Approximation-error-ADP-based optimal tracking control for chaotic systems with convergence proof*

Song Rui-Zhuo(宋睿卓)[a)†], Xiao Wen-Dong(肖文栋)[a)], Sun Chang-Yin(孙长银)[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 6 December 2012; revised manuscript received 25 March 2013)

In this paper, an optimal tracking control scheme is proposed for a class of discrete-time chaotic systems using the approximation-error-based adaptive dynamic programming (ADP) algorithm. Via the system transformation, the optimal tracking problem is transformed into an optimal regulation problem, and then the novel optimal tracking control method is proposed. It is shown that for the iterative ADP algorithm with finite approximation error, the iterative performance index functions can converge to a finite neighborhood of the greatest lower bound of all performance index functions under some convergence conditions. Two examples are given to demonstrate the validity of the proposed optimal tracking control scheme for chaotic systems.

Keywords: chaotic systems, approximation error, adaptive dynamic programming, optimal tracking control

PACS: 05.45.Gg **DOI:** 10.1088/1674-1056/22/9/090502

1. Introduction

During recent years, the control problem of chaotic systems has received considerable attention.[2,19] Many different methods are applied theoretically and experimentally to control chaotic systems, such as the adaptive synchronization control method,[22] and the impulsive control method.[8–10,22] However, the methods mentioned above just focus on designing the controller for chaotic systems. Few considered the optimal tracking control problem, which is an important index of chaotic systems.

As is well known, the dynamic programming method is very useful in solving the optimal control problems of nonlinear systems. However, due to the difficulties of solving nonlinear two-point boundary value problems and the time-varying Hamilton–Jacobi–Bellman (HJB) equations using dynamic programming, the optimal control is hardly obtained. Therefore, approximate solutions of the optimal control problems have been proposed.[1,11] Among these approximate approaches, the adaptive dynamic programming (ADP) algorithm, which is proposed by Werbos,[16] has attracted much attention. In the last few years, the ADP algorithm has been promoted by Lewis,[13] Si,[3,12] Liu,[14,23] and so on. In Ref. [17], the ADP approach is classified into four schemes: dual heuristic dynamic programming, heuristic dynamic programming, action dependent dual heuristic dynamic programming, and action-dependent heuristic dynamic programming. Moreover, the research overview and the development directions about ADP were introduced in Refs. [5] and [15].

Although the iterative ADP algorithm has improved greatly in control field, it is still an open problem about how to solve the optimal tracking control problem for chaotic systems based on the ADP algorithm. The reason is that in most implementation methods of the ADP algorithm, accurate optimal control and the performance index function are not obtained, because of the existence of approximate error between the approximate function and the expected one. While the approximate error will influence the control performance of the chaotic systems. Thus, the approximate error need to be considered in the ADP algorithm. This motivates our research. In this paper, we proposed an approximation-error ADP algorithm to deal with the optimal tracking control problem for chaotic systems. First, the optimal tracking problem of a chaotic system is transformed into an optimal regulation problem, and the corresponding performance index function is defined. Then, the approximation-error ADP algorithm is established. It is proved that the ADP algorithm with approximation error makes the iterative performance index functions converge to a finite neighborhood of the optimal performance index function. Finally, two simulation examples are given to show the effectiveness of the proposed optimal tracking control algorithm for chaotic systems.

This paper is organized as follows. In Section 2, we present the problem formulation. In Section 3, the optimal tracking control scheme is developed and the convergence proof is given. In Section 4, two examples are given to demonstrate the effectiveness of the proposed tracking control scheme. In Section 5, the conclusion is given.

*Project supported by the Open Research Project from 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), and the National Natural Science Foundation of China (Grant Nos. 61304079, 61125306, and 61034002).

†Corresponding author. E-mail: ruizhuosong@163.com

© 2013 Chinese Physical Society and IOP Publishing Ltd

2. Problem formulation and preliminaries

Consider the MIMO chaotic dynamic system which can be represented by the following form:

$$\begin{aligned} x_1(k+1) &= f_1(x(k)) + \sum_{j=1}^{m} g_{1j}u_j(k) \\ &\vdots \\ x_n(k+1) &= f_n(x(k)) + \sum_{j=1}^{m} g_{nj}u_j(k), \end{aligned} \tag{1}$$

where $\boldsymbol{x}(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^{\mathrm{T}}$ is the system state vector which is assumed to be available for measurement, $\boldsymbol{u}(k) = [u_1(k), u_2(k), \ldots, u_m(k)]^{\mathrm{T}}$ is the control input, g_{ij} $(i = 1,2,\ldots,n,\ j = 1,2,\ldots,m)$ is the constant control gain. If we denote $\boldsymbol{f}(x(k)) = [f_1(x(k)), f_2(x(k)), \ldots, f_n(x(k))]^{\mathrm{T}}$ and

$$\boldsymbol{g} = \begin{bmatrix} g_{11} \cdots g_{1m} \\ \vdots \\ g_{n1} \cdots g_{nm} \end{bmatrix}. \tag{2}$$

Then, the chaotic system (1) can be rewritten as

$$\boldsymbol{x}(k+1) = \boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{g}\boldsymbol{u}(k). \tag{3}$$

In fact, system (3) has a large family, and lots of chaotic systems can be described by (3), such as Hénon mapping,[4] the new discrete chaotic system proposed in Ref. [7], and many others.

The objective of this paper is to construct an optimal tracking controller, such that the system state x tracks the reference signal x_{d}, and all the signals in the closed-loop system remain bounded. To meet the objective, we make the following assumption.

Assumption 1 The desired trajectory signal x_{d} is continuous, bounded, and available for measurement. Furthermore, the desired trajectory x_{d} has the following form:[18]

$$\boldsymbol{x}_{\mathrm{d}}(k+1) = \boldsymbol{f}(x_{\mathrm{d}}(k)) + \boldsymbol{g}\boldsymbol{u}_{\mathrm{d}}(k). \tag{4}$$

Thus, we define the tracking error as follows:

$$\boldsymbol{e}(k+1) = \boldsymbol{x}(k+1) - \boldsymbol{x}_{\mathrm{d}}(k+1). \tag{5}$$

According to Refs. [21] and [24], we define the control error as

$$\boldsymbol{w}(k) = \boldsymbol{u}(k) - u_{\mathrm{d}}(k), \tag{6}$$

and $\boldsymbol{w}(k) = 0$, for $k < 0$, where $u_{\mathrm{d}}(k)$ denotes the expected control, which can be given as follows:

$$\boldsymbol{u}_{\mathrm{d}}(k) = \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{g}\boldsymbol{g}^{\mathrm{T}} + \varepsilon_0)^{-1}(\boldsymbol{x}_{\mathrm{d}}(k+1) - \boldsymbol{f}(\boldsymbol{x}_{\mathrm{d}}(k))), \tag{7}$$

where ε_0 is a constant matrix, which is used to guarantee the existence of the matrix inverse.

Then, the tracking error is obtained as

$$\begin{aligned} \boldsymbol{e}(k+1) &= \boldsymbol{x}(k+1) - \boldsymbol{x}_{\mathrm{d}}(k+1) \\ &= \boldsymbol{f}(x(k)) - \boldsymbol{f}(\boldsymbol{x}_{\mathrm{d}}(k)) + g\boldsymbol{w}(k) \\ &= \boldsymbol{f}(\boldsymbol{e}(k) + x_{\mathrm{d}}(k)) - \boldsymbol{f}(\boldsymbol{x}_{\mathrm{d}}(k)) + \boldsymbol{g}\boldsymbol{w}(k). \end{aligned} \tag{8}$$

Remark 1 Actually, the term $\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{g}\boldsymbol{g}^{\mathrm{T}} + \varepsilon_0)^{-1}$ in Eq. (7), is the generalized inverse of $\boldsymbol{g}$. As $\boldsymbol{g} \in \Re^{n\times m}$, the generalized inverse technique is used to obtain the inverse of $\boldsymbol{g}$.

To solve the optimal tracking control problem, we present the following performance index function:

$$\boldsymbol{J}(\boldsymbol{e}(k), \boldsymbol{w}(k)) = \sum_{l=k}^{\infty} \boldsymbol{U}(\boldsymbol{e}(l), \boldsymbol{w}(l)), \tag{9}$$

where $\boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) > 0$, $\forall e(k), w(k)$, is the utility function. In this paper, we define

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) &= \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{e}(k) + (\boldsymbol{w}(k) - \boldsymbol{w}(k-1))^{\mathrm{T}} \\ &\quad \times \boldsymbol{R}(\boldsymbol{w}(k) - \boldsymbol{w}(k-1)), \end{aligned} \tag{10}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are both diagonal positive definite matrices. In Eq. (10), the first term means the tracking error, and the second term means the difference of the control error.

According to Bellman's principle of optimality, the optimal performance index function satisfies the HJB equation as follows:

$$\boldsymbol{J}^*(\boldsymbol{e}(k)) = \inf_{\boldsymbol{w}(k)} \{\boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \boldsymbol{J}^*(\boldsymbol{e}(k+1))\}, \tag{11}$$

and the optimal control error $\boldsymbol{w}^*(k)$ is

$$\boldsymbol{w}^*(k) = \arg\inf_{\boldsymbol{w}(k)} \boldsymbol{J}(\boldsymbol{e}(k), \boldsymbol{w}(k)). \tag{12}$$

Thus, the HJB equation (11) can be rewritten as

$$\boldsymbol{J}^*(\boldsymbol{e}(k)) = \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}^*(k)) + \boldsymbol{J}^*(\boldsymbol{e}(k+1)), \tag{13}$$

and the optimal tracking control is

$$\boldsymbol{u}^*(k) = \boldsymbol{w}^*(k) + \boldsymbol{u}_{\mathrm{d}}(k). \tag{14}$$

We can see that if we want to obtain the optimal tracking control $\boldsymbol{u}^*(k)$, we must obtain $\boldsymbol{w}^*(k)$ and the optimal performance index function $\boldsymbol{J}^*(\boldsymbol{e}(k))$. Generally, $\boldsymbol{J}^*(\boldsymbol{e}(k))$ is unknown before all the control error $\boldsymbol{w}^*(k)$ is considered. If we adopt the traditional dynamic programming method to obtain the optimal performance index function, then we have to face the "the curse of dimensionality". This makes the optimal control nearly impossible to be obtained by the HJB equation. Thus, in the next part, we present an iterative ADP algorithm, based on Bellman's principle of optimality.

3. Optimal tracking control scheme based on the approximation-error ADP algorithm

3.1. Description of the approximation-error ADP algorithm

First, for $\forall \boldsymbol{e}(k)$, let the initial function $\Upsilon(\boldsymbol{e}(k))$ be an arbitrary function that satisfies $\Upsilon(\boldsymbol{e}(k)) \in \bar{\Upsilon}_{e(k)}$, where $\bar{\Upsilon}_{e(k)}$ is defined as follows.

Definition 1 Let

$$\bar{\Upsilon}_{e(k)} = \{\Upsilon(e(k)) : \Upsilon(e(k)) > 0, \Upsilon(e(k+1)) < \Upsilon(e(k))\} \quad (15)$$

be the initial positive definition function set, where $\boldsymbol{e}(k+1) = \boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}(k)) - \boldsymbol{x}_{\mathrm{d}}(k+1)$.

For $\forall \boldsymbol{e}(k)$, let the initial performance index function $\boldsymbol{J}^{[0]}(\boldsymbol{e}(k)) = \hat{\theta}\Upsilon(\boldsymbol{e}(k))$, where $\hat{\theta} > 0$ is a large enough finite positive constant. The initial iterative control policy $w^{[0]}(k)$ can be computed as follows:

$$w^{[0]}(k) = \arg\inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \boldsymbol{J}^{[0]}(\boldsymbol{e}(k+1)) \right\}, \quad (16)$$

where $J^{[0]}(e(k+1)) = \hat{\theta}\Upsilon(e(k+1))$. The performance index function can be updated as

$$J^{[1]}(\boldsymbol{e}(k)) = \boldsymbol{U}(\boldsymbol{e}(k), w^{[0]}(k)) + \boldsymbol{J}^{[0]}(\boldsymbol{e}(k+1)). \quad (17)$$

For $i = 1, 2, \ldots$, $\forall \boldsymbol{e}(k)$, the iterative ADP algorithm will iterate between

$$w^{[i]}(\boldsymbol{e}(k)) = \arg\inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \boldsymbol{J}^{[i]}(\boldsymbol{e}(k+1)) \right\}, \quad (18)$$

and the iterative performance index functions

$$\boldsymbol{J}^{[i+1]}(\boldsymbol{e}(k)) = \inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + J^{[i]}(\boldsymbol{e}(k+1)) \right\}. \quad (19)$$

In fact, the accurate iterative control policy $w^{[i]}(k)$ and the iterative performance index function $J^{[i]}(e(k))$ are generally impossible to obtain. For example, if neural networks are used to implement the iterative ADP algorithm, no matter what kind of neural networks we choose, the approximate error between the output of the neural networks and the expect output must exist. In fact, as the existence of the approximation error, the accurate iterative control policy of the chaotic systems cannot generally be obtained. Thus, the iterative ADP algorithm with finite approximation error is expressed as follows:

$$\hat{w}^{[0]}(\boldsymbol{e}(k)) = \arg\inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \hat{\boldsymbol{J}}^{[0]}(\boldsymbol{e}(k+1)) \right\} + \boldsymbol{\alpha}^{[0]}(\boldsymbol{e}(k)), \quad (20)$$

where $\hat{J}^{[0]}(e(k+1)) = \hat{\theta}\Upsilon(e(k+1))$. The performance index function can be updated as

$$\hat{\boldsymbol{J}}^{[1]}(\boldsymbol{e}(k)) = \boldsymbol{U}(\boldsymbol{e}(k), \hat{w}^{[0]}(k)) + \hat{\boldsymbol{J}}^{[0]}(\boldsymbol{e}(k+1))) + \boldsymbol{\beta}^{[0]}(\boldsymbol{e}(k)), \quad (21)$$

where $\alpha^{[0]}(\boldsymbol{e}(k))$ and $\beta^{[0]}(\boldsymbol{e}(k))$ are the approximation error functions of the iterative control and iterative performance index function, respectively.

For $i = 1, 2, \ldots$, the iterative ADP algorithm will iterate between

$$\hat{w}^{[i]}(\boldsymbol{e}(k)) = \arg\inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \hat{\boldsymbol{J}}^{[i]}(\boldsymbol{e}(k+1)) \right\} + \boldsymbol{\alpha}^{[i]}(\boldsymbol{e}(k)), \quad (22)$$

and the performance index function

$$\hat{\boldsymbol{J}}^{[i+1]}(\boldsymbol{e}(k)) = \inf_{\boldsymbol{w}(k)} \left\{ \boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \hat{\boldsymbol{J}}^{[i]}(\boldsymbol{e}(k+1)) \right\} + \boldsymbol{\beta}^{[i]}(\boldsymbol{e}(k)), \quad (23)$$

where $\boldsymbol{\alpha}^{[i]}(\boldsymbol{e}(k))$ and $\boldsymbol{\beta}^{[i]}(\boldsymbol{e}(k))$ are the approximation error functions of the iterative control and iterative performance index function, respectively.

Remark 2 The proposed approximation-error-ADP-based method for chaotic systems is the development of the traditional ADP method. In the traditional ADP method, the approximate error exists in the process of implementation. However, the analysis of the traditional ADP method has no regard for the approximate error. Therefore, in this paper, the approximate error is considered in the algorithm analysis, and the corresponding convergence proof is given. In the following theorems, we will prove that the iterative performance index function with approximate error is convergent if some conditions are satisfied.

3.2. Convergence analysis of the iterative ADP algorithm

In fact, the iterative index $i \to \infty$, the boundary of iterative approximation error will also increase to infinity, although the approximation error is finite in the single iteration.[6] The following theorem will show this property.

Theorem 1 Let $e(k)$ be an arbitrary controllable state. For $i = 1, 2, \ldots$, define a new iterative performance index function as

$$\Delta^{[i]}(\boldsymbol{e}(k)) = \inf_{\boldsymbol{w}(k)} \{\boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \hat{\boldsymbol{J}}^{[i-1]}(\boldsymbol{e}(k+1))\}, \quad (24)$$

where $\boldsymbol{w}(k)$ can be obtained accurately. If the initial iterative performance index function $\hat{\boldsymbol{J}}^{[0]}(\boldsymbol{e}(k)) = \Delta^{[0]}(\boldsymbol{e}(k))$, and there exists a finite constant ε_1 that makes

$$|\hat{\boldsymbol{J}}^{[i]}(\boldsymbol{e}(k)) - \Delta^{[i]}(\boldsymbol{e}(k))| \le \varepsilon_1 \quad (25)$$

hold uniformly, then we have

$$|\hat{\boldsymbol{J}}^{[i]}(\boldsymbol{e}(k)) - \boldsymbol{J}^{[i]}(\boldsymbol{e}(k))| \le \mathrm{i}\varepsilon_1, \quad (26)$$

where ε_1 is called uniform finite approximation error.

Proof The theorem can be proved by mathematical induction. First, let $i = 1$, we have

$$\Delta^{[1]}(e(k)) = \inf_{\boldsymbol{w}(k)} \{\boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) + \hat{\boldsymbol{J}}^{[0]}(\boldsymbol{e}(k+1))\}$$

$$= J^{[1]}(\boldsymbol{e}(k)). \tag{27}$$

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

$$|\hat{J}^{[1]}(\boldsymbol{e}(k)) - J^{[1]}(\boldsymbol{e}(k))| \leq \varepsilon_1. \tag{28}$$

Assume that Eq. (26) holds for $i-1$. Then, for i, we have

$$\begin{aligned}\Delta^{[i]}(\boldsymbol{e}(k)) &= \inf_{\boldsymbol{w}(k)}\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \hat{J}^{[i-1]}(\boldsymbol{e}(k+1))\}\\ &\leq \inf_{\boldsymbol{w}(k)}\{U(\boldsymbol{e}(k),\boldsymbol{w}(k))\\ &\quad + J^{[i-1]}(\boldsymbol{e}(k+1)) + (i-1)\varepsilon_1\}\\ &= J^{[i]}(\boldsymbol{e}(k)) + (i-1)\varepsilon_1. \end{aligned}\tag{29}$$

On the other hand, we can have

$$\begin{aligned}\Delta^{[i]}(\boldsymbol{e}(k)) &= \inf_{\boldsymbol{w}(k)}\{U(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \hat{J}^{[i-1]}(\boldsymbol{e}(k+1))\}\\ &\geq \inf_{\boldsymbol{w}(k)}\{U(\boldsymbol{e}(k),\boldsymbol{w}(k))\\ &\quad + J^{[i-1]}(\boldsymbol{e}(k+1)) - (i-1)\varepsilon_1\}\\ &= J^{[i]}(\boldsymbol{e}(k)) - (i-1)\varepsilon_1. \end{aligned}\tag{30}$$

Thus, we have

$$-(i-1)\varepsilon_1 \leq \Delta^{[i]}(\boldsymbol{e}(k)) - J^{[i]}(\boldsymbol{e}(k)) \leq (i-1)\varepsilon_1. \tag{31}$$

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

$$|\hat{J}^{[i]}(\boldsymbol{e}(k)) - J^{[i]}(\boldsymbol{e}(k))| \leq i\varepsilon_1. \tag{32}$$

It is obvious that the property analysis of the iterative performance index function $\hat{J}^{[i]}(\boldsymbol{e}(k))$ and iterative control policy $\hat{w}^{[i]}(k)$ are very difficult. In next part, the novel convergence analysis is built.

Theorem 2 Let $\boldsymbol{e}(k)$ be an arbitrary controllable state. For $\forall i = 0,1,\ldots$, let $\Delta^{[i]}(\boldsymbol{e}(k))$ be expressed as (24) and $\hat{J}^{[i]}(\boldsymbol{e}(k))$ be expressed as (23). Let both $\zeta < \infty$ and $1 \leq \eta < \infty$ be constants that make

$$J^*(\boldsymbol{e}(k+1)) \leq \zeta \boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)), \tag{33}$$

$$J^{[0]}(\boldsymbol{e}(k)) \leq \eta J^*(\boldsymbol{e}(k)) \tag{34}$$

hold uniformly. If there exists $1 \leq \iota < \infty$ that makes

$$\hat{J}^{[i]}(\boldsymbol{e}(k)) \leq \iota \Delta^{[i]}(\boldsymbol{e}(k)) \tag{35}$$

hold uniformly, then we have

$$\hat{J}^{[i]}(\boldsymbol{e}(k)) \leq \iota\left(1 + \sum_{j=1}^{i} \frac{\zeta^j \iota^{j-1}(\iota-1)}{(\zeta+1)^j} + \frac{\zeta^i \iota^i(\eta-1)}{(\zeta+1)^i}\right) J^*(\boldsymbol{e}(k)), \tag{36}$$

where, we define $\sum_{j}^{i}(\cdot) = 0$, for $\forall j > i$ and $i, j = 0,1,\ldots$.

Proof The theorem can be proved by mathematical induction. First, let $i = 0$. Then, equation (36) becomes

$$\hat{J}^{[0]}(\boldsymbol{e}(k)) \leq \iota\eta J^*(\boldsymbol{e}(k)). \tag{37}$$

As $\hat{J}^{[0]}(\boldsymbol{e}(k)) \leq \eta J^*(\boldsymbol{e}(k))$, then we can obtain $\hat{J}^{[0]}(\boldsymbol{e}(k)) \leq \eta J^*(\boldsymbol{e}(k)) \leq \iota\eta J^*(\boldsymbol{e}(k))$, which obtains (37). Thus, the conclusion holds for $i = 0$.

Next, let $i = 1$. According to Eq. (27), we have

$$\begin{aligned}\Delta^{[1]}(\boldsymbol{e}(k)) &= \inf_{\boldsymbol{w}(k)}\left\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \hat{J}^{[0]}(\boldsymbol{e}(k+1))\right\}\\ &\leq \inf_{\boldsymbol{w}(k)}\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \iota\eta J^*(\boldsymbol{e}(k+1))\}. \end{aligned}\tag{38}$$

As $1 \leq \iota < \infty$ and $1 \leq \eta < \infty$, then $\iota\eta - 1 \geq 0$. According to Eq. (33), we have

$$\begin{aligned}&\Delta^{[1]}(\boldsymbol{e}(k))\\ &\leq \inf_{\boldsymbol{w}(k)}\left\{\left(1 + \zeta\frac{\iota\eta-1}{\zeta+1}\right)\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k))\right.\\ &\quad \left. + \left(\iota\eta - \frac{\iota\eta-1}{\zeta+1}\right)\boldsymbol{J}^*(\boldsymbol{e}(k+1))\right\}\\ &= \left(1 + \zeta\frac{\iota\eta-1}{\zeta+1}\right)\inf_{\boldsymbol{w}(k)}\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \boldsymbol{J}^*(\boldsymbol{e}(k+1))\}\\ &= \left(1 + \frac{\zeta(\iota-1)}{\zeta+1} + \frac{\zeta\iota(\eta-1)}{\zeta+1}\right)\boldsymbol{J}^*(\boldsymbol{e}(k)). \end{aligned}\tag{39}$$

According to Eq. (35), we can obtain

$$\hat{J}^{[1]}(\boldsymbol{e}(k)) \leq \iota\left(1 + \frac{\zeta(\iota-1)}{\zeta+1} + \frac{\zeta\iota(\eta-1)}{\zeta+1}\right)\boldsymbol{J}^*(\boldsymbol{e}(k)), \tag{40}$$

which shows that equation (36) holds for $i = 1$.

Assume that equation (36) holds for $i-1$. Then, for i, we have

$$\begin{aligned}&\Delta^{[i]}(\boldsymbol{e}(k))\\ &= \inf_{\boldsymbol{w}(k)}\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \hat{J}^{[i-1]}(\boldsymbol{e}(k+1))\}\\ &\leq \inf_{\boldsymbol{w}(k)}\left\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \iota\left(1 + \sum_{j=1}^{i}\frac{\zeta^j\iota^{j-1}(\iota-1)}{(\zeta+1)^j}\right.\right.\\ &\quad \left.\left. + \frac{\zeta^{i-1}\iota^{i-1}(\eta-1)}{(\zeta+1)^{i-1}}\right)J^*(\boldsymbol{e}(k+1))\right\}. \end{aligned}\tag{41}$$

Hence, equation (41) can be rewritten as

$$\begin{aligned}&\Delta^{[i]}(\boldsymbol{e}(k))\\ &\leq \inf_{\boldsymbol{w}(k)}\left\{\left(1 + \zeta\sum_{j=1}^{i-1}\frac{\zeta^{j-1}\iota^{j-1}(\iota-1)}{(\zeta+1)^j}\right.\right.\\ &\quad \left. + \frac{\zeta^{i-1}\iota^{i-1}(\iota\eta-1)}{(\zeta+1)^{i-1}}\right)\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k))\\ &\quad + \left(\iota\left(1 + \sum_{j=1}^{i-1}\frac{\zeta^j\iota^{j-1}(\iota-1)}{(\zeta+1)^j} + \frac{\zeta^{i-1}\iota^{i-1}(\eta-1)}{(\zeta+1)^{i-1}}\right)\right.\\ &\quad - \left(\sum_{j=1}^{i-1}\frac{\zeta^{j-1}\iota^{j-1}(\iota-1)}{(\zeta+1)^j}\right.\\ &\quad \left.\left. + \frac{\zeta^{i-1}\iota^{i-1}(\iota\eta-1)}{(\zeta+1)^{i-1}}\right)\right)J^*(\boldsymbol{e}(k+1))\Bigg\}\\ &= \left(1 + \sum_{j=1}^{i}\frac{\zeta^j\iota^{j-1}(\iota-1)}{(\zeta+1)^j} + \frac{\zeta^i\iota^i(\eta-1)}{(\zeta+1)^i}\right) \end{aligned}$$

$$\times \inf_{\boldsymbol{w}(k)} \{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \boldsymbol{J}^*(\boldsymbol{e}(k+1))\}$$
$$= \left(1 + \sum_{j=1}^{i} \frac{\zeta^j \iota^{j-1}(\iota-1)}{(\zeta+1)^j} + \frac{\zeta^i \iota^i(\eta-1)}{(\zeta+1)^i}\right) \boldsymbol{J}^*(\boldsymbol{e}(k)). \quad (42)$$

Then, according to Eq. (35), we can obtain Eq. (36) which proves the conclusion for $\forall i = 0, 1, \ldots$.

Theorem 3 Let $\boldsymbol{e}(k)$ be an arbitrary controllable state. Suppose Theorem 2 holds for $\forall \boldsymbol{e}(k)$. If for $\zeta < \infty$ and $\iota \geq 1$, the inequality

$$\iota < \frac{\zeta+1}{\zeta} \quad (43)$$

holds, then as $i \to \infty$, the iterative performance index function $\hat{J}^{[i]}(\boldsymbol{e}(k))$ in the iterative ADP algorithm (Eqs. (20)–(23) is uniformly convergent into a bounded neighborhood of the optimal performance index function $J^*(\boldsymbol{e}(k))$, i.e.,

$$\lim_{i\to\infty} \hat{J}^{[i]}(\boldsymbol{e}(k)) = \hat{J}^{\infty}(\boldsymbol{e}(k)) \leq \iota \left(1 + \frac{\zeta(\iota-1)}{1-\zeta(\iota-1)}\right) J^*(\boldsymbol{e}(k)). \quad (44)$$

Proof According to (42) in Theorem 2, we can see that for $j = 1, 2, \ldots$, the sequence $\left\{\frac{\zeta^j \iota^{j-1}(\iota-1)}{(\zeta+1)^j}\right\}$ is a geometrical series. Then, equation (42) can be written as

$$\Delta^{[i]}(\boldsymbol{e}(k)) \leq \left(1 + \frac{\frac{\zeta(\iota-1)}{\zeta+1}\left(1 - \left(\frac{\zeta\iota}{\zeta+1}\right)^i\right)}{1 - \frac{\zeta\iota}{\zeta+1}} + \frac{\zeta^i \iota^i(\eta-1)}{(\zeta+1)^i}\right) J^*(\boldsymbol{e}(k)). \quad (45)$$

As $i \to \infty$, if $1 \leq \iota < (\zeta+1)/\zeta$, then Eq. (45) becomes

$$\lim_{i\to\infty} \Delta^{[i]}(\boldsymbol{e}(k)) = \Delta^{\infty}(\boldsymbol{e}(k)) \leq \left(1 + \frac{\zeta(\iota-1)}{1-\zeta(\iota-1)}\right) \boldsymbol{J}^*(\boldsymbol{e}(k)). \quad (46)$$

According to Eq. (35), let $i \to \infty$, we have

$$\hat{J}^{\infty}(\boldsymbol{e}(k)) \leq \iota \Delta^{\infty}(\boldsymbol{e}(k)). \quad (47)$$

Considering Eqs. (47) and (46), we can obtain

$$\hat{J}^{\infty}(\boldsymbol{e}(k)) \leq \iota \left(1 + \frac{\zeta(\iota-1)}{1-\zeta(\iota-1)}\right) \boldsymbol{J}^*(\boldsymbol{e}(k)). \quad (48)$$

Furthermore, we can obtain the following corollary.

Corollary 1 Let $\boldsymbol{e}(k)$ be an arbitrary controllable state. Suppose Theorem 2 holds for $\forall \boldsymbol{e}(k)$. If for $\zeta < \infty$ and $\iota \geq 1$, the inequality (43) holds, then the iterative control policy $\hat{w}^{[i]}(k)$ of the iterative ADP algorithm (20)–(23) is convergent, i.e.,

$$\hat{w}^{\infty}(k) = \lim_{i\to\infty} \hat{w}^{[i]}(k) = \arg \inf_{\boldsymbol{w}(k)} \left\{\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) + \hat{J}^{\infty}(\boldsymbol{e}(k+1))\right\}. \quad (49)$$

Therefore, it is proved that the new iterative ADP algorithm with finite approximation error is convergent. In the next section, the examples will be given to illustrate the performance of the proposed method for chaotic systems.

4. Simulation study

To evaluate the performance of our iterative ADP algorithm with finite approximation error, we choose two examples with quadratic utility functions.

Example 1 Consider the following Hénon mapping as follows:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 + bx_2(k) - ax_1^2(k) \\ x_1(k) \end{bmatrix} + \begin{bmatrix} 0.2 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \quad (50)$$

where $a = 1.4$, $b = 0.3$, $x_1(0) = 0.1$, and $x_2(0) = -0.1$. The chaotic signal orbits are given by Fig. 1.

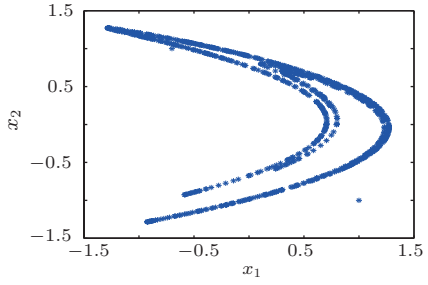


Fig. 1. (color online) The Hénon chaotic orbits.

The desired orbit $x_d(k)$ is generated by the following expression:

$$\boldsymbol{x}_d(k+1) = \boldsymbol{A}\boldsymbol{x}_d(k), \quad (51)$$

with

$$\boldsymbol{A} = \begin{bmatrix} \cos wT & \sin wT \\ -\sin wT & \cos wT \end{bmatrix},$$

where $T = 0.1$ s and $w = 0.8\pi$. The initial state is selected as $[1, \quad 0.5]^{\mathrm{T}}$, and the utility function is defined as $\boldsymbol{U}(\boldsymbol{e}(k),\boldsymbol{w}(k)) = \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{e}(k) + \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{R}\boldsymbol{w}(k)$, where $\boldsymbol{Q} = \boldsymbol{R} = \boldsymbol{I}_2$. Let $\hat{\theta} = 6$ and $\Upsilon(\boldsymbol{e}(k)) = \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{e}(k)$ to initialize the algorithm. For illustrating the performance of the proposed

optimal tracking control scheme, two different approximate precisions, i.e., $\varepsilon_1 = 10^{-2}$ and $\varepsilon_1 = 10^{-8}$ are selected. Then, for $\varepsilon_1 = 10^{-2}$, we obtain the performance index function trajectory as shown in Fig. 2. In Fig. 2, the iterative performance index function is not convergent. The state tracking trajectories are shown in Figs. 3 and 4, and the tracking error trajectories are as shown in Fig. 5. We can see that the iterative control cannot make the system track the desired orbit satisfactorily. For $\varepsilon_1 = 10^{-8}$, the convergence conditions are satisfied, and the performance index function is shown in Fig. 6. The state trajectories are shown in Figs. 7 and 8, and the tracking error trajectories are as shown in Fig. 9. From the results of the simulation, we can see that the optimal tracking control of the chaotic systems can be obtained if the convergence conditions are satisfied.

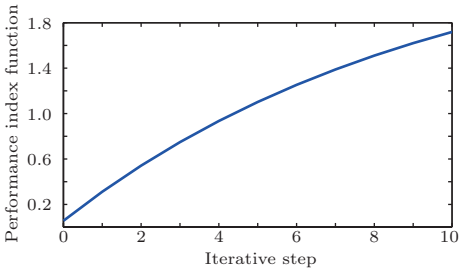


Fig. 2. (color online) The performance index function trajectory for $\varepsilon_1 = 10^{-2}$.

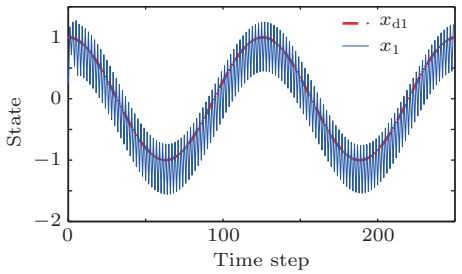


Fig. 3. (color online) The trajectories of x_1 and x_{d1} for $\varepsilon_1 = 10^{-2}$.

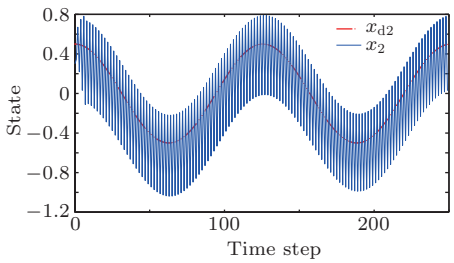


Fig. 4. (color online) The trajectories of x_2 and x_{d2} for $\varepsilon_1 = 10^{-2}$.

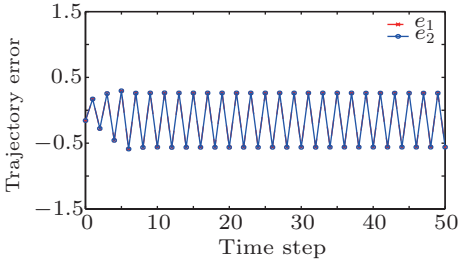


Fig. 5. (color online) The trajectories of e_1 and e_2 for $\varepsilon_1 = 10^{-2}$.

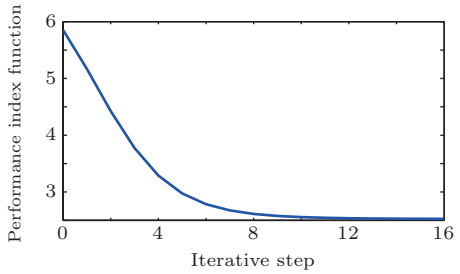


Fig. 6. (color online) The performance index function trajectory for $\varepsilon_1 = 10^{-8}$.

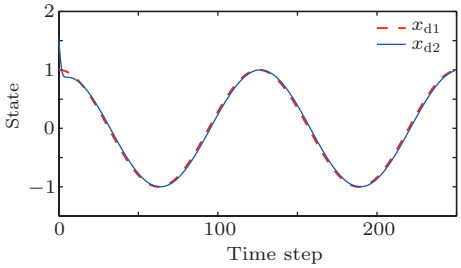


Fig. 7. (color online) The trajectories of x_1 and x_{d1} for $\varepsilon_1 = 10^{-8}$.

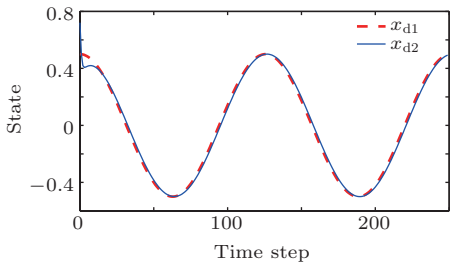


Fig. 8. (color online) The trajectories of x_2 and x_{d2} for $\varepsilon_1 = 10^{-8}$.

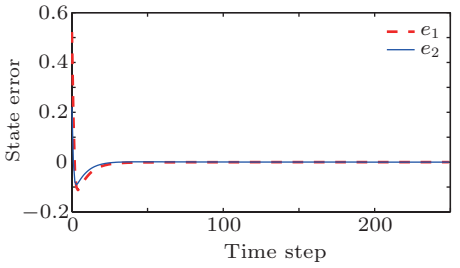


Fig. 9. (color online) The trajectories of e_1 and e_2 for $\varepsilon_1 = 10^{-8}$.

Example 2

Consider the following continuous time chaotic system:[10]

$$\begin{aligned} \dot{x}_1 &= \theta_1(x_2 - h(x_1)) + u_1, \\ \dot{x}_2 &= x_1 - x_2 + x_3 + u_2, \\ \dot{x}_3 &= -\theta_2 x_2 + u_3, \end{aligned} \tag{52}$$

where $h(x_1) = m_1 x_1 + (1/2)(m_0 - m_1)(|x_1 + \theta_3| - |x_1 - \theta_3|)$, $\theta_1 = 9$, $\theta_2 = 14.28$, $\theta_3 = 1$, $m_0 = -1/7$, and $m_1 = 2/7$.

According to Euler's discretization method, the continuous time chaotic system can be represented as follows:

$$x_1(k+1) = x_1(k) + \hat{T}(\theta_1(x_2(k) - h(x_1(k))) + u_1(k)),$$

$$x_2(k+1) = x_2(k)+\hat{T}(x_1(k)-x_2(k)+x_3(k)+u_2(k)),$$
$$x_3(k+1) = x_3(k)+\hat{T}(-\theta_2 x_2(k)+u_3(k)), \quad (53)$$

where $\hat{T} = 0.001$. The discrete chaotic system is given in Fig. 10.

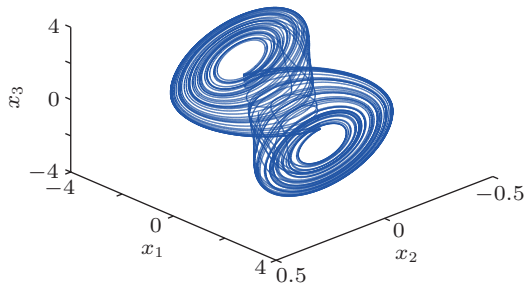


Fig. 10. (color online) The chaotic attractor of system (53).

The initial state is selected as $[0.2, 0.13, 0.17]^{\mathrm{T}}$, and the reference trajectory is $[0.5, -0.5, 0.4]^{\mathrm{T}}$. The utility function is defined as $\boldsymbol{U}(\boldsymbol{e}(k), \boldsymbol{w}(k)) = \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{e}(k) + \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{R}\boldsymbol{w}(k)$, where $\boldsymbol{Q} = \boldsymbol{R} = \boldsymbol{I}_3$. Let $\hat{\theta} = 6$ and $\Upsilon(\boldsymbol{e}(k)) = \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{e}(k)$ to initialize the algorithm. The performance index function is shown in Fig. 11. The system states are shown as in Fig. 12. The state errors are given in Fig. 13. It can be seen that the proposed algorithm is effective and the simulation results are satisfactory.

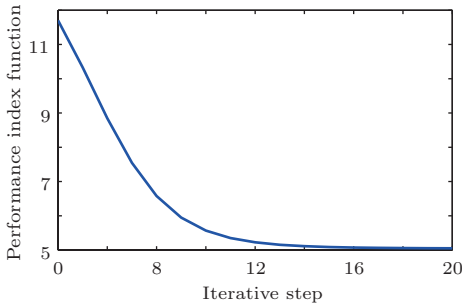


Fig. 11. (color online) The performance index function trajectory for $\varepsilon_1 = 10^{-8}$.

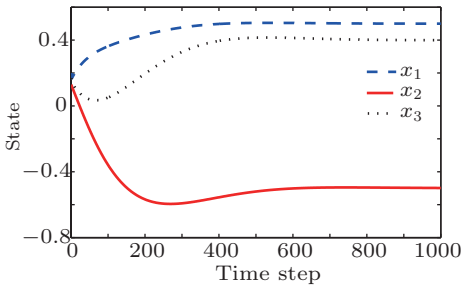


Fig. 12. (color online) The trajectories of x_1, x_2, and x_3 for $\varepsilon_1 = 10^{-8}$.

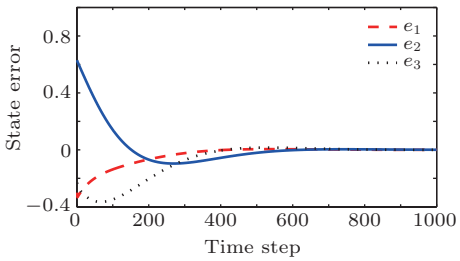


Fig. 13. (color online) The trajectories of e_1, e_2, and e_3 for $\varepsilon_1 = 10^{-8}$.

5. Conclusion

This paper proposed an optimal tracking control method for chaotic systems. Via system transformation, the optimal tracking problem was transformed into an optimal regulation problem, and then the approximation-error ADP algorithm was introduced to deal with the optimal regulation problem with rigorous convergence analysis. Finally, two examples have been given to demonstrate the validity of the proposed optimal tracking control scheme for chaotic systems.

References

[1] Balakrishnan S and Biega V 1996 *J. Guid. Control Dynam.* **19** 893
[2] Chen S and Lu J 2002 *Chaos Soliton. Fract.* **14** 643
[3] Enns R and Si J 2003 *IEEE Tran. Neural Netw.* **14** 929
[4] Héno M 1976 *Commun. Math. Phys.* **50** 69
[5] Lewis F and Vrabie D 2009 *IEEE Circ. Syst. Mag.* **9** 32
[6] Liu D and Wei Q 2012 *IEEE Tran. Syst. Man Cyber. B* **43** 779
[7] Lu J, Wu X, Lü J and Kang L 2004 *Chaos Soliton. Fract.* **22** 311
[8] Ma T and Fu J 2011 *Chin. Phys. B* **20** 050511
[9] Ma T, Fu J and Sun Y 2010 *Chin. Phys. B* **19** 090502
[10] Ma T, Zhang H and Fu J 2008 *Chin. Phys. B* **17** 4407
[11] Murray J, Cox C, Lendaris G and Saeks R 2002 *IEEE Tran. Syst. Man Cyber. B* **32** 140
[12] Si J and Wang Y 2001 *IEEE Tran. Neural Netw.* **12** 264
[13] Vamvoudakis K and Lewis F 2010 *Automatica* **46** 878
[14] Wang F, Jin N, Liu D and Wei Q 2011 *IEEE Tran. Neural Netw.* **22** 24
[15] Wang F, Zhang H and Liu D 2009 *IEEE Computational Intelligence Magazine* **4** 39
[16] Werbos P 1991 *Neural Network Control* (edited by Miller W T, Sutton R S and Werbos P J) (Cambridge: MIT Press) pp. 67–95
[17] Werbos P 1992 *Approximate Dynamic Programming for Real-Time Control and Neural Modelingin Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches* (edited by White D A and Sofge D A) (New York: Van Nostrand Reinhold) ch. 13
[18] Zhang H, Cui L, Zhang X and Luo Y 2011 *IEEE Tran. Neural Netw.* **22** 2226
[19] Zhang H, Huang W, Wang Z and Chai T 2006 *Phys. Lett. A* **350** 363
[20] Zhang H, Ma T, Fu J and Tong S 2009 *Chin. Phys. B* **18** 3751
[21] Zhang H, Song R, Wei Q and Zhang T 2011 *IEEE Tran. Neural Netw.* **22** 1851
[22] Zhang H, Wang Z and Liu D 2004 *Int. J. Bifur. Chaos* **14** 1
[23] Zhang H, Wei Q and Liu D 2011 *Automatica* **47** 207
[24] Zhang H, Wei Q and Luo Y 2008 *IEEE Tran. Syst. Man Cyber. B* **38** 937