A new approach of optimal control for a class of continuous-time chaotic systems by an online ADP algorithm*

Song Rui-Zhuo(宋睿卓)[a], Xiao Wen-Dong(肖文栋)[a], and Wei Qing-Lai(魏庆来)[b]†

[a] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

(Received 18 October 2013; revised manuscript received 16 November 2013; published online 1 April 2014)

We develop an online adaptive dynamic programming (ADP) based optimal control scheme for continuous-time chaotic systems. The idea is to use the ADP algorithm to obtain the optimal control input that makes the performance index function reach an optimum. The expression of the performance index function for the chaotic system is first presented. The online ADP algorithm is presented to achieve optimal control. In the ADP structure, neural networks are used to construct a critic network and an action network, which can obtain an approximate performance index function and the control input, respectively. It is proven that the critic parameter error dynamics and the closed-loop chaotic systems are uniformly ultimately bounded exponentially. Our simulation results illustrate the performance of the established optimal control method.

Keywords: adaptive dynamic programming, adaptive critic designs, optimal control, continuous-time chaotic system

PACS: 05.45.Gg **DOI:** 10.1088/1674-1056/23/5/050504

1. Introduction

Chaotic systems have been widely applied in the areas of biological engineering, communication, chemical processing, and secure information processing.[1–4] Researchers have studied the control problem of chaotic systems,[5,6] such as impulsive control method[7–10] and adaptive synchronization control method.[11] It is noted that optimal control is a very important aspect in the control field. In Ref. [12], an optimal tracking control scheme was established using the iterative approximation-error-based adaptive dynamic programming (ADP) algorithm for a class of discrete-time chaotic systems. In this paper, we will continue to study the continuous-time chaotic systems via an online adaptive ADP algorithm.

The ADP was proposed by Werbos in Ref. [13] in order to obtain approximate solutions of optimization problems. A detailed discussion and progress regarding the ADP can be found in Refs. [14]– [16]. For the optimal control problems of discrete-time nonlinear systems, most work has adopted an iterative ADP algorithm.[17–20] It is challenging to prove the stability and convergence of ADP-based optimal controllers for continuous-time systems. In Ref. [21], the Hamilton–Jacobi–Bellman (HJB) framework was used to construct a stable iterative ADP algorithm. In Ref. [22], a general multiple-input-multiple output ADP structure was developed and its application in the looper system of a hot strip mill was discussed. In Ref. [23], only measured system input/output data were used to implement the ADP algorithm. In Ref. [24], a practical design method was developed for the cooperative tracking control of higher-order nonlinear systems with a dynamic leader. None of the aforementioned approaches consider the optimal control problem of continuous-time chaotic systems, which is very important for nonlinear control systems.

In this paper, we propose an online ADP algorithm to deal with the optimal control problem for continuous-time chaotic systems for the first time. First, the performance index function is presented for the chaotic system. Then, the critic and action networks are constructed by neural networks, to obtain the approximate performance index function and control input. The critic parameter error dynamics exponential convergence is proven. Furthermore, it is also proven that the closed-loop chaotic system is uniformly ultimately bounded. Next, the detailed implement process of the online ADP algorithm for chaotic systems is introduced. Finally, two chaotic systems are given, and the effectiveness of the proposed optimal control algorithm for chaotic systems is validated by the simulation results.

The rest of this paper is organized as follows. In Section 2, we present the problem statement. In Section 3, the optimal control scheme that has been developed and the convergence proof is given. In Section 4, two examples are given to demonstrate the effectiveness of the proposed optimal control scheme. In Section 5, the conclusion is drawn.

*Project supported by the Open Research Project from the SKLMCCS (Grant No. 20120106), the Fundamental Research Funds for the Central Universities of China (Grant No. FRF-TP-13-018A), the Postdoctoral Science Foundation of China (Grant No. 2013M530527), the National Natural Science Foundation of China (Grant Nos. 61304079 and 61374105), and the Natural Science Foundation of Beijing, China (Grant No. 4132078 and 4143065).

†Corresponding author. E-mail: ql_wei@126.com

© 2014 Chinese Physical Society and IOP Publishing Ltd

2. Problem statement

Let us consider the following nonlinear continuous-time chaotic systems:

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

where the system state $\boldsymbol{x}(t) \in \boldsymbol{R}^n$, the constant control gains $\boldsymbol{g} \in \boldsymbol{R}^{n\times m}$, and the control input $\boldsymbol{u}(\boldsymbol{x}(t)) \in \boldsymbol{R}^m$. We assume that function $\boldsymbol{f}(\boldsymbol{x}(t))$ is Lipschitz continuous on a set $R_1 \in \boldsymbol{R}^n$ and $f(0) = 0$.

We aim to find the control input $\boldsymbol{u}(\boldsymbol{x})$ of chaotic system (1), which optimizes the following performance index function:

$$\boldsymbol{J}(\boldsymbol{x}(t)) = \int_t^\infty \boldsymbol{U}(\boldsymbol{x}(\tau), u(x(\tau)))\,\mathrm{d}\tau, \tag{2}$$

where $\boldsymbol{U}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{x}(\tau))) = x^{\mathrm{T}}(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{u}^{\mathrm{T}}(x(\tau))\boldsymbol{R}\boldsymbol{u}(\boldsymbol{x}(\tau))$ is the utility function. Note that the performance index function (2) can be expanded as follows:

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{x}(t)) &= \int_t^{\mathrm{T}} \boldsymbol{U}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{x}(\tau)))\,\mathrm{d}\tau \\ &\quad + \int_T^\infty \boldsymbol{U}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{x}(\tau)))\,\mathrm{d}\tau \\ &= \int_t^{\mathrm{T}} \boldsymbol{U}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{x}(\tau)))\,\mathrm{d}\tau + \boldsymbol{J}(\boldsymbol{x}(T)),\end{aligned} \tag{3}$$

where $T > t$.

Remark 1 In fact, many chaotic systems can be described by Eq. (1), such as the Chen system, Lorenz system, Rössler system, Lü system, van der Pol oscillator, unified system, several variants of Chua's circuits, and Duffing oscillator.[6]

The infinitesimal version of Eq. (3) is the nonlinear Lyapunov equation

$$0 = \boldsymbol{U}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}\boldsymbol{u}(\boldsymbol{x})), \tag{4}$$

where $\boldsymbol{J}_x = \partial \boldsymbol{J}/\partial x$ is the partial derivative of the performance index function J.

Hence, we can define the Hamiltonian function as

$$\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{J}_x) = \boldsymbol{U}(\boldsymbol{x}, u) + \boldsymbol{J}_x^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}\boldsymbol{u}(\boldsymbol{x})), \tag{5}$$

and the HJB equation

$$0 = \min_{u\in\bar{u}}\{\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{J}_{\boldsymbol{x}}^*)\}, \tag{6}$$

where $\bar{\boldsymbol{u}}$ is a set of admissible policies, and the optimal performance index function $\boldsymbol{J}^*$ is defined as

$$\boldsymbol{J}^*(x) = \min_{u\in\bar{u}} \int_t^\infty \boldsymbol{U}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{x}(\tau)))\,\mathrm{d}\tau. \tag{7}$$

Furthermore, we obtain the optimal feedback control as

$$\boldsymbol{u}^*(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\boldsymbol{J}_x^*. \tag{8}$$

Then, we have the following HJB equation:

$$\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}^*, \boldsymbol{J}_x^*) = \boldsymbol{J}_x^*(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{u}) + \boldsymbol{U}(\boldsymbol{x}, \boldsymbol{u}^*) = 0. \tag{9}$$

3. Optimal control based on the online ADP algorithm

In this section, the optimal control design method using the ADP will be established. As we know, the critic network and the action network are two vital modules of the ADP method that are used to obtain the performance index function $\boldsymbol{J}(\boldsymbol{x})$ and the control input $\boldsymbol{u}(\boldsymbol{x})$, respectively. Two neural networks are used as the critic network and the action network. The whole structure is shown in Fig. 1.

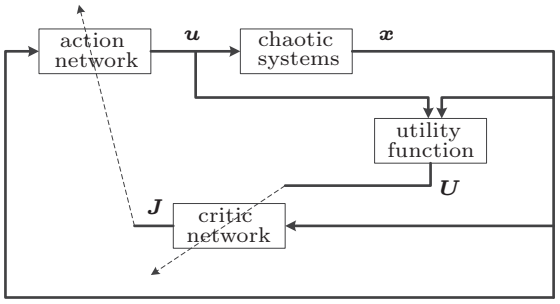


Fig. 1. The whole structure of the proposed online ADP algorithm.

In the neural network, if the number of hidden layer neurons is L, the weight matrix between the input layer and hidden layer is $\boldsymbol{Y}$, the weight matrix between the hidden layer and output layer is $\boldsymbol{W}$, and the input vector of the neural network is $\boldsymbol{X}$, then the output of a three-layer neural network is expressed as

$$\boldsymbol{F}_N(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W}) = \boldsymbol{W}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}(\boldsymbol{Y}\boldsymbol{X}), \tag{10}$$

where $\hat{\boldsymbol{\sigma}}(\boldsymbol{Y}\boldsymbol{X})$ is the activation function. For convenience of analysis, only the output weight $\boldsymbol{W}$ is updated during the training, while the hidden weight is kept unchanged. Hence, in the following part, the neural network function (10) can be simplified into

$$\boldsymbol{F}_N(\boldsymbol{X}, \boldsymbol{W}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\sigma}(\boldsymbol{X}). \tag{11}$$

3.1. Design method of the critic network and the action network

The critic network is used to obtain the performance index function $\boldsymbol{J}(\boldsymbol{x})$ in the form of

$$\boldsymbol{J}(x) = \boldsymbol{W}_{\mathrm{c}}^{\mathrm{T}}\phi_{\mathrm{c}}(x) + \varepsilon_{\mathrm{c}}(x), \tag{12}$$

where $\boldsymbol{W}_{\mathrm{c}}$ is the ideal constant weight, $\phi_{\mathrm{c}}(x)$ is the activation function of critic network, and $\varepsilon_{\mathrm{c}}(x)$ is the approximation error.

Hence, we can have the derivative of the performance index function $\boldsymbol{J}(x)$ with respect to x

$$\boldsymbol{J}_x = \boldsymbol{\nabla}\phi_{\mathrm{c}}^{\mathrm{T}}\boldsymbol{W}_{\mathrm{c}} + \boldsymbol{\nabla}\varepsilon_{\mathrm{c}}, \tag{13}$$

where $\nabla\phi_c = \partial\phi_c(x)/\partial x$ and $\nabla\varepsilon_c = \partial\varepsilon_c(x)/\partial x$. From Eq. (13), the Hamiltonian function (5) can be expressed as

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{W}_c) &= \boldsymbol{U}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{W}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(\boldsymbol{x}) \\ &\quad + \boldsymbol{g}\boldsymbol{u}(\boldsymbol{x})) - \varepsilon_H(x), \end{aligned} \tag{14}$$

where $\varepsilon_H(x) = -\nabla\varepsilon_c^{\mathrm{T}}(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x))$.

If we define the estimation of $\boldsymbol{W}_c$ as $\hat{\boldsymbol{W}}_c$, then we can have the estimation $\hat{\boldsymbol{J}}(x)$ of $\boldsymbol{J}(x)$ as

$$\hat{\boldsymbol{J}}(x) = \hat{\boldsymbol{W}}_c^{\mathrm{T}}\phi_c(x), \tag{15}$$

and the corresponding estimation of $\boldsymbol{J}_x$ is given by

$$\hat{\boldsymbol{J}}_x = \nabla\phi_c^{\mathrm{T}}\hat{\boldsymbol{W}}_c. \tag{16}$$

Then we have the approximate Hamiltonian function derived by

$$\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},\hat{\boldsymbol{W}}_c) = \boldsymbol{U}(\boldsymbol{x},\boldsymbol{u}) + \hat{\boldsymbol{W}}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x)). \tag{17}$$

If we let the weight estimation error of the critic network be

$$\tilde{\boldsymbol{W}}_c = \boldsymbol{W}_c - \hat{\boldsymbol{W}}_c, \tag{18}$$

the estimation error of the Hamiltonian function be

$$\boldsymbol{e}_c = \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{W}_c) - \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},\hat{\boldsymbol{W}}_c). \tag{19}$$

Furthermore, as $\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{W}_c) = 0$, from (14) we have

$$\varepsilon_H(x) = \boldsymbol{U}(x,u) + \boldsymbol{W}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x)). \tag{20}$$

Then we have

$$\begin{aligned} e_c &= \boldsymbol{W}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x)) - \varepsilon_H(x) \\ &\quad - \hat{\boldsymbol{W}}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x)) \\ &= -\boldsymbol{U}(\boldsymbol{x},\boldsymbol{u}) - \hat{\boldsymbol{W}}_c^{\mathrm{T}}\nabla\phi_c(\boldsymbol{f}(x) + \boldsymbol{g}\boldsymbol{u}(x)). \end{aligned} \tag{21}$$

The weight update law for the critic network is a gradient descent algorithm that aims to minimize the squared residual error $\boldsymbol{E}_c = \frac{1}{2}\boldsymbol{e}_c^{\mathrm{T}}\boldsymbol{e}_c$. Hence, we obtain the weight update law of $\hat{\boldsymbol{W}}_c$ as

$$\dot{\hat{\boldsymbol{W}}}_c = -\beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}\left(\varphi_1^{\mathrm{T}}\hat{\boldsymbol{W}}_c + \boldsymbol{U}\right), \tag{22}$$

where $\varphi_1 = \nabla\phi_c(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{u})$ and the adaptive parameter $\beta_c > 0$. Thus, from Eq. (22), we have

$$\begin{aligned} \dot{\hat{\boldsymbol{W}}}_c &= \beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}\left(\varphi_1^{\mathrm{T}}(\boldsymbol{W}_c - \tilde{\boldsymbol{W}}_c) + \boldsymbol{U}\right) \\ &= -\beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}\varphi_1^{\mathrm{T}}\tilde{\boldsymbol{W}}_c \\ &\quad + \beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}(\varphi_1^{\mathrm{T}}\boldsymbol{W}_c + U). \end{aligned} \tag{23}$$

As a result, equation (23) can be rewritten as

$$\dot{\tilde{\boldsymbol{W}}}_c = -\beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}\varphi_1^{\mathrm{T}}\tilde{W}_c + \beta_c\frac{\varphi_1}{(\varphi_1^{\mathrm{T}}\varphi_1 + 1)^2}\varepsilon_H. \tag{24}$$

Define $\varphi_2 = \varphi_1^{\mathrm{T}}\varphi_1 + 1$ and $\varphi_3 = \dfrac{\varphi_1}{\varphi_1^{\mathrm{T}}\varphi_1 + 1}$, then equation (24) can be changed into

$$\dot{\tilde{\boldsymbol{W}}}_c = -\beta_c\varphi_3\varphi_3^{\mathrm{T}}\tilde{\boldsymbol{W}}_c + \beta_c\frac{\varphi_3}{\varphi_2}\varepsilon_H. \tag{25}$$

In the following, it is desired to give the detailed method for action network weights updating. First, the expression of the action network is introduced as

$$\boldsymbol{u}(x) = \boldsymbol{W}_a^{\mathrm{T}}\phi_a(x) + \varepsilon_a, \tag{26}$$

where $\boldsymbol{W}_a$ is the ideal constant weights, $\phi_a(x)$ is the activation function of action network, and $\varepsilon_a(x)$ is the approximation error.

Let the estimation of $\boldsymbol{W}_a$ be $\hat{\boldsymbol{W}}_a$, and the estimation of $\boldsymbol{u}(\boldsymbol{x})$ be

$$\hat{\boldsymbol{u}}(x) = \hat{\boldsymbol{W}}_a^{\mathrm{T}}\phi_a(x), \tag{27}$$

then the error between the actual output of the action network and the feedback control input exists, which is expressed as

$$\boldsymbol{e}_a = \hat{\boldsymbol{W}}_a^{\mathrm{T}}\phi_a + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\hat{\boldsymbol{W}}_c. \tag{28}$$

The objective function to be minimized by the action network is defined as

$$\boldsymbol{E}_a = \frac{1}{2}\boldsymbol{e}_a^{\mathrm{T}}\boldsymbol{e}_a. \tag{29}$$

The weight update law for the action network weight is a gradient descent algorithm, which is given by

$$\dot{\hat{\boldsymbol{W}}}_a = -\beta_a\phi_a\left(\hat{\boldsymbol{W}}_a^{\mathrm{T}}\phi_a + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\hat{\boldsymbol{W}}_c\right)^{\mathrm{T}}, \tag{30}$$

where $\beta_a > 0$ is the adaptive parameter. Define

$$\tilde{\boldsymbol{W}}_a = \boldsymbol{W}_a - \hat{\boldsymbol{W}}_a, \tag{31}$$

then from Eq. (30), we have

$$\begin{aligned} \dot{\tilde{\boldsymbol{W}}}_a &= \beta_a\phi_a\Big(\left(W_a - \tilde{\boldsymbol{W}}_a\right)^{\mathrm{T}}\phi_a \\ &\quad + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\left(\boldsymbol{W}_c - \tilde{\boldsymbol{W}}_c\right)\Big)^{\mathrm{T}} \\ &= \beta_a\phi_a\Big(W_a^{\mathrm{T}}\phi_a + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\boldsymbol{W}_c \\ &\quad - \tilde{\boldsymbol{W}}_a^{\mathrm{T}}\phi_a - \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\tilde{\boldsymbol{W}}_c\Big)^{\mathrm{T}}. \end{aligned} \tag{32}$$

According to Eqs. (13) and (26), equation (8) can be written as

$$\boldsymbol{W}_a^{\mathrm{T}}\phi_a(x) + \varepsilon_a = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\phi_c^{\mathrm{T}}\boldsymbol{W}_c - \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\nabla\varepsilon_c, \tag{33}$$

which means

$$\boldsymbol{W}_{\rm a}^{\rm T}\phi_{\rm a}(x)+\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\phi_{\rm c}^{\rm T}\boldsymbol{W}_{\rm c}=-\varepsilon_{\rm a}-\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\varepsilon_{\rm c}. \quad (34)$$

Hence, equation (32) can be expressed as

$$\dot{\tilde{\boldsymbol{W}}}_{\rm a}=-\beta_{\rm a}\phi_{\rm a}\left(\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}+\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\phi_{\rm c}^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}-\varepsilon_{\rm w}\right)^{\rm T}, \quad (35)$$

where $\varepsilon_{\rm w}=-\left(\varepsilon_{\rm a}+\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\varepsilon_{\rm c}\right)^{\rm T}$.

3.2. Stability analysis

In this subsection, theorems are presented to indicate the closed-loop system state and network weights estimation errors are uniformly ultimately bounded. First, the assumption will be presented.

Assumption 1 i) The approximation errors of the critic network and action network, that is, $\varepsilon_{\rm c}$ and $\varepsilon_{\rm a}$, satisfy $||\varepsilon_{\rm c}||\leq\varepsilon_{\rm cM}$ and $||\varepsilon_{\rm a}||\leq\varepsilon_{\rm aM}$. The gradient of the critic network approximation error is upper bounded, that is, $||\nabla\varepsilon_{\rm c}||\leq\varepsilon_{\rm dM}$. $\varepsilon_{\rm cM}$, $\varepsilon_{\rm aM}$, and $\varepsilon_{\rm dM}$ are non-negative numbers.

ii) The residual error is upper bounded, that is, $||\varepsilon_H||\leq\varepsilon_{\rm HM}$, $\varphi_{\rm 3m}\leq||\varphi_3||\leq\varphi_{\rm 3M}$, and $\varphi_{\rm 2m}\leq||\varphi_2||$. $\varepsilon_{\rm HM}$, $\varphi_{\rm 3m}$, $\varphi_{\rm 2m}$, and $\varphi_{\rm 3M}$ are non-negative numbers.

iii) The vectors of the activation functions of the action network satisfy $\phi_{\rm am}\leq||\phi_{\rm a}||\leq\phi_{\rm aM}$. The gradient of the activation function vector for the critic network is upper bounded, that is, $||\nabla\phi_{\rm c}||\leq\phi_{\rm dM}$. $\phi_{\rm am}$, $\phi_{\rm aM}$, and $\phi_{\rm dM}$ are non-negative numbers.

Theorem 1 If the critic network weight updating method is provided by Eq. (22), then the critic parameter error dynamics (25) converges exponentially.

Proof We define the Lyapunov function candidate as

$$V_1(t)=\frac{1}{2\beta_{\rm c}}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm c}^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}\right\}, \quad (36)$$

and, hence, the derivative of Eq. (36) as

$$\begin{aligned}\dot{V}_1(t)&={\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm c}^{\rm T}(-\varphi_3\varphi_3^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}+\frac{\varphi_3}{\varphi_2}\varepsilon_H)\right\}\\&\leq-\left\|\frac{\varphi_1}{\varphi_2}\tilde{\boldsymbol{W}}_{\rm c}\right\|\left(\left\|\frac{\varphi_1^{\rm T}}{\varphi_2}\tilde{\boldsymbol{W}}_{\rm c}\right\|-\left\|\frac{\varepsilon_H}{\varphi_2}\right\|\right). \quad (37)\end{aligned}$$

As $\varphi_2\geq1$, $\|\varepsilon_H/\varphi_2\|<\varepsilon_{\rm HM}$. Thus, if $\left\|(\varphi_1^{\rm T}/\varphi_2)\tilde{\boldsymbol{W}}_{\rm c}\right\|>\varepsilon_{\rm HM}$, then $\dot{V}_1(t)\leq0$. This means that $V_1(t)$ decreases and $||\varphi_3^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}||$ is bounded. Therefore, in the light of Ref. [25] and Technical Lemma 2 in Ref. [26], the estimation error $\tilde{\boldsymbol{W}}_{\rm c}$ converges exponentially.

The proof is completed.

We now present the main theorem, which provides the updating methods for the actor and critic networks that guarantee convergence of the ADP algorithm to the optimal controller, and meanwhile guarantees the stability of closed-loop chaotic systems.

Theorem 2 Let the chaotic system dynamics be given by Eq. (1), the critic network be given by Eq. (14), and the action network be given by Eq. (27), respectively. If the weight updating methods are given by Eqs. (21) and (30), and there exist the parameters $\beta_{\rm c}$, $\beta_{\rm a}$, and l such that

$$\beta_{\rm c}<\frac{4\beta_{\rm a}\varphi_{\rm 3m}^2}{2\beta_{\rm a}\left(\frac{\varphi_{\rm 3M}}{\varphi_{\rm 3m}}\right)^2+||\boldsymbol{R}^{-1}||^2||\boldsymbol{g}||^2\varphi_{\rm dM}^2}, \quad (38)$$

$$\beta_{\rm a}<\frac{4\phi_{\rm am}^2}{3\phi_{\rm aM}^2+4||\boldsymbol{g}||^2\phi_{\rm dM}^2}, \quad (39)$$

and

$$l>\max\left\{\frac{||\boldsymbol{g}||^2}{\lambda_{\min}(R)},\frac{2k+3}{\lambda_{\min}(Q)}\right\}, \quad (40)$$

then the closed-loop chaotic system (1), the weight estimation errors $\tilde{\boldsymbol{W}}_{\rm c}$, and $\tilde{\boldsymbol{W}}_{\rm a}$ are uniformly ultimately bounded.

Proof Define the Lyapunov function as

$$V(t)=V_1(t)+V_2(t)+V_3(t), \quad (41)$$

where $V_1(t)$ is given by Eq. (36),

$$V_2(t)=\frac{\beta_{\rm c}}{2\beta_{\rm a}}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\tilde{\boldsymbol{W}}_{\rm a}\right\}, \quad (42)$$

and

$$V_3(t)=\beta_{\rm c}\beta_{\rm a}\left(\boldsymbol{x}^{\rm T}\boldsymbol{x}+l\boldsymbol{J}(\boldsymbol{x})\right),l\geq0. \quad (43)$$

Take the partial derivative of $V(t)$, then we have

$$\dot{V}(t)=\dot{V}_1(t)+\dot{V}_2(t)+\dot{V}_3(t). \quad (44)$$

In the following, we will deduce the expressions of $\dot{V}_1(t)$, $\dot{V}_2(t)$, and $\dot{V}_3(t)$.

First, according to Eq. (25), we can obtain

$$\begin{aligned}\dot{V}_1(t)&={\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm c}^{\rm T}(-\varphi_3\varphi_3^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}+\frac{\varphi_3}{\varphi_2}\varepsilon_H)\right\}\\&\leq-\varphi_{\rm 3m}^2\|\tilde{\boldsymbol{W}}_{\rm c}\|^2+\frac{\beta_{\rm c}}{2}\left(\frac{\varphi_{\rm 3M}}{\varphi_{\rm 2m}}\right)^2\|\tilde{\boldsymbol{W}}_{\rm c}\|^2\\&\quad+\frac{1}{2\beta_{\rm c}}\varepsilon_{\rm HM}^2. \quad (45)\end{aligned}$$

Next, according to Assumption 1, $\varepsilon_{\rm w}$ in Eq. (35) satisfies

$$\varepsilon_{\rm w}\leq\varepsilon_{\rm wM}, \quad (46)$$

where $\varepsilon_{\rm wM}=\varepsilon_{\rm aM}+\frac{1}{2}||\boldsymbol{R}^{-1}||||\boldsymbol{g}^{\rm T}||\varepsilon_{\rm dM}$. Furthermore, from (35) we obtain

$$\begin{aligned}&\dot{V}_2(t)\\&=-\beta_{\rm c}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}\left(\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}+\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\phi_{\rm c}^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}-\varepsilon_{\rm w}\right)^{\rm T}\right\}\\&=-\beta_{\rm c}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}\left(\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}+\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\phi_{\rm c}^{\rm T}\tilde{\boldsymbol{W}}_{\rm c}\right)^{\rm T}\right\}\end{aligned}$$

$$
\begin{aligned}
&+\beta_{\rm c}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}\varepsilon_{\rm w}^{\rm T}\right\}\\
\leq\ &-\beta_{\rm c}\phi_{\rm am}^2\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2-\frac{\beta_{\rm c}}{2}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}(\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\nabla\phi_{\rm c}^{\rm T}\tilde{\boldsymbol{W}}_{\rm c})^{\rm T}\right\}\\
&+\beta_{\rm c}\,{\rm tr}\left\{\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}\varepsilon_{\rm w}^{\rm T}\right\}\\
\leq\ &-\beta_{\rm c}\phi_{\rm am}^2\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2+\frac{\beta_{\rm c}}{2\beta_{\rm a}}\varepsilon_{\rm wM}^2+\frac{\beta_{\rm c}}{4\beta_{\rm a}}\left\|\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\right\|^2\phi_{\rm dM}^2\left\|\tilde{\boldsymbol{W}}_{\rm c}\right\|^2\\
&+\frac{3}{4}\beta_{\rm a}\beta_{\rm c}\phi_{\rm aM}^2\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2. \qquad (47)
\end{aligned}
$$

At last, as

$$\hat{\boldsymbol{u}}=\hat{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}=(\boldsymbol{W}_{\rm a}-\tilde{\boldsymbol{W}}_{\rm a})^{\rm T}\phi_{\rm a}=\boldsymbol{u}-\varepsilon_{\rm a}-\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}, \qquad (48)$$

then we have

$$\dot{\boldsymbol{x}}=\boldsymbol{f}+\boldsymbol{g}\hat{\boldsymbol{u}}=\boldsymbol{f}+\boldsymbol{g}(\boldsymbol{u}-\varepsilon_{\rm a}-\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}). \qquad (49)$$

and thus

$$
\begin{aligned}
\dot{V}_3(t)\ &=\ \beta_{\rm c}\beta_{\rm a}\left(2\boldsymbol{x}^{\rm T}\dot{\boldsymbol{x}}+l\dot{J}\right)\\
&=\ 2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}(\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}-\boldsymbol{g}\varepsilon_{\rm a}-\boldsymbol{g}\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a})-l\beta_{\rm c}\beta_{\rm a}U\\
&=\ 2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{f}+2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{g}\boldsymbol{u}-2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{g}\varepsilon_{\rm a}\\
&\quad-2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{g}\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}-l\beta_{\rm c}\beta_{\rm a}U. \qquad (50)
\end{aligned}
$$

Since $\boldsymbol{f}(\boldsymbol{x})$ is Lipschitz, that is, $||\boldsymbol{f}(\boldsymbol{x})||\leq\boldsymbol{L}||\boldsymbol{x}||$, we have

$$2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{f}\leq 2\beta_{\rm c}\beta_{\rm a}\boldsymbol{L}||\boldsymbol{x}||^2. \qquad (51)$$

Furthermore, we have

$$2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{g}\boldsymbol{u}\leq\beta_{\rm c}\beta_{\rm a}(||\boldsymbol{x}||^2+||\boldsymbol{g}\boldsymbol{u}||^2), \qquad (52)$$

$$-2\beta_{\rm c}\beta_{\rm a}\boldsymbol{x}^{\rm T}\boldsymbol{g}\varepsilon_{\rm a}\leq\beta_{\rm c}\beta_{\rm a}(||\boldsymbol{x}||^2+||\boldsymbol{g}||^2\varepsilon_{\rm aM}^2), \qquad (53)$$

$$-2\beta_{\rm c}\beta_{\rm a}x^{\rm T}\boldsymbol{g}\tilde{\boldsymbol{W}}_{\rm a}^{\rm T}\phi_{\rm a}\leq\beta_{\rm c}\beta_{\rm a}(||x||^2+||\boldsymbol{g}||^2\left\|\tilde{\boldsymbol{W}}_{\rm c}\right\|^2\phi_{\rm aM}^2), \qquad (54)$$

and

$$-l\beta_{\rm c}\beta_{\rm a}U\leq-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(Q)||\boldsymbol{x}||^2-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(\boldsymbol{R})||\boldsymbol{u}||^2. \qquad (55)$$

Then we have

$$
\begin{aligned}
\dot{V}_3(t)\ &\leq\ 2\beta_{\rm c}\beta_{\rm a}L||\boldsymbol{x}||^2+\beta_{\rm c}\beta_{\rm a}(||\boldsymbol{x}||^2+||\boldsymbol{g}||^2||\boldsymbol{u}||^2)\\
&\quad+\beta_{\rm c}\beta_{\rm a}(||\boldsymbol{x}||^2+||\boldsymbol{g}||^2\varepsilon_{\rm aM}^2)\\
&\quad+\beta_{\rm c}\beta_{\rm a}(||\boldsymbol{x}||^2+||\boldsymbol{g}||^2\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2\phi_{\rm aM}^2)\\
&\quad-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(Q)||\boldsymbol{x}||^2-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(R)||\boldsymbol{u}||^2\\
&=\ (2\beta_{\rm c}\beta_{\rm a}L+3\beta_{\rm c}\beta_{\rm a}-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(Q))||\boldsymbol{x}||^2\\
&\quad+(\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(R))||\boldsymbol{u}||^2\\
&\quad+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\phi_{\rm aM}^2\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\varepsilon_{\rm aM}^2. \qquad (56)
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
&\dot{V}(t)\\
\leq\ &(2\beta_{\rm c}\beta_{\rm a}L+3\beta_{\rm c}\beta_{\rm a}-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(Q))||\boldsymbol{x}||^2\\
&+(\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(R))||\boldsymbol{u}||^2\\
&+(-\beta_{\rm c}\phi_{\rm am}^2+\frac{3}{4}\beta_{\rm a}\beta_{\rm c}\phi_{\rm aM}^2+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\phi_{\rm aM}^2)\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2\\
&+\left(-\varphi_{\rm 3m}^2+\frac{\beta_{\rm c}}{2}\left(\frac{\varphi_{\rm 3M}}{\varphi_{\rm 2m}}\right)^2+\frac{\beta_{\rm c}}{4\beta_{\rm a}}\left\|\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\right\|^2\phi_{\rm dM}^2\right)\left\|\tilde{\boldsymbol{W}}_{\rm c}\right\|^2\\
&+\frac{1}{2\beta_{\rm c}}\varepsilon_{\rm HM}^2+\frac{\beta_{\rm c}}{2\beta_{\rm a}}\varepsilon_{\rm wM}^2+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\varepsilon_{\rm aM}^2. \qquad (57)
\end{aligned}
$$

Let

$$
\begin{aligned}
\gamma_x&=2\beta_{\rm c}\beta_{\rm a}L+3\beta_{\rm c}\beta_{\rm a}-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(Q),\\
\gamma_u&=\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2-l\beta_{\rm c}\beta_{\rm a}\lambda_{\min}(R),\\
\gamma_{\rm a}&=-\beta_{\rm c}\phi_{\rm am}^2+\frac{3}{4}\beta_{\rm a}\beta_{\rm c}\phi_{\rm aM}^2+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\phi_{\rm aM}^2,\\
\gamma_{\rm c}&=-\varphi_{\rm 3m}^2+\frac{\beta_{\rm c}}{2}\left(\frac{\varphi_{\rm 3M}}{\varphi_{\rm 2m}}\right)^2+\frac{\beta_{\rm c}}{4\beta_{\rm a}}\left\|\boldsymbol{R}^{-1}\boldsymbol{g}^{\rm T}\right\|^2\phi_{\rm dM}^2,
\end{aligned}
$$

and

$$
\begin{aligned}
V_{\rm b}&=\gamma_x||\boldsymbol{x}||^2+\gamma_u||\boldsymbol{u}||^2+\gamma_{\rm a}\left\|\tilde{\boldsymbol{W}}_{\rm a}\right\|^2+\gamma_{\rm c}\left\|\tilde{\boldsymbol{W}}_{\rm c}\right\|^2,\\
\varepsilon_{\rm V}&=\frac{1}{2\beta_{\rm c}}\varepsilon_{\rm HM}^2+\frac{\beta_{\rm c}}{2\beta_{\rm a}}\varepsilon_{\rm wM}^2+\beta_{\rm c}\beta_{\rm a}||\boldsymbol{g}||^2\varepsilon_{\rm aM}^2,
\end{aligned}
$$

then equation (57) can be written as

$$\dot{V}(t)\leq V_{\rm b}+\varepsilon_{\rm V}. \qquad (58)$$

From Eqs. (38)–(40), we have $V_{\rm b}<0$. Therefore, according to the standard Lyapunov extension theorem,[26] the above analysis demonstrates that the state and the weight errors are uniformly ultimately bounded.

The proof is completed.

3.3. Online ADP algorithm implement

On the basis of the above theoretical analysis, we introduce the following detailed online ADP algorithm implement process.

1) Give the values of Q and R, and the initial chaotic system state;

2) Give an arbitrary admissible control input for chaotic system (1);

3) Construct the structure of the critic network and the action network, that is, the adaptive parameters $\beta_{\rm c}$ and $\beta_{\rm a}$, and the activation functions of the critic network and the action network;

4) Give the initial weights of critic network and action network, and update $\hat{\boldsymbol{W}}_{\rm c}$ and $\hat{\boldsymbol{W}}_{\rm a}$ according to Eqs. (22) and (30), respectively;

5) Compute $\hat{\boldsymbol{J}}(\boldsymbol{x})$ and $\hat{\boldsymbol{u}}(\boldsymbol{x})$ according to Eqs. (15) and (27), respectively; and,

6) Return the control input $\hat{\boldsymbol{u}}(x)$.

4. Simulation results

To evaluate the online ADP algorithm for chaotic systems, we consider two nonlinear chaotic systems.

4.1. Example 1

Consider the following Chua circuit:[10]

$$\begin{cases} \dot{x}_1 = \theta_1(x_2 - h(x_1)) + u_1, \\ \dot{x}_2 = x_1 - x_2 + x_3 + u_2, \\ \dot{x}_3 = -\theta_2 x_2 + u_3, \end{cases} \tag{59}$$

where the nonlinear characteristic $h(x_1) = m_1 x_1 + \frac{1}{2}(m_0 - m_1)(|x_1 + \theta_3| - |x_1 - \theta_3|)$, and the parameters $\theta_1 = 9$, $\theta_2 = 14.28$, $\theta_3 = 1$, $m_0 = -\frac{1}{7}$, and $m_1 = \frac{2}{7}$. Figure 2 depicts the chaotic attractor of the system (59).

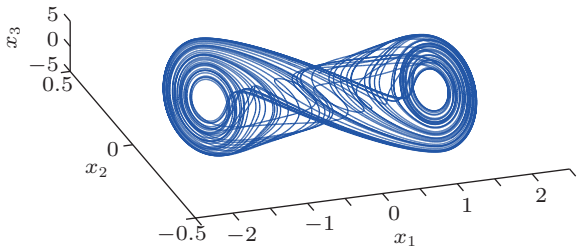


Fig. 2. (color online) The chaotic attractor of Chua's circuit (59).

In this paper, we select adaptive parameters of the critic and action networks as $\beta_c = \beta_a = 0.02$. The activation functions in the critic network and the action network are hyperbolic tangent functions. The structures of action network and critic network are 3–10–1 and 3–3–1, respectively. The initial weights W_a and $\boldsymbol{W}_c$ are selected arbitrarily from $(-0.5, 0.5)$, respectively. For the performance index function, Q and R in the utility function are identity matrices of appropriate dimensions. After 120 time steps, we obtain that $\hat{\boldsymbol{W}}_c$ converges to $\hat{\boldsymbol{W}}_c = [-0.1477; 0.4399; 0.3774]$, and $\hat{\boldsymbol{W}}_a$ converges to $\hat{\boldsymbol{W}}_a = [-0.0982, -0.0090, -0.4034; -0.4240, -0.0107, -0.3680; -0.2601, -0.1623, 0.4421; -0.3767, 0.4001, 0.4561; -0.3161, -0.1308, 0.0752; -0.2600, -0.3888, -0.4402; -0.0827, 0.2803, -0.2652; -0.4503, -0.1103, -0.1468; 0.4027, -0.2584, 0.3211; 0.4448, -0.0961, -0.4846]$. From the initial state $[0.2; 0.13; 0.17]$, we obtain the simulation figures shown in Figs. 3–4. From the two figures, we can see that the state and the control of system (59) are convergent. It is clear that the optimal control method in this paper is very effective and feasible.

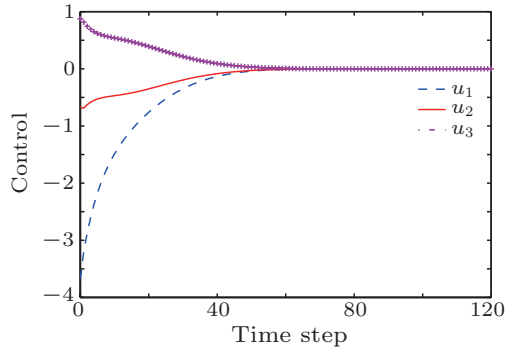


Fig. 3. (color online) The control trajectories of system (59).

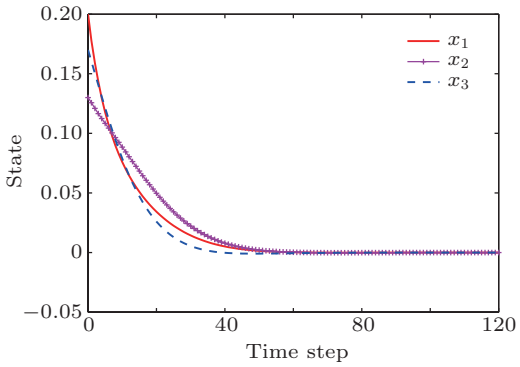


Fig. 4. (color online) The state trajectories of system (59).

4.2. Example 2

The well-known Lorenz system is given by

$$\begin{cases} \dot{x}_1 = \sigma(x_2 - x_1) + 10u_1, \\ \dot{x}_2 = \rho x_1 - x_1 x_3 - x_2 + 10u_2, \\ \dot{x}_3 = x_1 x_2 - \beta x_3 + 10u_3, \end{cases} \tag{60}$$

where the parameters $\beta = 8/3$, $\rho = 28$, and $\sigma = 10$. The chaotic figure is shown in Fig. 5.

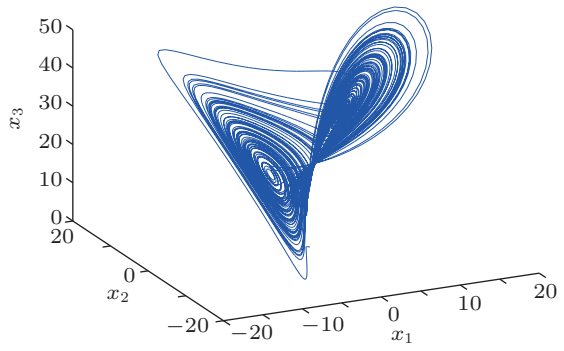


Fig. 5. (color online) The attractor of Lorenz system (60).

The structures of action network and critic network are 3–6–1 and 3–3–1, respectively. The initial weights of the two networks are selected from $(-0.5, 0.5)$. Adaptive parameters of the critic and action networks are $\beta_c = \beta_a = 0.02$. The activation functions in the critic network and the action network are hyperbolic tangent functions, as in Example 1. For the performance index function, Q and R in the utility function are identity matrices of appropriate dimensions. After 200 time steps, we obtain that $\hat{\boldsymbol{W}}_c$ converges to $\hat{\boldsymbol{W}}_c = [0.0968; 0.0717; 0.0570]$, and $\hat{\boldsymbol{W}}_a$ converges to $\hat{\boldsymbol{W}}_a = [0.0629, -0.0442, 0.0913; 0.0811, 0.0093, -0.0029; -0.0746, 0.0915, 0.0599; 0.0826, 0.0929, -0.0715; 0.0264, -0.0684, -0.0156; -0.0804, 0.0941, 0.0831]$. The initial state of the chaotic system (60) is $[0.5; -0.5; 0.9]$. The obtained figures are shown in Figs. 6 and 7. We can see that the closed-loop system state and control inputs converge to zero as the time step increases. Thus, the proposed optimal control method for chaotic systems in this paper is very effective.

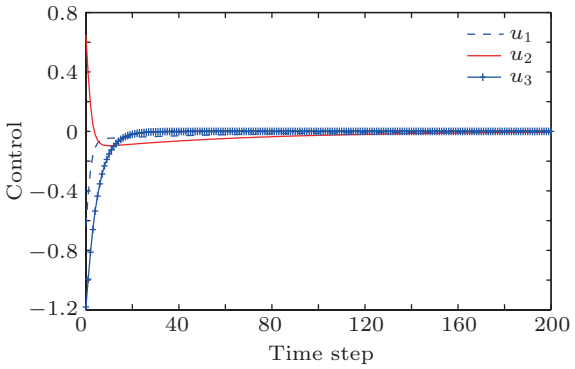


Fig. 6. (color online) The control trajectories of system (60).

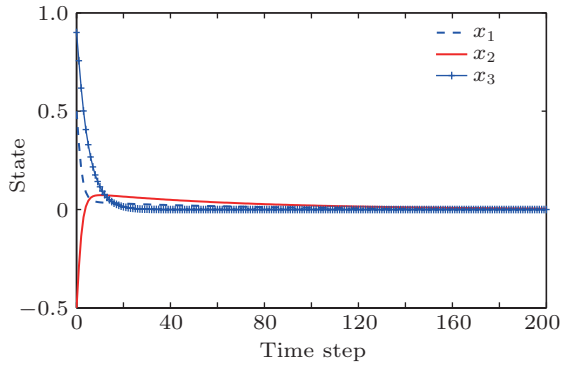


Fig. 7. (color online) The state trajectories of system (60).

5. Conclusions

This paper has developed an optimal control scheme based on an online ADP algorithm for continuous-time chaotic systems. The ADP algorithm is used to obtain the optimal control input that makes the performance index function reach an optimum. First, we presented the performance index function of the chaotic system. Then, the online ADP algorithm is presented, in which the neural networks are used to construct the critic network and action network. The approximate performance index function and control input are obtained by an online ADP algorithm. It is proven that the iterative performance index functions can converge to a finite neighborhood of the lowest bound of all performance index functions. Two simulation examples are given to show the effectiveness of the proposed optimal control approach.

References

[1] Liu S and Chen L 2013 *Chin. Phys. B* **22** 100506
[2] Feng J, Zhang Q, Wang W and Hao S 2013 *Chin. Phys. B* **22** 090503
[3] Luo Y and Du M 2013 *Chin. Phys. B* **22** 080503
[4] Giuseppe G 2013 *Chin. Phys. B* **22** 080505
[5] Chen S and Lu J 2002 *Chaos Soliton. Fract.* **14** 643
[6] Zhang H, Huang W, Wang Z, Wang a and Chai T 2006 *Phys. Lett. A* **350** 363
[7] Ma T and Fu J 2011 *Chin. Phys. B* **20** 050511
[8] Ma T, Fu J and Sun Y 2010 *Chin. Phys. B* **19** 090502
[9] Zhang H, Ma T, Fu J and Tong S 2009 *Chin. Phys. B* **18** 3751
[10] Ma T, Zhang H and Fu J 2008 *Chin. Phys. B* **17** 4407
[11] Zhang H, Wang Z and Liu D 2004 *Int. J. Bifur. Chaos* **14** 3505
[12] Song R, Xiao W, Sun C and Wei Q 2013 *Chin. Phys. B* **22** 090502
[13] Werbos P 1977 *General Systems Yearbook* **22** 25
[14] Zhang H, Song R, Wei Q and Zhang T 2011 *IEEE Trans. Neural Netw.* **22** 1851
[15] Zhang H, Wei Q and Liu D 2011 *Automatica* **47** 207
[16] Liu D and Wei Q 2013 *Automatica*
[17] Zhang H, Wei Q and Luo Y 2008 *IEEE Trans. Syst. Man Cybernet. B* **38** 937
[18] Song R, Zhang H, Luo Y and Wei Q 2010 *Neurocomputing* **73** 3020
[19] Song R, Xiao W and Sun C 2013 *Acta Automatica Sinica* **39** 1413
[20] Song R, Xiao W and Zhang H 2013 *Neurocomputing* **119** 212
[21] Murray J, Cox C, Lendaris G and Saeks R 2002 *IEEE Trans. Syst. Man Cybernet. C* **32** 140
[22] Fu J, He H and Zhou X 2011 *IEEE Trans. Neural Netw.* **22** 1133
[23] Lewis F and Vamvoudakis K 2011 *IEEE Trans. Syst. Man Cybernet. B* **41** 14
[24] Zhang H and Lewis F 2012 *Automatica* **48** 1432
[25] Khalil H 2002 *Nonlinear System* (NJ: Prentice-Hall)
[26] Vamvoudakis K and Lewis F 2010 *Automatica* **46** 878