online reinforcement learning algorithm satisfying probably approximately correct principle," *Neural Computing and Applications*, vol. 26, no. 4, pp. 775–787, 2015.

[4] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477 – 484, 2009.

[5] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237 – 246, 2009.

[6] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699 – 2704, 2012.

[7] Z.-P. Jiang and Y. Jiang, "Robust adaptive dynamic programming for linear and nonlinear systems: An overview," *European Journal of Control*, vol. 19, no. 5, pp. 417 – 425, 2013.

[8] B. Luo, H.-N. Wu, T. Huang, and D. Liu, "Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design," *Automatica*, vol. 50, no. 12, pp. 3281 – 3290, 2014.

[9] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624 – 2632, 2014.

[10] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 5, pp. 916–932, 2015.

[11] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[12] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82 – 92, 2013.

[13] H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct 2013.

[14] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2686–2710, 2014.

[15] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193 – 202, 2014.