Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Online Adaptive Critic Robust Control of Discrete-Time Nonlinear Systems With Unknown Dynamics ⋆

Hao Fu *,** **Xin Chen** *,** **Min Wu** *,**

* *School of Automation, China University of Geosciences, Wuhan, 430074, China (e-mail: chenxin@cug.edu.cn).*
** *Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, 430074, China*

Abstract: This paper concerns the optimal model reference adaptive control problem for unknown discrete-time nonlinear systems. For such problem, it is challenging to improve online learning efficiency and guaranteeing robustness to the uncertainty. To this end, we develop an online adaptive critic robust control method. In this method, a critic network and a new supervised action network are constructed to not only improve the real-time learning efficiency, but also obtain the optimal control performance. By combining the designed compensation control term, robustness is further guaranteed by compensating the uncertainty. The comparative simulation study is conducted to show the superiority of our developed method.

Keywords: Approximate dynamic programming (ADP), unknown nonlinear systems, neural network (NN), supervised learning, model reference adaptive control (MRAC), robust control.

1. INTRODUCTION

During past several decades, reinforcement learning (RL) has gained a great deal of research attention in the artificial intelligence community. In the control system society, approximate/adaptive dynamic programming (ADP) Werbos (1992) (also called adaptive critic design), which combines RL and the adaptive control, has been employed to address the optimal regulation issue firstly via the action-critic network framework. Fruitful results Chen et al. (2019); Lian et al. (2016); Ha et al. (2018); Wang et al. (2020); Pang & Jiang (2019); Si et al. (2001) have been reported on ADP in recent years.

In the aforementioned results, an online ADP method Si et al. (2001) has been developed with no requirement of system dynamics. Convergence of this algorithm has been analyzed via the Lyapunov extension theorem Liu et al. (2012). On this basis, He *et al*. He et al. (2012) have further proposed a new ADP framework with an additional reference/goal network integrated into the action-critic network. What's more, this algorithm has witnessed extensive studies in terms of the optimal tracking control Yang et al. (2009); Ni et al. (2013); Mu et al. (2017) as well.

The model reference adaptive control (MRAC) aims at enforcing the controlled systems to track the desired reference model rather than a tracking trajectory. Then, the closed-loop control system has the characteristics of the reference model. The optimal MRAC is far more worth deserving investigation than those tracking control. From the state-of-the-art developments of this investigation, only Radac et al. (2017, 2018); Fu et al. (2017); Wang et al. (2018) have developed the ADP-based optimal MRAC approach.

As existence of the reference input in MRAC, there invariably exists a feedforward control term dependent on input dynamics of the systems. The input dynamics needs to be derived via identification. To obviate this requirement, change of the reference input in Radac et al. (2017, 2018) is ignored during the learning process. Fu et al. (2017); Wang et al. (2018) don't consider the uncertainty resulting from the identification error. As such, it is still a challenge to investigate the ADP-based MRAC method with robustness to such uncertainty.

On the other hand, in the beginning of training phase, this online ADP method is easy to cause inefficiency or high failure rate with unknown dynamics Zhao et al. (2013); Fathinezhad et al. (2009). The inefficiency or high failure rate is an unacceptable and fatal risk in the real-time control.

Motivated by above discussions, we develop an online adaptive critic robust control method for discrete-time nonlinear systems with unknown dynamics. This method ensures that closed-loop control systems have robustness to uncertainty and high-efficiency learning performance.

The main contributions of this study include the following two aspects.

(1) In contrast to the existing online ADP methods Yang et al. (2009); Ni et al. (2013); Mu et al. (2017), our developed control method greatly reduces the failure rate and improves the learning efficiency via a critic network and a new supervised action network.
(2) Unlike Radac et al. (2017, 2018); Fu et al. (2017); Wang et al. (2018), our developed control method well guarantees robustness to the uncertainties resulting from iden-

⋆ This work was supported in part by the National Natural Science Foundation of China under Grant 61873248, in part by the Hubei Provincial Natural Science Foundation of China under Grant 2017CFA030 and Grant 2015CFA010, and in part by the 111 project of China under Grant B17040.

tification and the exterior disturbance by introducing the compensation control into the learning process.

The outline of this paper is arranged as follows. The problem description is stated in Section 2. In Section 3, the online adaptive critic robust control method is given. In Section 4 provides the comparative simulation. In Section 5, the conclusions are stated.

2. PROBLEM FORMULATIONS

Consider the following discrete-time nonlinear system:

$$\begin{cases} x_i(t+1) = x_{i+1}(t), i = 1,2,\dots,n-1 \\ x_n(t+1) = f(x(t)) + g(x(t))u(t) + d(t), \end{cases} \tag{1}$$

in which $x(t) = [x_1^T(t), x_2^T(t), \dots, x_n^T(t)]^T \in R^{nm}$ denotes the state with $x_i(t) \in R^m$, $f : R^{nm} \to R^m$ and $g : R^{nm} \to R^{m\times m}$ are unknown smooth nonlinear functions, $u(t) \in R^m$ represents the control input, and $d(t) \in R^m$ denotes an unknown persistent disturbance. Note that, under the full-state feedback linearization, the general nonlinear systems can be converted to the formation (1) via the coordinate transformation.

Assumption 1: The nonlinear function $g(t)$ is always bounded and nonsingular for $\forall x(t)$.

Define a reference model as

$$\begin{cases} x_{ri}(t+1) = x_{ri+1}(t), i = 1,2,\dots,n-1 \\ x_{rn}(t+1) = A_r x_r(t) + B_r u_r(t), \end{cases} \tag{2}$$

where $x_r(t) = [x_{r1}^T(t), x_{r2}^T(t), \dots, x_{rn}^T(t)]^T \in R^{nm}$ denotes the reference state with $x_{ri}(t) \in R^m$, $A_r \in R^{m\times nm}$ and $B_r \in R^{m\times m}$ represent the constant matrices of the reference model, $u_r(t)$ is the reference control input. Here, $x_r(t)$ and $u_r(t)$ are all assumed to be bounded.

The objective of this paper is to enable the system (1) to track the reference model (2) on behavior with optimum via designing an optimal control law $u(t)$. Subtracting (2) from (1) yields the model reference tracking error dynamics

$$\begin{cases} e_i(t+1) = e_{i+1}(t), i = 1,2,\dots,n-1 \\ e_n(t+1) = f(t) + g(t)u(t) + d(t) - A_r x_r(t) - B_r u_r(t), \end{cases} \tag{3}$$

where $e(t) = x(t) - x_r(t)$ denotes the model reference tracking error with $e_i(t) = x_i(t) - x_{ri}(t)$.

To realize the optimum, it is needed to minimize the performance index function or cost function

$$J(t) = \sum_{k=t}^{\infty} \gamma^{k-t} r(k), \tag{4}$$

in which γ is a discount factor, $r(t) = e^T(t)Qe(t) + u^T(t)Ru(t)$ is defined as the utility function or reward with the positive symmetric matrices Q and R.

In accordance with Bellman's optimality principle, the optimal cost function $J^*(t)$ satisfies the following Bellman equation:

$$J^*(t) = \min_{u(t)} \{r(t) + \gamma J^*(t+1)\}. \tag{5}$$

Due to unknown dynamics for (1), it is difficult to solve the Bellman equation (5). To overcome this difficulty, the ADP-based MRAC methods Radac et al. (2017, 2018); Fu et al. (2017); Wang et al. (2018) have been proposed. But, Radac et al. (2017, 2018) have no real-time control performance. In Fu et al. (2017); Wang et al. (2018), the system uncertainty resulting from identification is not considered. On the other hand, inefficiency or the high failure rate is always existent in the online ADP methods Yang et al. (2009); Ni et al. (2013); Mu et al. (2017), which is an unacceptable and fatal risk in the real-time control.

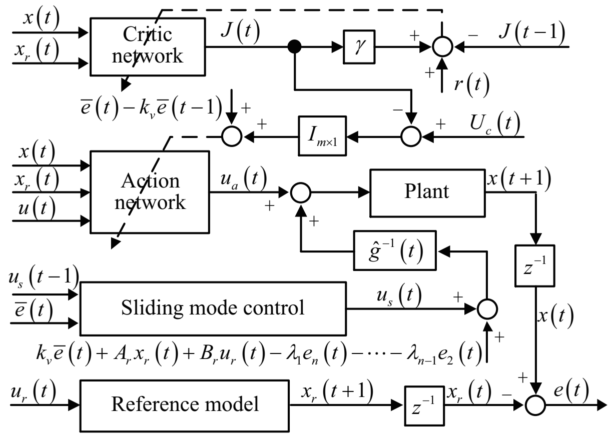

Fig. 1. Online adaptive critic robust control structure diagram.

3. ADAPTIVE CRITIC ROBUST CONTROL

In this section, an online adaptive critic robust control method is developed to achieve robustness to the uncertainty and learning efficiency. Its control structure diagram is depicted in Fig. 1.

Define a filtered model reference tracking error as

$$\bar{e}(t) = e_n(t) + \lambda_1 e_{n-1}(t) + \dots + \lambda_{n-1} e_1(t), \tag{6}$$

where $\lambda_1, \dots, \lambda_{n-1}$ are constants such that $|z^{n-1} + \lambda_1 z^{n-2} + \dots + \lambda_{n-1}|$ is stable. Then, the filtered model reference tracking error dynamics can be formulated as

$$\begin{aligned} \bar{e}(t+1) =& f(t) + g(t)u(t) + d(t) - A_r x_r(t) - B_r u_r(t) \\ &+ \lambda_1 e_n(t) + \dots + \lambda_{n-1} e_2(t). \end{aligned} \tag{7}$$

An adaptive critic robust control law is designed as

$$\begin{aligned} u(t) =& \hat{g}^{-1}(t)(u_s(t) + k_v \bar{e}(t) + A_r x_r(t) + B_r u_r(t) \\ &- \lambda_1 e_n(t) - \dots - \lambda_{n-1} e_2(t)) + u_a(t), \end{aligned} \tag{8}$$

where $k_v \in R^{m\times m}$ is the gain matrix, $u_a(t)$ denotes a neural network (NN) control term, $u_s(t)$ represents a compensation control term, and $\hat{g}(t)$ is the estimation of $g(t)$. Note that, $\hat{g}(t)$ is usually obtained by the model identification method Zhao et al. (2016); Jiang et al. (2018). According to Assumption 1 and the results of Wang et al. (2002), it is deduced that $\hat{g}(t)$ is also bounded away from singularity.

A desirable value of $u(t)$ is given by

$$\begin{aligned} u_d(t) =& g^{-1}(t)(u_s(t) + k_v \bar{e}(t) - f(t) - d(t) + A_r x_r(t) + B_r u_r(t) \\ &- \lambda_1 e_n(t) - \dots - \lambda_{n-1} e_2(t)). \end{aligned} \tag{9}$$

Using (9) and substituting (8) into (7) yields

$$\begin{aligned} \bar{e}(t+1) =& k_v \bar{e}(t) + g(t)(u(t) - u_d(t)) \\ =& k_v \bar{e}(t) + f_1(t) + g(t)u_a(t) + u_s(t) + d_1(t), \end{aligned} \tag{10}$$

where $f_1(t) = f(t) + (k_v - \lambda_1 I_m)x_n(t) + (k_v\lambda_1 - \lambda_2 I_m)x_{n-1}(t) + \dots + (k_v\lambda_{n-2} - \lambda_{n-1} I_m)x_2(t) + k_v\lambda_{n-1}x_1(t)$, $d_1(t) = g(t)(\hat{g}^{-1}(t) - g^{-1}(t))(u_s(t) + A_r x_r(t) + B_r u_r(t) + (\lambda_1 I_m - k_v)x_{rn}(t) + (\lambda_2 I_m - k_v\lambda_1)x_{rn-1}(t) + \dots + (\lambda_{n-1} I_m - k_v\lambda_{n-2})x_{r2}(t) - k_v\lambda_{n-1}x_{r1}(t)) + d(t)$, and $I_m \in R^{m\times m}$ is an identity matrix. According to the results of Fu et al. (2018), it is inferred from Assumption 1 that $d_1(t)$ is bounded.

For requirement of the optimal control, the critic network and the supervised action network are constructed as follows.

Since it is intractable to acquire the analytical resolution of $J^*(t)$ by solving (5), NN is employed to near the cost function $J(t)$ as follows

$$J(t) = w_c^{*T}(t)\phi_c(v_c^{*T}(t)z_c(t)) + \varepsilon_c, \tag{11}$$

where $z_c(t) = [e^T(t), u_a^T(t)]^T$ denotes the input of the critic network with h_c neurons of the hidden layer, $\phi_c(\cdot)$ represents the active function of the critic network, $w_c^*(t) \in R^{h_c \times 1}$ and $v_c^*(t) \in R^{(nm+m)\times h_c}$ denote the ideal weights, and ε_c is the critic network approximation error.

Similarly, since $w_c^{*T}(t)$ and $v_c^{*T}(t)$ cannot be obtained directly, the estimation of $J(t)$ is constructed as

$$\hat{J}(t) = w_c^T(t)\phi_c(v_c^T(t)z_c(t)), \tag{12}$$

where $w_c^T(t)$ and $v_c^T(t)$ are the estimations of $w_c^{*T}(t)$ and $v_c^{*T}(t)$.

Due to unknown dynamics for (10), the supervised action network $u_a(t)$ has an NN representation as

$$u_a(t) = \phi_{a2}(w_a^{*T}(t)\phi_{a1}(v_a^{*T}(t)z_a(t))) + \varepsilon_a, \tag{13}$$

where $z_a(t) = x(t)$ denotes the input of the supervised action network with h_a neurons of the hidden layer, $\phi_{a1}(\cdot)$ and $\phi_{a2}(\cdot)$ represent the active functions, $w_a^*(t) \in R^{h_a \times m}$ and $v_a^*(t) \in R^{nm \times h_a}$ denote the ideal weights, and ε_a is the supervised action network approximation error.

Since the ideal weights $w_a^{*T}(t)$ and $v_a^{*T}(t)$ cannot be obtained directly, the actual control term $u_a(t)$ is constructed as

$$u_a(t) = \phi_{a2}(w_a^T(t)\phi_{a1}(v_a^T(t)z_a(t))), \tag{14}$$

where $w_a^T(t)$ and $v_a^T(t)$ are the estimations of $w_a^{*T}(t)$ and $v_a^{*T}(t)$. For simplicity, $\phi_c(t)$, $\phi_{a1}(t)$, and $\phi_{a2}(t)$ are used to represent $\phi_c(v_c^T(t)z_c(t))$, $\phi_{a1}(v_a^T(t)z_a(t))$, and $\phi_{a2}(w_a^T(t)\phi_{a1}(v_a^T(t)z_a(t)))$, respectively.

The prediction error of the supervised action network is represented as

$$e_a(t) = (\hat{J}(t) - U_c)I_{m\times 1} + f_1(t) + g(t)u_a(t), \tag{15}$$

where $U_c = 0$ is the ultimate cost objective and $I_{m\times 1} \in R^{m\times 1}$ is a matrix whose elements are all 1.

Remark 1: There are two targets in the supervised action network design. one is to minimize the error between the cost function estimation $\hat{J}(t)$ and the ultimate cost objective U_c. Its motivation is that the cost function estimation $\hat{J}(t)$ approximates the optimal cost function $J^*(t)$. Another target is to minimize the error between the output of the action network and $g^{-1}(t)f(t)$, which is similar to the supervised learning or the adaptive NN control.

Due to no prior knowledge of $f(t)$ and $g(t)$, by using (10), (15) is reformulated as

$$e_a(t) = (\hat{J}(t) - U_c)I_{m\times 1} + \bar{e}(t+1) - k_v\bar{e}(t) - u_s(t). \tag{16}$$

Define its objective function as

$$E_a(t) = \tfrac{1}{2}e_a^T(t)e_a(t). \tag{17}$$

The weights of the supervised action network are updated by

$$\Delta w_a(t) = -\eta_a\phi_{a1}(t)e_a^T(t)w_{ac}(t)\mathrm{diag}(\phi_{a2}^{'}(t)), \tag{18a}$$

$$\Delta v_a(t) = -\eta_a z_a(t)e_a^T(t)w_{ac}(t)\mathrm{diag}(\phi_{a2}^{'}(t)w_a^T(t)\mathrm{diag}(\phi_{a1}^{'}(t)), \tag{18b}$$

where η_a is the learning rate, $\mathrm{diag}(\cdot)$ is the diagonalized operator, $\phi_{a1}^{'}(t)$ and $\phi_{a2}^{'}(t)$ respectively represent the derivative of $\phi_{a1}(t)$ and $\phi_{a2}(t)$, and $w_{ac}(t)$ is defined as

$$w_{ac}(t) = \alpha_c I_{m\times 1}w_c^T(t)\mathrm{diag}(\phi_c^{'}(t))v_{c2}^T(t) + (1-\alpha_c)g(t), \tag{19}$$

with which α_c will be designed in details later, $\phi_c^{'}(t)$ represents the derivative of $\phi_c(t)$, the matrix $v_{c2}(t) \in R^{m\times h_c}$ satisfies $v_c(t) = [v_{c1}^T(t), v_{c2}^T(t)]^T$ with $v_{c1}(t) \in R^{nm\times h_c}$.

In the critic network, via the Bellman equation (5), the prediction error can be represented as

$$e_c(t) = \gamma\hat{J}(t) - \hat{J}(t-1) + r(t). \tag{20}$$

Define its objective function as

$$E_c(t) = \tfrac{\alpha_c}{2}e_c(t)e_c(t). \tag{21}$$

By using the gradient descent algorithm, the weights of the critic network are updated by

$$\Delta w_c(t) = -\eta_c\alpha_c\gamma e_c(t)\phi_c(t), \tag{22a}$$

$$\Delta v_c(t) = -\eta_c\alpha_c\gamma e_c(t)z_c(t)w_c^T(t)\mathrm{diag}(\phi_c^{'}(t)). \tag{22b}$$

Design a learning schedule factor α_c as

$$\alpha_c = \begin{cases} 1, \text{ if } \dfrac{1}{N_\alpha}\displaystyle\sum_{k=t-N_\alpha+1}^{t}\|\bar{e}(t)\| < \varepsilon_\alpha, \\ 0, \text{ if } \dfrac{1}{N_\alpha}\displaystyle\sum_{k=t-N_\alpha+1}^{t}\|\bar{e}(t)\| \geq \varepsilon_\alpha, \end{cases} \tag{23}$$

where $\varepsilon_\alpha > 0$ is a design constant and N_α is a positive integer.

It is worth pointing out that the traditional online action-critic framework is easy to lead to some inefficiency for the real-time control problem. Specifically, in the beginning of online training phase, the state of the system (1) may be far away from the reference state, which results in the training failure risk. This case can be viewed as $\frac{1}{N_\alpha}\sum_{k=t-N_\alpha+1}^{t}\|\bar{e}(t)\| \geq \varepsilon_\alpha$, i.e. $\alpha = 0$. The supervised action network guides the system state back to the neighbor of the reference state. Once $\frac{1}{N_\alpha}\sum_{k=t-N_\alpha+1}^{t}\|\bar{e}(t)\| < \varepsilon_\alpha$ holds, the online adaptive critic learning works to further derive the optimal control policy. It is obvious that the high failure risk is avoided in the beginning of the online training phase.

It can be concluded that the learning process is convergent and $e(t)$ is uniformly ultimately bounded (UUB) and its boundary relies on $d_1(t)$, whose proof is omitted here for saving the space. Then, we can deduce from the UUB property that $\bar{e}(t)$ and $\bar{e}(t) - k_v\bar{e}(t-1)$ are also bounded. Without loss of generality, let

$$\|\bar{e}(t+1) - k_v\bar{e}(t)\| \leq \delta_e, \tag{24}$$

where $\delta_e > 0$.

From (10), we have $\|f_1(t) + g(t)u_a(t) + d_1(t)\| \leq \delta_e$. Let

$$d_2(t) = f_1(t) + g(t)u_a(t) + d_1(t). \tag{25}$$

Then, we get

$$\bar{e}(t+1) = k_v\bar{e}(t) + u_s(t) + d_2(t). \tag{26}$$

Remark 2: When the weights of the critic network or the supervised action network are close to a convergent region, it is necessary to reduce the learning rate values Ni et al. (2013); Mu et al. (2017). Without loss of generality, let $\frac{1}{N_s}\sum_{k=t-N_s+1}^{t}\|w_c(t) - w_c(t-1)\| < \varepsilon_s$ represent that the weights are close to the convergent region. In this case, due to reducing the learning rates, when adding a weak compensation control signal $u_s(t)$, the system state change resulting from $u_s(t)$ has few impact on the learning process. Then, (25) still holds. Thus, the linear system (26) with a persistent disturbance $d_2(t)$ is always existent.

Since the specific information of $d_2(t)$ is unavailable, the disturbance observer Kim et al. (2016) is designed by

$$\begin{cases} \hat{d}_2(t) = k_d \bar{e}(t) - z_d(t), \\ z_d(t+1) = z_d(t) + k_d((k_v - I_m)\bar{e}(t) + u_s(t) + \hat{d}_2(t)), \end{cases} \tag{27}$$

in which $\hat{d}_2(t)$ is an estimation of $d_2(t)$, $k_d \in R^{m\times m}$ is a diagonal observer matrix, and $z_d(t)$ is a new state variable. From the conclusion of Kim et al. (2016), it is known that $\hat{d}_2(t)$ is convergent to $d_2(t)$.

Inspired by Du et al. (2016), we design a chattering-free compensation control as

$$u_s(t) = \begin{cases} (q_{s1} - k_v)\bar{e}(t) - q_{s2}\operatorname{sig}^{\alpha_s}(\bar{e}(t)) - \hat{d}_2(t), \text{ if } \alpha_s = 1 \\ \quad \text{and } \dfrac{1}{N_s} \displaystyle\sum_{k=t-N_s+1}^{t} \|w_c(t) - w_c(t-1)\| < \varepsilon_s, \\ 0, \text{ otherwise}, \end{cases} \tag{28}$$

where $0 < q_{s1} < 1$, $0 < q_{s2} < 1$, $0 < \alpha_s < 1$, $\operatorname{sig}^{\alpha_s}(\cdot) = \operatorname{sgn}(\cdot)|\cdot|^{\alpha_s}$, $\varepsilon_s > 0$ is a design constant, and N_s is a positive integer.

According to the results of Du et al. (2016), we know that the compensation control $u_s(t)$ is a chattering-free signal and has a capability of the disturbance attenuation. Then, $u_s(t)$ ensures the system's robust to the uncertainty by compensating $d_2(t)$. As such, our developed online adaptive critic robust control method not only has the high-efficiency optimal control property in real time, but also keeps robust to the uncertainty. The procedure to realize this method is summarized as Algorithm 1.

4. SIMULATION

To verify the superiority of the theoretical results, a simulation example on our developed method is conducted by comparing with the traditional online ADP methods. Dynamics of a one-link robot manipulator is considered in the following:

$$G\ddot{\theta} + D\dot{\theta} + MgL\sin(\theta) = \tau, \tag{29}$$

where $g = 9.8\,\mathrm{m/s^2}$ is a gravitational acceleration, $D = 1$ represents a viscous friction coefficient, $L = 1\,\mathrm{m}$ stands for the length of the link, $M = 1\,\mathrm{kg}$ represents the payload mass, $G = 1\,\mathrm{kg\cdot m^2}$ stands for the inertia moment, θ is the angle position, τ is the torque, and τ_d is a disturbance. Note that, its dynamics is unavailable for the controller design.

Discretizing (29) using Euler methods with the sampling interval $T_s = 0.05\,\mathrm{s}$ yields

$$\begin{cases} x_1(t+1) = x_2(t), \\ x_2(t+1) = \frac{2G - DT_s}{G} x_2(t) - \frac{G - DT_s}{G} x_1(t) \\ \qquad - \frac{MgLT_s^2}{G}\sin(x_1(t)) + \frac{T_s^2}{G} u(t) + \frac{T_s^2}{G} d(t), \end{cases} \tag{30}$$

where $d(t) = 0.08\cos(1.8T_s t - \frac{\pi}{4})\sin(T_s t + \frac{\pi}{3})$.

The reference model is given by

$$\begin{cases} x_{r1}(t+1) = x_{r2}(t), \\ x_{r2}(t+1) = (2 - 1.5T_s)x_{r2}(t) + (1.5T_s - 2.5T_s^2 - 1) \\ \qquad \times x_{r1}(t) + T_s^2 u_r(t), \end{cases} \tag{31}$$

where $u_r(t) = \sin(0.2T_s t)\cos(0.4T_s t + \frac{\pi}{2})$.

Algorithm 1:

* i_{tri}: the maximal trial numbers;
t_c: the cumulative time for breaking;
t_t: the simulation terminal time;
i_c, i_a: the maximal iteration numbers in the critic network and the supervised action network, respectively;
E_{ct}, E_{at}: the objective function thresholds in the critic network and the supervised action network, respectively;

1): set the coefficients $\lambda_1, \lambda_2, \ldots, \lambda_{n-1}$, γ, Q, R, k_v, k_d, η_a, η_c, N_a, N_s, ε_a, ε_s, ε_e, q_{s1}, and q_{s2};
2): **for** 1 to i_{tri} **do** * trial
3): initialize $x(0)$, $x^*(0)$;
4): initialize $w_c(0)$, $v_c(0)$, $w_a(0)$, and $v_a(0)$ randomly;
5): $u_s(0) = 0$ and $u_a(0) \leftarrow (14)$;
6): **while** $t \le t_t$ **do**
7): $w_a(t) = w_a(t-1)$, $v_a(t) = v_a(t-1)$, $w_c(t) = w_c(t-1)$, and $v_c(t) = v_c(t-1)$;
8): calculate $u(t-1) \leftarrow (8)$;
9): $x(t) \leftarrow (1)$, $x^*(t) \leftarrow (2)$, $e(t) \leftarrow (3)$, and $\bar{e}(t) \leftarrow (6)$;
10): **if** $t > t_c$ and $\frac{1}{t_c}\sum_{k=t-t_c+1}^{t_c} \|\bar{e}(k)\| > \varepsilon_e$
11): break this trial;
12): **endif**
13): calculate $\hat{J}(t) \leftarrow (11)$, $u_a(t) \leftarrow (14)$, $\hat{d}_2(t) \leftarrow (27)$, $u_s(t) \leftarrow (28)$, and $\alpha_c \leftarrow (23)$;
14): $r(t) = e^T(t)Qe(t) + u^T(t)Ru(t)$;
15): calculate $E_c(t)$ and set $i = 0$;
16): **while** $((i < i_c)$ & $(E_c(t) > E_{ct}))$ **do**
17): update $w_c(t) = w_c(t-1) + \Delta w_c(t)$ and $v_c(t) = v_c(t-1) + \Delta v_c(t)$;
18): $\hat{J}(t) \leftarrow (11)$;
19): **if** $\frac{1}{N_s}\sum_{k=t-N_s+1}^{t} \|w_c(t) - w_c(t-1)\| < \varepsilon_s$
20): reduce η_a and η_c;
21): **else**
22): reset η_a and η_c;
23): **endif**
24): calculate $E_c(t)$ and set $i = i+1$;
25): **endwhile** * critic network
26): calculate $E_a(t)$ and set $j = 0$;
27): **while** $((j < i_a)$ & $(E_a(t) > E_{at}))$ **do**
28): update $w_a(t) = w_a(t-1) + \Delta w_a(t)$ and $v_a(t) = v_a(t-1) + \Delta v_a(t)$;
29): $u_a(t) \leftarrow (14)$ and $u(t) \leftarrow (8)$;
30): $\hat{J}(t) \leftarrow (11)$;
31): calculate $E_a(t)$ and set $j = j+1$;
32): **endwhile** * action network
33): $t = t+1$;
34): **endwhile**
35): **endfor**

The matrices Q and R is chosen as $\operatorname{diag}\{0.5, 0.5\}$ and 0.3, respectively. The critic network and the supervised action network are constructed by two three-layer back propagation NNs with structures of 3-4-1 and 2-3-1, respectively. The activation functions $\phi_c(\cdot)$ and $\phi_a(\cdot)$ are selected as the hyperbolic tangent function. The initial weights for both the networks are randomly generated from $[-1, 1]$. In view of the result of the adaptive critic robust control method, some parameters used in the simulation are presented in Table 1. In addition, by combining with the results of Kim et al. (2016); Du et al. (2016), the observer and compensation control parameters are chosen as $k_d = 0.3$, $q_{s1} = 0.6$, and $q_{s2} = 0.4$.

The trajectories of the system state $x_1(t)$ and the reference state $x_{r1}(t)$ are presented in Fig. 2. The curve of the model reference tracking error $e_1(t)$ is depicted in Fig. 3. It is observed that the system (31) can exactly track the reference model (30) on

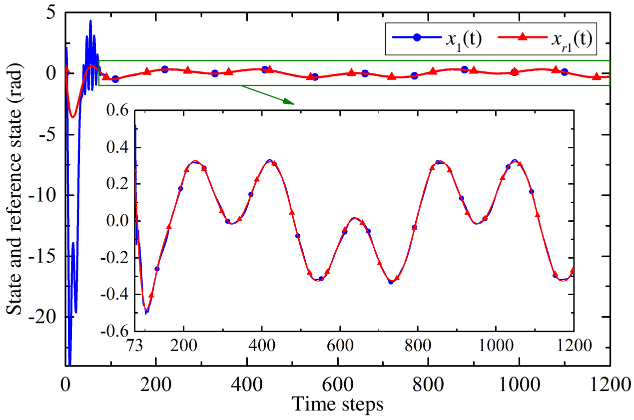

Fig. 2. System state $x_1(t)$ and reference state $x_{r1}(t)$.

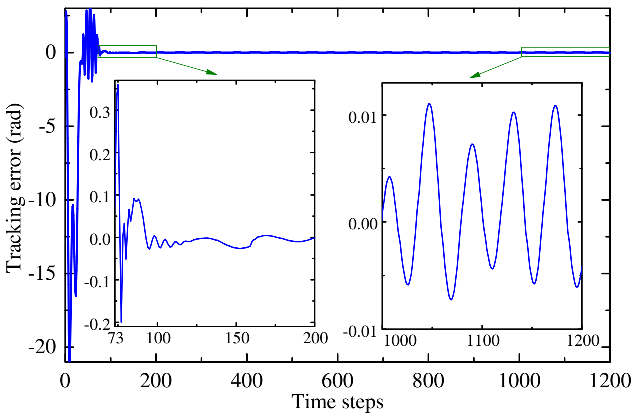

Fig. 3. Model reference tracking error $e_1(t)$.

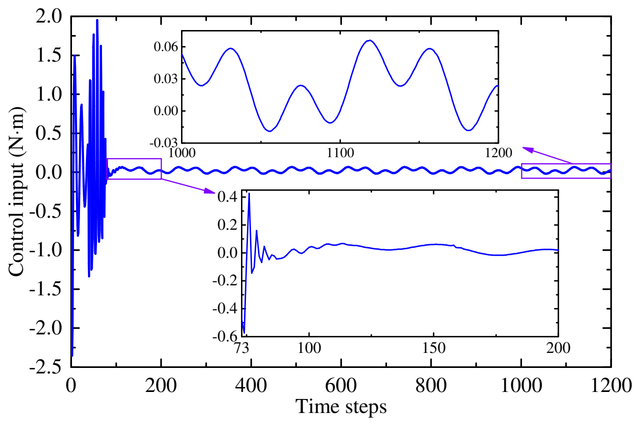

Fig. 4. Control input curve.

behavior by using our developed method. The control input curve is shown in Fig. 4.

To highlight the better learning efficiency and robustness of our developed online adaptive critic robust control method, it

Table 1. Parameters in the example

Para.	Value	Para.	Value	Para.	Value	Para.	Value	Para.	Value
λ_1	0.3	γ	0.95	k_v	0.1	η_a	0.001	η_c	0.004
Para.	Value	Para.	Value	Para.	Value	Para.	Value	Para.	Value
i_a	300	i_c	200	E_{at}	1e-4	E_{ct}	6e-4	N_a	10
Para.	Value	Para.	Value	Para.	Value	Para.	Value	-	-
N_s	10	ε_a	0.7	ε_e	1	ε_s	0.1	-	-

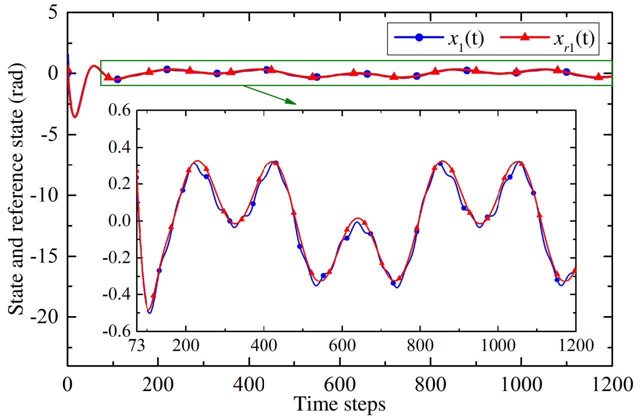

Fig. 5. System state $x_1(t)$ and reference state $x_{r1}(t)$ in Ni et al. (2013).

is necessary to conduct a comparative simulation experiment by comparing with the previous method Ni et al. (2013). As Ni et al. (2013) is a tracking control method, the trajectory of the reference model needs to be obtained beforehand, and then the tracking control is implemented. The reference network, the critic network, and the action network are constructed by three similar NNs with structures of 5-4-1, 6-5-1, and 4-3-1, respectively. Let $u(t) = \bar{u}(t)/\hat{g}(t)$ in (30), in which $\bar{u}(t)$ is the output of the action network. Some parameters of this method selected in the simulation are presented in Table 2, in which η_r, i_r, and E_{rt} represent the learning rate, the maximal iteration numbers, and the objective function threshold for the reference network, respectively. Other parameter setting and some initial conditions are set as same as those of our developed online adaptive critic robust control method, such as λ_1, γ, k_v, the initial states, and the initial weights.

Table 2. Parameters in the example for Ni et al. (2013)

Para.	Value	Para.	Value	Para.	Value	Para.	Value	Para.	Value
η_a	0.001	η_c	0.004	η_r	0.001	i_a	300	i_c	200
Para.	Value	Para.	Value	Para.	Value	Para.	Value	Para.	Value
i_r	150	E_{at}	1e-4	E_{ct}	6e-4	E_{rt}	2e-4	-	-

By using the method proposed in Ni et al. (2013), we can obtain the trajectories of $x_1(t)$ and $x_{r1}(t)$, and the model reference tracking error curve, which are shown in Figs. 5 and 6, respectively. Through the comparisons between Figs. 2 and 3 and that between Figs. 5 and 6, it can be seen that our developed method produces the smaller model reference tracking error than the method proposed in Ni et al. (2013) does. This means that our developed method has the superior robustness.

Table 3. Simulation results on both the methods

Methods	Number of experiments	Number of trials	Success rate (%)
Traditional method	100	20	53
Our method	100	20	100

In this comparative simulation study, a run consists of a maximum of 20 consecutive trials. It is considered successful if $\frac{1}{N_\alpha}\sum_{k=t-N_\alpha+1}^{t}\|\bar{e}(t)\| \leq 0.07$ for $\forall t > 900$ holds. Otherwise, if the controller is unable to learn to make the system (30) track the reference model (31) on behavior within 20 trials, then the run is considered unsuccessful. We run 100 experiments for the traditional method Ni et al. (2013) and our developed method, whose simulation results are listed in Table 3. It is observed

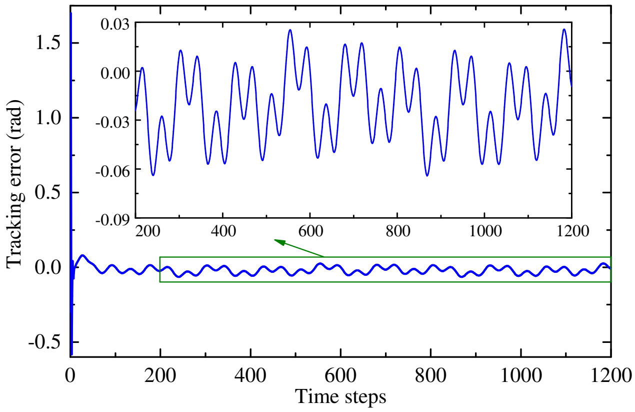

Fig. 6. Model reference tracking error $e_1(t)$ in Ni et al. (2013).

that, in contrast to the traditional online ADP method Ni et al. (2013), our developed method reduces greatly the learning failure rate.

5. CONCLUSION

An online adaptive critic robust control method has been developed to handle the optimal MRAC problem for the nonlinear systems. The online adaptive critic robust controller consists of the critic network, the supervised action network, and the compensation control term. Via the new defined learning schedule factor, such controller not only achieves the high-efficiency learning as well as the optimality in real time, but also has the robustness to the uncertainty. A comparative simulation has been provided to show the superiority of our developed method. Further investigation and experimentation are recommended into the stability analysis, optimization of the algorithm, and applications to the real systems.

REFERENCES

Chen, X., Wang, W., Cao, W., & Wu, M. (2019). Gaussian-kernel-based adaptive critic design using two-phase value iteration. *Information Sciences*, 482, 139–155.

Du, H., Yu, X., Chen, M. Z. Q., & Li, S. (2016). Chattering-free discrete-time sliding mode control. *Automatica*, 68, 87-91.

Fathinezhad, F., Derhami, V., & Rezaeian, M. (2016). Supervised fuzzy reinforcement learning for robot navigation. *Applied Soft Computing*, 40, 33-41.

Fu, H., Chen, X., & Wang, W. (2017). A model reference adaptive control with ADP-to-SMC strategy for unknown nonlinear systems. *Proceedings of the 11th Asian Control Conference*, 1537-1542.

Fu, H. Chen, X. Wang, W. & Wu, M. (2020). MRAC for unknown discrete-time nonlinear systems based on supervised neural dynamic programming. *Neurocomputing*, 384, 130-141.

Ha, M., Wang, D., & Liu, D. (2018). Event-triggered adaptive critic control design for discrete-time constrained nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, doi: 10.1109/TSMC.2018.2868510.

He, H., Ni, Z., & Fu, J. (2012). A three-network architecture for on-line learning and optimization based on adaptive dynamic programming. *Neurocomputing*, 78(1), 3-13.

Jiang, H., Zhang, H., Xiao, G., & Cui, X. (2018). Data-based approximate optimal control for nonzero-sum games of multi-player systems using adaptive dynamic programming. *Neurocomputing*, 275, 192-199.

Kim, K. & Rew, H. (2013). Reduced order disturbance observer for discrete-time linear systems. *Automatica*, 49(4), 968-975.

Lian, C., Xu, X., Chen, H., & He, H. (2016). Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming. *IEEE Transactions on Cybernetics*, 46(11), 2484-2496.

Liu, F., Sun, J., Si, J., Guo, W., & Mei, S. (2012). A boundedness result for the direct heuristic dynamic programming. *Neural Network*, 32, 229-235.

Mu, C., Ni, Z., Sun, C., & He, H. (2017). Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 584–598.

Mu, C., Ni, Z., Sun, C., & He, H. (2017). Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems. *IEEE Transactions on Cybernetics*, 47(6), 1460-1470.

Ni, Z., He, H., & Wen, J. (2013). Adaptive learning in tracking control based on the dual critic network design. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6), 913-928.

Pang, B. & Jiang, Z. P. (2019). Adaptive optimal control of linear periodic systems: An off-policy value iteration approach. arXiv: 1901.08650.

Radac, M. B., Precup, R. E., & Roman, R. C. (2017). Model-free control performance improvement using virtual reference feedback tuning and reinforcement Q-learning. *International Journal of Systems Science*, *48*(5), 1071–1083.

Radac, M. B., Precup, R. E., & Roman, R. C. (2018). Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-learning. *ISA Transactions*, *73*, 227–238.

Si, J. & Wang, Y. T. (2001). On-line learning control by association and reinforcement. *IEEE Transactions on Neural Network*, 12(2), 264-276.

Wang, W., Chen, X., Wang, F., & Fu, H. (2018). ADP-based model reference adaptive control design for unknown discrete-time nonlinear systems. *Proceedings of the 37th Chinese Control Conference*, 8049–8054.

Wang, W. Y., Chan, M. L., Hsu, C. C. J., & Lee, T. T. (2002). H_∞ tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(4), 483-492.

Wang, Z., Wei, Q., & Liu, D. (2020). Event-triggered adaptive dynamic programming for discrete-time multi-player games. *Information Sciences*, 506, 457-470.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In D. A. White, & D. A. Sofge (Eds.), *Handbook of intelligent control*. New York: Van Nostrand Reinhold, (Chapter 13).

Yang, L., Si, J., Tsakalis, K. S., & Rodriguez, A. A. (2009). Direct heuristic dynamic programming for nonlinear tracking control with filtered tracking error. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6), 1617-1622.

Zhao, D., Wang, B., & Liu, D. (2013). A supervised actor-critic approach for adaptive cruise control. *Soft Computing*, 17(11), 2089-2099.

Zhao, D., Zhang, Q., Wang, D., & Zhu, Y. (2016). Experience replay for optimal control of nonzero-sum game systems with unknown dynamics. *IEEE Transactions on Cybernetics*, 46(3), 854-865.