Chaotic system optimal tracking using data-based synchronous method with unknown dynamics and disturbances*

Ruizhuo Song(宋睿卓)[1] and Qinglai Wei(魏庆来)[2,†]

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

(Received 9 October 2016; revised manuscript received 18 December 2016; published online January 26 2017)

We develop an optimal tracking control method for chaotic system with unknown dynamics and disturbances. The method allows the optimal cost function and the corresponding tracking control to update synchronously. According to the tracking error and the reference dynamics, the augmented system is constructed. Then the optimal tracking control problem is defined. The policy iteration (PI) is introduced to solve the min-max optimization problem. The off-policy adaptive dynamic programming (ADP) algorithm is then proposed to find the solution of the tracking Hamilton–Jacobi–Isaacs (HJI) equation online only using measured data and without any knowledge about the system dynamics. Critic neural network (CNN), action neural network (ANN), and disturbance neural network (DNN) are used to approximate the cost function, control, and disturbance. The weights of these networks compose the augmented weight matrix, and the uniformly ultimately bounded (UUB) of which is proven. The convergence of the tracking error system is also proven. Two examples are given to show the effectiveness of the proposed synchronous solution method for the chaotic system tracking problem.

Keywords: adaptive dynamic programming, approximate dynamic programming, chaotic system, zero-sum

PACS: 05.45.Gg **DOI:** 10.1088/1674-1056/26/3/030505

1. Introduction

Chaotic system has complex nonlinear dynamics and its response exhibits some specific characteristics such as sensitivity to the initial condition, broad Fourier transform spectra, and irregular identities of the motion in phase space.[1–4] An early development in the history of chaotic systems dates back to 1963 in which an American meteorologist Lorenz tried to bring forward system equations to simulate the weather changes. The Lorenz system, as the first chaotic model, revealed the complex and fundamental behaviors of the nonlinear dynamical systems.[5] And then the concept of the generalized Lorenz system was extended by Lü, and a class of generalized Lorenz-like systems was discussed.[6] Until now, many efficient approaches have been proposed for controlling chaotic systems, such as impulsive control method,[7–9] adaptive dynamic programming (ADP) method,[10] and neural adaptive control.[11] Most of these methodologies consider the chaotic system without disturbance.

In this paper, we provide a new method to design an optimal tracking controller for chaotic systems with unknown dynamics and disturbance based on ADP algorithms. ADP characterized by strong abilities of self-learning and adaptivity has received significantly increased attention and becomes an important brainlike intelligent optimal control method for nonlinear systems.[12–15] ADP algorithms include value iteration (VI) and policy iteration (PI) according to the different iterative methods.[16,17] In Ref. [18], a complex-valued ADP algorithm was discussed, where for the first time the optimal control problem of complex-valued nonlinear systems was successfully solved by PI. In Ref. [19], based on neurocognitive psychology, a novel controller based on multiple actor-critic structures was developed for unknown systems and the proposed controller traded off fast actions based on stored behavior patterns with real-time exploration using current input–output data. In Ref. [20], an effective off-policy learning based integral reinforcement learning (IRL) algorithm was presented, which successfully solved the optimal control problem for completely unknown continuous-time systems with unknown disturbances. Note that the development of ADP promotes the development of adaptive control and neural networks control. For example, in Ref. [21], a novel adaptive nonlinear controller was designed to achieve stochastic synchronization of complex networks, which is less conservative and may be more widely used than the traditional adaptive linear controller. In Ref. [22], the synchronization control of memristor-based recurrent neural networks with impulsive perturbations or boundary perturbations was studied. Two kinds of controllers were designed so that the memristive neural networks with perturbations could converge to the equilibrium points, which evoke human's memory patterns. In Ref. [23], pinning adaptive synchronization for uncertain complex dynamic networks with multi-link against network dete-

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61304079, 61673054, and 61374105), the Fundamental Research Funds for the Central Universities, China (Grant No. FRF-TP-15-056A3), and the Open Research Project from SKLMCCS, China (Grant No. 20150104).

†Corresponding author. E-mail: qinglai.wei@ia.ac.cn

© 2017 Chinese Physical Society and IOP Publishing Ltd

rioration was proposed, new synchronization criteria for networks with multi-link were derived to ensure the synchronized states to be local or global stable with uncertainty and deterioration. It is worth noting that the ADP algorithms have been successfully applied to control chaotic systems. In Ref. [24], an optimal tracking control scheme was proposed for a class of discrete-time chaotic systems using the approximation-error-based ADP algorithm. In that work, the system was defined to be known and the disturbance was not considered. But it lays a foundation of ADP and chaotic system control.

It is known that H_∞ optimal control has been extensively used in the effort to design controllers for nonlinear systems with disturbances.[25] In Ref. [26], significant insight into the design of H_∞ control problems has been provided, after it was formulated as a min-max two-player zero-sum (ZS) game problem. The optimal control in such a scenario is equivalent to finding the Nash equilibrium of the corresponding two-player zero-sum game,[27] which results in solving the so-called Hamilton–Jacobi–Isaacs (HJI) equation. During the last few years, strong connections between ADP and the optimal control have prompted a major effort towards developing reinforcement learning algorithms to learn the solution to the HJI equation arising in the H_∞ optimal regulation problem. In Refs. [28] and [29], the ZS differential games were discussed in the framework of ADP. In Ref. [30], multiperson ZS differential games for a class of uncertain nonlinear systems were studied. In Ref. [31], multiperson non-zero-sum differential games were presented by the off-policy IRL method. The previous research supplies us with a kind of new perspective for optimal tracking control of chaotic system with disturbances.

Based on the previous research works, this paper studies the chaotic system optimal tracking problem with unknown dynamics and disturbances. First an augmented system from the tracking error dynamics and the reference dynamics is constructed and a new cost function is introduced for the optimal tracking problem. The tracking control problem is then transformed to a min-max optimization problem. The PI method is introduced to obtain the iterative cost function using the system dynamics. The off-policy ADP algorithm is then developed to find the solution of the tracking HJI equation online using the measured data and without any knowledge about the system dynamics. The critic neural network (CNN), action neural network (ANN), and disturbance neural network (DNN) are used to approximate the cost function, control, and disturbance. The neural network (NN) implementation is given with convergence analyses. At last, two examples are given, and the effectiveness of the proposed synchronous solution method for optimal tracking control problem of chaotic systems is shown by the simulation results.

The rest of this paper is organized as follows. In Section 2, we present the motivations and preliminaries of the discussed problem. In Section 3, the synchronous solution is developed. In Section 4, the NN implementation is given with convergence analyses. In Section 5, two examples are given to demonstrate the effectiveness of the proposed scheme. In Section 6, the conclusion is drawn.

2. Problem formulation

Let us consider the chaotic system with disturbance described by

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

where $\boldsymbol{x} \in \mathbf{R}^n$ is the chaotic system state, $\boldsymbol{u} \in \mathbf{R}^{m_1}$ is the control, $\boldsymbol{w} \in \mathbf{R}^{m_2}$ is the disturbance, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbf{R}^n$ and $\boldsymbol{g}(\boldsymbol{x}) \in \mathbf{R}^{n\times m_1}$ are unknown system dynamics with $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$, and $\boldsymbol{h}(\boldsymbol{x}) \in \mathbf{R}^{n\times m_2}$ is the unknown disturbance gain. Actually, many nonlinear chaotic dynamical systems can be expressed as Eq. (1), such as Lü system,[32] Chen system,[33,34] Lorenz system,[35,36] several variants of Chua's circuits,[37,38] and Duffing oscillator.[39,40]

Let x_d be the constant reference trajectory and we have

$$\dot{\boldsymbol{x}}_\mathrm{d} = \mathbf{0}. \tag{2}$$

The focus of this paper is to find an optimal control $\boldsymbol{u}(\boldsymbol{x})$. It makes the chaotic system (1) track the given trajectory $\boldsymbol{x}_\mathrm{d}$. And the control input $\boldsymbol{u}(\boldsymbol{x})$ makes a cost function optimal. Therefore, the tracking error system is first defined as

$$\dot{\boldsymbol{e}} = \boldsymbol{F}(\boldsymbol{e}) + \boldsymbol{G}(\boldsymbol{e})\boldsymbol{\alpha}(\boldsymbol{e}) + \boldsymbol{H}(\boldsymbol{e})\boldsymbol{\gamma}(\boldsymbol{e}), \tag{3}$$

where the tracking error $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_\mathrm{d}$, $\boldsymbol{F}(\boldsymbol{e}) = \boldsymbol{f}(\boldsymbol{e}+\boldsymbol{x}_\mathrm{d})$, $\boldsymbol{G}(\boldsymbol{e}) = \boldsymbol{g}(\boldsymbol{e}+\boldsymbol{x}_\mathrm{d})$, and $\boldsymbol{H}(\boldsymbol{e}) = \boldsymbol{h}(\boldsymbol{e}+\boldsymbol{x}_\mathrm{d})$. $\boldsymbol{\alpha}(\boldsymbol{e}) = \boldsymbol{u}(\boldsymbol{e}+\boldsymbol{x}_\mathrm{d})$ and $\boldsymbol{\gamma}(\boldsymbol{e}) = \boldsymbol{w}(\boldsymbol{e}+\boldsymbol{x}_\mathrm{d})$ are the control and disturbance of system (2). In this paper, we assume that $\boldsymbol{F}(\mathbf{0}) = \mathbf{0}$ and $\boldsymbol{F}(\boldsymbol{e})$ is lipschitz. In the tracking error system (3), $\boldsymbol{\gamma}$ is the disturbance, which can be seen as another input, and makes the cost function maximum. Then ZS differential game will be adopted to the optimal tracking control problem. The cost function is defined as

$$\bar{J}(\boldsymbol{e},\boldsymbol{\alpha},\boldsymbol{\gamma}) = \int_t^\infty (\boldsymbol{e}^\mathrm{T}\bar{\boldsymbol{Q}}\boldsymbol{e} + \boldsymbol{\alpha}^\mathrm{T}\boldsymbol{R}\boldsymbol{\alpha} - \boldsymbol{\gamma}^\mathrm{T}\boldsymbol{S}\boldsymbol{\gamma})\,\mathrm{d}\tau, \tag{4}$$

where $\bar{\boldsymbol{Q}}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ are positive definite matrices.

Putting Eqs. (2) and (3) together yields the augmented system

$$\dot{\boldsymbol{\chi}} = \boldsymbol{\xi}(\boldsymbol{\chi}) + \boldsymbol{\mu}(\boldsymbol{\chi})\boldsymbol{\alpha} + \boldsymbol{\nu}(\boldsymbol{\chi})\boldsymbol{\gamma}, \tag{5}$$

where $\boldsymbol{\chi} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{x}_\mathrm{d} \end{bmatrix}$, $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{F} \\ \mathbf{0} \end{bmatrix}$, $\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{G} \\ \mathbf{0} \end{bmatrix}$, and $\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{H} \\ \mathbf{0} \end{bmatrix}$. By using the augmented system (5), the cost function (4) becomes

$$J(\boldsymbol{\chi},\boldsymbol{\alpha},\boldsymbol{\gamma}) = \int_t^\infty (\boldsymbol{\chi}^\mathrm{T}\boldsymbol{Q}\boldsymbol{\chi} + \boldsymbol{\alpha}^\mathrm{T}\boldsymbol{R}\boldsymbol{\alpha} - \boldsymbol{\gamma}^\mathrm{T}\boldsymbol{S}\boldsymbol{\gamma})\,\mathrm{d}\tau, \tag{6}$$

where $\boldsymbol{Q} = \mathrm{diag}\{\bar{\boldsymbol{Q}}, \boldsymbol{0}\}$.

Then the two-player ZS differential game is

$$\boldsymbol{V}^*(\boldsymbol{\chi}) = \min_{\boldsymbol{\alpha}} \max_{\boldsymbol{\gamma}} J(\boldsymbol{\chi}, \boldsymbol{\alpha}, \boldsymbol{\gamma}), \tag{7}$$

where $\boldsymbol{V}^*$ is the value of the game. In this paper, we assume that the two-player optimal control problem has a unique solution, i.e., the Nash condition holds

$$\boldsymbol{V}^*(\boldsymbol{\chi}) = \min_{\boldsymbol{\alpha}} \max_{\boldsymbol{\gamma}} J(\boldsymbol{\chi}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = \max_{\boldsymbol{\gamma}} \min_{\boldsymbol{\alpha}} J(\boldsymbol{\chi}, \boldsymbol{\alpha}, \boldsymbol{\gamma}). \tag{8}$$

By Leibniz's formula and differentiating, the nonlinear ZS game Bellman equation, which is given in terms of the Hamiltonian function, is obtained as

$$\begin{aligned} \boldsymbol{\nu}(\boldsymbol{\chi}, \boldsymbol{\nabla V}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = {} & \boldsymbol{\chi}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\chi} + \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\alpha} - \boldsymbol{\gamma}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{\gamma} \\ & + \boldsymbol{\nabla V}^{\mathrm{T}} (\boldsymbol{\xi} + \boldsymbol{\mu} \boldsymbol{\alpha} + \boldsymbol{\nu} \boldsymbol{\gamma}) = 0, \end{aligned} \tag{9}$$

where $\boldsymbol{\nabla V} = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{\chi}}$. The stationary conditions are

$$\frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\alpha}} = \boldsymbol{0}, \tag{10}$$

$$\frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\gamma}} = \boldsymbol{0}. \tag{11}$$

According to Eq. (9), we have the optimal control and the disturbance

$$\boldsymbol{\alpha} = -\frac{1}{2} \boldsymbol{R}^{-1} \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\nabla V}, \tag{12}$$

$$\boldsymbol{\gamma} = \frac{1}{2} \boldsymbol{S}^{-1} \boldsymbol{\nu}^{\mathrm{T}} \boldsymbol{\nabla V}. \tag{13}$$

Substitute Eqs. (12) and (13) into Bellman equation (9), we can derive $\boldsymbol{V}$ from the solution of the HJI equation

$$\begin{aligned} 0 = {} & \boldsymbol{\chi}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\chi} + \boldsymbol{\nabla V}^{\mathrm{T}} \boldsymbol{\xi} - \frac{1}{4} \boldsymbol{\nabla V}^{\mathrm{T}} \boldsymbol{\mu} \boldsymbol{R}^{-1} \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\nabla V} \\ & + \frac{1}{4} \boldsymbol{\nabla V}^{\mathrm{T}} \boldsymbol{\nu} \boldsymbol{S}^{-1} \boldsymbol{\nu}^{\mathrm{T}} \boldsymbol{\nabla V}. \end{aligned} \tag{14}$$

The HJI equation provides the solution to the optimal control problem for the ZS game. When it can be solved, it provides an optimal control in state-variable feedback (i.e., closed-loop) form. The Bellman equation is a partial differential equation for the value. That is, given any stabilizing feedback control policies $\boldsymbol{u}(x)$ and $\boldsymbol{d}(x)$ yielding finite values, the solution to the Bellman equation (9) is the value given by Eq. (6).

As the system dynamics is unknown, a data-based synchronous method will be established to solve the HJI equation (14).

3. Data-based synchronous method

In this section, the PI algorithm is first given. Then the off-policy learning is used to transform the PI algorithm to a synchronous method without system dynamics.

The PI algorithm starts from the initial admissible control pair $(\boldsymbol{\alpha}^{[0]}, \boldsymbol{\gamma}^{[0]})$. Then for iterative step $k \geq 0$, $\boldsymbol{V}^{[k]}$ is obtained by

$$\begin{aligned} 0 = {} & \boldsymbol{\chi}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\chi} + \boldsymbol{\alpha}^{[k]\mathrm{T}} \boldsymbol{R} \boldsymbol{\alpha}^{[k]} - \boldsymbol{\gamma}^{[k]\mathrm{T}} \boldsymbol{S} \boldsymbol{\gamma}^{[k]} \\ & + \boldsymbol{\nabla V}^{[k]\mathrm{T}} (\boldsymbol{\xi} + \boldsymbol{\mu} \boldsymbol{\alpha}^{[k]} + \boldsymbol{\nu} \boldsymbol{\gamma}^{[k]}), \end{aligned} \tag{15}$$

and the policy pair updates by

$$\boldsymbol{\alpha}^{[k+1]} = -\frac{1}{2} \boldsymbol{R}^{-1} \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\nabla V}^{[k]}, \tag{16}$$

$$\boldsymbol{\gamma}^{[k+1]} = \frac{1}{2} \boldsymbol{S}^{-1} \boldsymbol{\nu}^{\mathrm{T}} \boldsymbol{\nabla V}^{[k]}. \tag{17}$$

From Eqs. (15)–(17), we can see that the system dynamics is necessary for the PI algorithm. Therefore, the following synchronous method is given based on the PI algorithm and off-policy learning.

Let $\boldsymbol{\alpha}^{[k]}$ and $\boldsymbol{\gamma}^{[k]}$ be obtained by Eqs. (16) and (17), then the original system (5) is rewritten as

$$\dot{\boldsymbol{\chi}} = \boldsymbol{\xi} + \boldsymbol{\mu} \boldsymbol{\alpha}^{[k]} + \boldsymbol{\nu} \boldsymbol{\gamma}^{[k]} + \boldsymbol{\mu} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{[k]}) + \boldsymbol{\nu} (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{[k]}). \tag{18}$$

According to Eq. (18), we have

$$\begin{aligned} & \boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t+T)) - \boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t)) \\ = {} & \int_t^{t+T} \boldsymbol{\nabla V}^{[k]\mathrm{T}} \dot{\boldsymbol{\chi}} \, \mathrm{d}\tau \\ = {} & \int_t^{t+T} \boldsymbol{\nabla V}^{[k]\mathrm{T}} (\boldsymbol{\xi} + \boldsymbol{\mu} \boldsymbol{\alpha}^{[k]} + \boldsymbol{\nu} \boldsymbol{\gamma}^{[k]}) \, \mathrm{d}\tau \\ & + \int_t^{t+T} \boldsymbol{\nabla V}^{[k]\mathrm{T}} (\boldsymbol{\mu} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{[k]}) + \boldsymbol{\nu} (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{[k]})) \, \mathrm{d}\tau. \end{aligned} \tag{19}$$

From Eqs. (16) and (17), we have

$$\boldsymbol{\nabla V}^{[k]\mathrm{T}} \boldsymbol{\mu} = -2 \boldsymbol{\alpha}^{[k+1]\mathrm{T}} \boldsymbol{R}, \tag{20}$$

$$\boldsymbol{\nabla V}^{[k]\mathrm{T}} \boldsymbol{\nu} = 2 \boldsymbol{\gamma}^{[k+1]\mathrm{T}} \boldsymbol{S}. \tag{21}$$

Then equation (19) is

$$\begin{aligned} & \boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t+T)) - \boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t)) \\ = {} & \int_t^{t+T} \boldsymbol{\nabla V}^{[k]\mathrm{T}} (\boldsymbol{\xi} + \boldsymbol{\mu} \boldsymbol{\alpha}^{[k]} + \boldsymbol{\nu} \boldsymbol{\gamma}^{[k]}) \, \mathrm{d}\tau \\ & + \int_t^{t+T} (-2 \boldsymbol{\alpha}^{[k+1]\mathrm{T}} \boldsymbol{R} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{[k]}) \\ & + 2 \boldsymbol{\gamma}^{[k+1]\mathrm{T}} \boldsymbol{S} (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{[k]})) \, \mathrm{d}\tau. \end{aligned} \tag{22}$$

Thus from Eq. (15), the off-policy Bellman equation for the ZS games is expressed as

$$\boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t+T)) - \boldsymbol{V}^{[k]}(\boldsymbol{\chi}(t))$$

$$= -\int_t^{t+T} \chi^{\mathrm{T}}\boldsymbol{Q}\chi + \boldsymbol{\alpha}^{[k]\mathrm{T}}\boldsymbol{R}\boldsymbol{\alpha}^{[k]} - \boldsymbol{\gamma}^{[k]\mathrm{T}}\boldsymbol{S}\boldsymbol{\gamma}^{[k]}\,\mathrm{d}\tau + \int_t^{t+T} (-2\boldsymbol{\alpha}^{[k+1]\mathrm{T}}\boldsymbol{R}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{[k]}) + 2\boldsymbol{\gamma}^{[k+1]\mathrm{T}}\boldsymbol{S}(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{[k]}))\,\mathrm{d}\tau. \quad (23)$$

In Eq. (23), the system dynamics is avoided. In the following section, the NN implementation procedure is presented with convergence analysis.

4. NN implementation

The NN implementation procedure is first given in this section. Then the convergence of the NN weight is analyzed.

The neural network expression of CNN is given as

$$\boldsymbol{V}^{[k]}(\chi) = \boldsymbol{W}_{\boldsymbol{V}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{V}}(\chi) + \boldsymbol{\delta}_{\boldsymbol{V}}(\chi), \quad (24)$$

where $\boldsymbol{W}_{\boldsymbol{V}}^{[k]}$ is the ideal weight of the critic network, $\boldsymbol{\phi}_{\boldsymbol{V}}(\chi)$ is the active function, and $\boldsymbol{\delta}_{\boldsymbol{V}}(\chi)$ is the residual error. Let the estimation of $\boldsymbol{W}_{\boldsymbol{V}}^{[k]}$ be $\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]}$. Then the estimation of $\boldsymbol{V}^{[k]}(\chi)$ is

$$\hat{\boldsymbol{V}}^{[k]}(\chi) = \hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{V}}(\chi), \quad (25)$$

$$\boldsymbol{\nabla}\hat{\boldsymbol{V}}^{[k]}(\chi) = \boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}}^{\mathrm{T}}(\chi)\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]}. \quad (26)$$

The neural network expression of ANN is

$$\boldsymbol{\alpha}^{[k]} = \boldsymbol{W}_{\boldsymbol{\alpha}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\chi) + \boldsymbol{\delta}_{\boldsymbol{\alpha}}(\chi), \quad (27)$$

where $\boldsymbol{W}_{\boldsymbol{\alpha}}^{[k]}$ is the ideal weight of the action network, $\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\chi)$ is the active function, and $\boldsymbol{\delta}_{\boldsymbol{\alpha}}(\chi)$ is the residual error. Let $\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k]}$ be the estimation of $\boldsymbol{W}_{\boldsymbol{\alpha}}^{[k]}$, then the estimation of $\boldsymbol{\alpha}^{[k]}$ is

$$\hat{\boldsymbol{\alpha}}^{[k]} = \hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\chi). \quad (28)$$

The neural network expression of DNN is

$$\boldsymbol{\gamma}^{[k]} = \boldsymbol{W}_{\boldsymbol{\gamma}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\gamma}}(\chi) + \boldsymbol{\delta}_{\boldsymbol{\gamma}}(\chi), \quad (29)$$

where $\boldsymbol{W}_{\boldsymbol{\gamma}}^{[k]}$ is the ideal weight of the action network, $\boldsymbol{\phi}_{\boldsymbol{\gamma}}(\chi)$ is the active function, and $\boldsymbol{\delta}_{\boldsymbol{\gamma}}(\chi)$ is the residual error. Let $\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k]}$ be the estimation of $\boldsymbol{W}_{\boldsymbol{\gamma}}^{[k]}$, then the estimation of $\boldsymbol{\gamma}^{[k]}$ is

$$\hat{\boldsymbol{\gamma}}^{[k]} = \hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\gamma}}(\chi). \quad (30)$$

Substituting Eqs. (25), (28), and (30) into Eq. (23), we can define the equation error as

$$\varepsilon^{[k]} = \hat{\boldsymbol{V}}^{[k]}(\chi(t)) - \hat{\boldsymbol{V}}^{[k]}(\chi(t+T)) - \int_t^{t+T} \chi^{\mathrm{T}}\boldsymbol{Q}\chi + \hat{\boldsymbol{\alpha}}^{[k]\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\alpha}}^{[k]} - \hat{\boldsymbol{\gamma}}^{[k]\mathrm{T}}\boldsymbol{S}\hat{\boldsymbol{\gamma}}^{[k]}\,\mathrm{d}\tau + \int_t^{t+T} (-2\hat{\boldsymbol{\alpha}}^{[k+1]T}\boldsymbol{R}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}^{[k]}) + 2\hat{\boldsymbol{\gamma}}^{[k+1]\mathrm{T}}\boldsymbol{S}(\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}}^{[k]}))\,\mathrm{d}\tau. \quad (31)$$

Substituting Eqs. (25), (28), and (30) into Eq. (31), we have

$$\varepsilon^{[k]} = -\int_t^{t+T} \chi^{\mathrm{T}}\boldsymbol{Q}\chi + \hat{\boldsymbol{\alpha}}^{[k]\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\alpha}}^{[k]} - \hat{\boldsymbol{\gamma}}^{[k]\mathrm{T}}\boldsymbol{S}\hat{\boldsymbol{\gamma}}^{[k]}\,\mathrm{d}\tau + \int_t^{t+T} (-2\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]T}\boldsymbol{\phi}_{\boldsymbol{\alpha}})^{\mathrm{T}}\boldsymbol{R}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}^{[k]}) + 2(\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\gamma}})^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{\gamma}-\hat{\hat{\boldsymbol{\gamma}}}^{[k]})\,\mathrm{d}\tau - \hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]\mathrm{T}}\boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}}, \quad (32)$$

where $\boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}} = \boldsymbol{\phi}_{\boldsymbol{V}}(\chi(t+T)) - \boldsymbol{\phi}_{\boldsymbol{V}}(\chi(t))$. By kronecker product, it has $\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]\mathrm{T}}\boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}} = \left(-\boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}}^{\mathrm{T}} \otimes \boldsymbol{I}\right)\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]}$,

$$\boldsymbol{\phi}_{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]}\boldsymbol{R}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}^{[k]}) = \left(\left((\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}^{[k]})^{\mathrm{T}}\boldsymbol{R}\right)\otimes\boldsymbol{\phi}_{\boldsymbol{\alpha}}^{\mathrm{T}}\right)\mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]}), \quad (33)$$

$$\boldsymbol{\phi}_{\boldsymbol{\gamma}}^{\mathrm{T}}\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]}\boldsymbol{S}(\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}}^{[k]}) = \left(\left((\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}}^{[k]})^{\mathrm{T}}\boldsymbol{S}\right)\otimes\boldsymbol{\phi}_{\boldsymbol{\gamma}}^{\mathrm{T}}\right)\mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]}). \quad (34)$$

Then we can define

$$\boldsymbol{\Psi}^{[k]} = -\boldsymbol{\nabla}\boldsymbol{\phi}_{\boldsymbol{V}}{}^{\mathrm{T}} \otimes \boldsymbol{I}, \quad (35)$$

$$\boldsymbol{\Xi}^{[k]} = -\int_t^{t+T} \chi^{\mathrm{T}}\boldsymbol{Q}\chi + \hat{\boldsymbol{\alpha}}^{[k]\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\alpha}}^{[k]} - \hat{\boldsymbol{\gamma}}^{[k]\mathrm{T}}\boldsymbol{S}\hat{\boldsymbol{\gamma}}^{[k]}\,\mathrm{d}\tau, \quad (36)$$

$$\boldsymbol{\Sigma}^{[k]} = -2\int_t^{t+T} \left(\left((\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}^{[k]})^{\mathrm{T}}\boldsymbol{R}\right)\otimes\boldsymbol{\phi}_{\boldsymbol{\alpha}}^{\mathrm{T}}\right)\mathrm{d}\tau, \quad (37)$$

$$\boldsymbol{\Omega}^{[k]} = 2\int_t^{t+T} \left(\left((\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}}^{[k]})^{\mathrm{T}}\boldsymbol{S}\right)\otimes\boldsymbol{\phi}_{\boldsymbol{\gamma}}^{\mathrm{T}}\right)\mathrm{d}\tau. \quad (38)$$

Therefore, equation (32) becomes

$$\varepsilon^{[k]} = \boldsymbol{\Psi}^{[k]}\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]} + \boldsymbol{\Sigma}^{[k]}\mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]}) + \boldsymbol{\Omega}^{[k]}\mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]}) + \boldsymbol{\Xi}^{[k]} = \left[\boldsymbol{\Psi}^{[k]}\boldsymbol{\Sigma}^{[k]}\boldsymbol{\Omega}^{[k]}\right]\begin{bmatrix}\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]}\\ \mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]})\\ \mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]})\end{bmatrix} + \boldsymbol{\Xi}^{[k]} = \boldsymbol{\Phi}^{[k]}\hat{\boldsymbol{W}}^{[k]} + \boldsymbol{\Xi}^{[k]}, \quad (39)$$

where $\hat{\boldsymbol{W}}^{[k]} = \begin{bmatrix}\hat{\boldsymbol{W}}_{\boldsymbol{V}}^{[k]}\\ \mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\alpha}}^{[k+1]})\\ \mathrm{vec}(\hat{\boldsymbol{W}}_{\boldsymbol{\gamma}}^{[k+1]})\end{bmatrix}$.

Define

$$\boldsymbol{E}^{[k]} = \frac{1}{2}\varepsilon^{[k]\mathrm{T}}\varepsilon^{[k]}. \quad (40)$$

According to the gradient descent method, we have the update law of $\hat{\boldsymbol{W}}^{[k]}$ as

$$\dot{\hat{\boldsymbol{W}}}^{[k]} = -\iota\boldsymbol{\Phi}^{[k]\mathrm{T}}\left(\boldsymbol{\Phi}^{[k]}\hat{\boldsymbol{W}}^{[k]} + \boldsymbol{\Xi}^{[k]}\right), \quad (41)$$

where $\iota > 0$ is a positive number.

Based on the NN implementation and off-policy learning, the convergence of the synchronous method is proposed.

Theorem 1 Let the update method for the augmented matrix be as in Eq. (41). Define the weight estimation error as $\tilde{\boldsymbol{W}}^{[k]} = \boldsymbol{W}^{[k]} - \hat{\boldsymbol{W}}^{[k]}$, then $\tilde{\boldsymbol{W}}^{[k]}$ is UUB.

Proof Let the Lyapunov function candidate be

$$\boldsymbol{L}_1^{[k]} = \frac{b_1}{2\iota}\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\tilde{\boldsymbol{W}}^{[k]}, b_1 > 0. \tag{42}$$

According to Eq. (41), we have

$$\begin{aligned}\dot{\tilde{\boldsymbol{W}}}^{[k]} &= \iota\boldsymbol{\Phi}^{[k]\mathrm{T}}\left(\boldsymbol{\Phi}^{[k]}(\boldsymbol{W}^{[k]} - \tilde{\boldsymbol{W}}^{[k]}) + \boldsymbol{\Xi}^{[k]}\right)\\ &= -\iota\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\tilde{\boldsymbol{W}}^{[k]} + \iota\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\boldsymbol{W}^{[k]}\\ &\quad + \iota\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Xi}^{[k]}.\end{aligned} \tag{43}$$

Therefore, the gradient of Eq. (42) is

$$\begin{aligned}\dot{\boldsymbol{L}}^{[k]} &= \frac{b_1}{\iota}\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\dot{\tilde{\boldsymbol{W}}}^{[k]}\\ &= b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\big(-\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\tilde{\boldsymbol{W}}^{[k]}\\ &\quad + \boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\boldsymbol{W}^{[k]} + \boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Xi}^{[k]}\big)\\ &= -b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\tilde{\boldsymbol{W}}^{[k]} + b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\boldsymbol{W}^{[k]}\\ &\quad + b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Xi}^{[k]}\\ &\le -b_1||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]}\boldsymbol{W}^{[k]}\\ &\quad + b_1\tilde{\boldsymbol{W}}^{[k]\mathrm{T}}\boldsymbol{\Phi}^{[k]\mathrm{T}}\boldsymbol{\Xi}^{[k]}\\ &\le -b_1||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + \frac{1}{2}||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2\\ &\quad + \frac{b_1^2}{2}||\boldsymbol{W}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2\\ &\quad + \frac{1}{2}||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + \frac{b_1^2}{2}||\boldsymbol{\Xi}^{[k]}||^2.\end{aligned} \tag{44}$$

By merger of similar items, equation (44) becomes

$$\begin{aligned}\dot{\boldsymbol{L}}_1^{[k]} &\le (-b_1+1)||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2\\ &\quad + \frac{b_1^2}{2}||\boldsymbol{W}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + \frac{b_1^2}{2}||\boldsymbol{\Xi}^{[k]}||^2.\end{aligned} \tag{45}$$

Define

$$\boldsymbol{A}^{[k]} = \frac{b_1^2}{2}||\boldsymbol{W}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + \frac{b_1^2}{2}||\boldsymbol{\Xi}^{[k]}||^2. \tag{46}$$

Then equation (45) is

$$\dot{\boldsymbol{L}}_1^{[k]} \le (-b_1+1)||\tilde{\boldsymbol{W}}^{[k]}||^2||\boldsymbol{\Phi}^{[k]}||^2 + \boldsymbol{A}^{[k]}. \tag{47}$$

Thus, if

$$b_1 > 1, \tag{48}$$

$$||\tilde{\boldsymbol{W}}^{[k]}||^2 > \frac{\boldsymbol{A}^{[k]}}{(b_1-1)||\boldsymbol{\Phi}^{[k]}||^2}, \tag{49}$$

then $\tilde{\boldsymbol{W}}^{[k]}$ is UUB. The proof completes.

According to the PI algorithm, $\hat{\boldsymbol{W}}^{[k]}$ is convergent, as $k \to \infty$. In the next theorem, the property of the chaotic tracking error system is given.

Theorem 2 Let $\hat{\boldsymbol{W}}^{[k]} \to \hat{\boldsymbol{W}}$, $\hat{\boldsymbol{\alpha}}^{[k]} \to \hat{\boldsymbol{\alpha}}$, and $\hat{\gamma}^{[k]} \to \hat{\gamma}$, as $k \to \infty$. The inputs of the chaotic tracking error system are $\hat{\boldsymbol{\alpha}}$ and $\hat{\gamma}$. Then the tracking error $\boldsymbol{e}$, control input $\hat{\boldsymbol{\alpha}}$, and disturbance input $\hat{\gamma}$ are UUB.

Proof Define the Lyapunov function candidate as

$$\boldsymbol{L} = \boldsymbol{L}_2 + \boldsymbol{L}_3, \tag{50}$$

where

$$\boldsymbol{L}_2 = b_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}, \quad b_2 > 0, \tag{51}$$

$$\boldsymbol{L}_3 = \int_t^{\infty}(\boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{Q}}\boldsymbol{e} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\alpha}} + \hat{\gamma}^{\mathrm{T}}\boldsymbol{S}\hat{\gamma})\,\mathrm{d}\tau. \tag{52}$$

Then we have

$$\begin{aligned}\dot{\boldsymbol{L}}_2 &= 2b_2\boldsymbol{e}^{\mathrm{T}}\dot{\boldsymbol{e}} = 2b_2\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{F} + \boldsymbol{G}\hat{\boldsymbol{\alpha}} + \boldsymbol{H}\hat{\gamma})\\ &\le 2b_2l||\boldsymbol{e}||^2 + b_2||\boldsymbol{e}||^2 + b_2||\boldsymbol{G}||^2||\hat{\boldsymbol{\alpha}}||^2\\ &\quad + b_2||\boldsymbol{e}||^2 + b_2||\boldsymbol{H}||^2||\hat{\gamma}||^2,\end{aligned} \tag{53}$$

$$\begin{aligned}\dot{\boldsymbol{L}}_3 &= -\boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{Q}}\boldsymbol{e} - \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\alpha}} - \hat{\gamma}^{\mathrm{T}}\boldsymbol{S}\hat{\gamma}\\ &\le -\lambda_{\min}(\bar{\boldsymbol{Q}})||\boldsymbol{e}||^2 - \lambda_{\min}(\boldsymbol{R})||\hat{\boldsymbol{\alpha}}||^2\\ &\quad - \lambda_{\min}(\boldsymbol{S})||\hat{\gamma}||^2,\end{aligned} \tag{54}$$

where $l > 0$ is the lipschitz constant.

From Eqs. (53) and (54), we have

$$\begin{aligned}\dot{\boldsymbol{L}} &\le \left(2b_2l + 2b_2 - \lambda_{\min}(\bar{\boldsymbol{Q}})\right)||\boldsymbol{e}||^2\\ &\quad + \left(b_2||\boldsymbol{G}||^2 - \lambda_{\min}(\boldsymbol{R})\right)||\hat{\boldsymbol{\alpha}}||^2\\ &\quad + \left(b_2||\boldsymbol{H}||^2 - \lambda_{\min}(\boldsymbol{S})\right)||\hat{\gamma}||^2.\end{aligned} \tag{55}$$

Therefore, if

$$b_2 \le \min\left\{\frac{\lambda_{\min}(\bar{\boldsymbol{Q}})}{2l+2}, \frac{\lambda_{\min}(\boldsymbol{R})}{||\boldsymbol{G}||^2}, \frac{\lambda_{\min}(\boldsymbol{S})}{||\boldsymbol{H}||^2}\right\}, \tag{56}$$

then $\boldsymbol{e}$, $\hat{\boldsymbol{\alpha}}$, and $\hat{\gamma}$ are UUB. The proof completes.

5. Simulation study

In this paper, two simulation examples are given to demonstrate the effectiveness of the proposed method.

5.1. Example 1

Consider the chaotic systems described by the following differential equation:[41]

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\beta)\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{d} + \boldsymbol{u}, \tag{57}$$

where $\boldsymbol{x} \in \mathbf{R}^3$, $\boldsymbol{u}$ is the control input, and $\boldsymbol{d}$ is the disturbance input. Let

$$\boldsymbol{A}(\beta) = \begin{bmatrix} -25\beta - 10 & 25\beta + 10 & 0\\ 28 - 35\beta & 29\beta - 1 & 0\\ 0 & 0 & -\dfrac{8+\beta}{3}\end{bmatrix}, \tag{58}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ -x_1 x_3 \\ x_1 x_2 \end{bmatrix}. \tag{59}$$

Here, let $\beta = 0$, system (57) becomes the Lorenz system when perturbations are not present. The trajectory of system (57) is shown in Fig. 1.

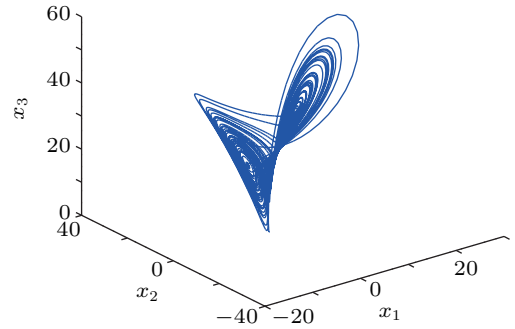


Fig. 1. (color online) Lorenz system where $\beta = 0$.

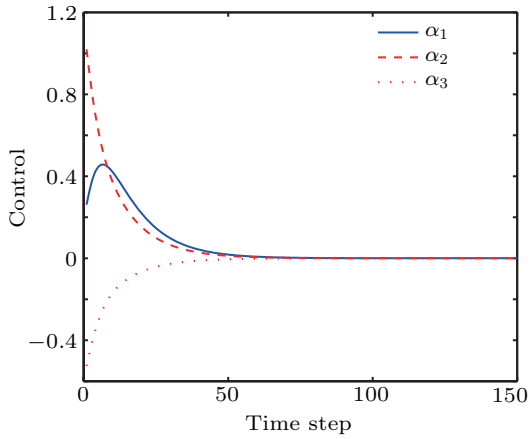


Fig. 2. (color online) Control input trajectories.

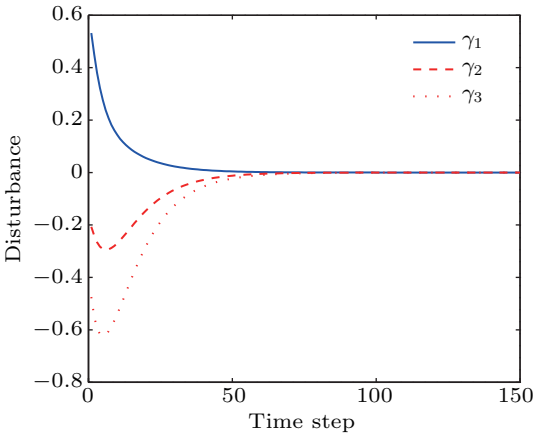


Fig. 3. (color online) Disturbance input trajectories.

In this paper, the desired trajectory is $\boldsymbol{x}_d = [0.1;\ 0.2;\ -0.3]$. We select hyperbolic tangent functions as the activation functions of critic, action, and disturbance networks. The structures of critic, action, and disturbance networks are $3-8-1$, $3-8-3$, and $3-8-3$, respectively. The initial weight W is selected arbitrarily from $(-1,1)$. For the cost function, $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ in the utility function are identity matrices of appropriate dimensions. After 150 time steps, the simulation results are obtained. Figures 2 and 3 are the control and disturbance input trajectories. Based on the inputs, the tracking error is given in Fig. 4, which shows that the tracking error is convergent. The chaotic system state is demonstrated in Fig. 5. Is is clear that the chaotic system tracks the given trajectories.

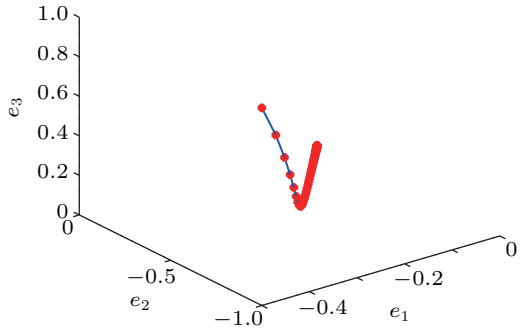


Fig. 4. (color online) Tracking error.

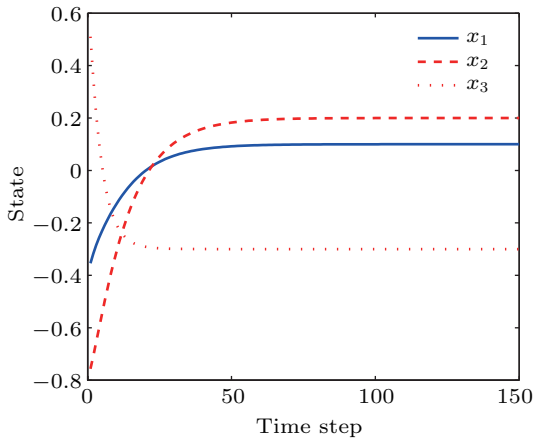


Fig. 5. (color online) System state.

5.2. Example 2

Consider the following chaotic system:[42,43]

$$\dot{\boldsymbol{x}} = \boldsymbol{f} + \boldsymbol{g}u + \boldsymbol{h}w, \tag{60}$$

where $\boldsymbol{f} = \begin{bmatrix} a(x_2 - x_1) \\ -x_1 x_3 + c x_2 \\ x_1 x_2 - b x_3 \end{bmatrix}$, $\boldsymbol{g} = \mathrm{diag}(5,5,5)$, and $\boldsymbol{h} = \mathrm{diag}(1,1,1)$. When $a = 36$, $b = 3$, and $c = 20$, the internal system (46) without perturbations is shown in Fig. 6.

In this example, the desired trajectory is $\boldsymbol{x}_d = [-1;\ 0.5;\ 1.5]$. The matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ in the utility function are identity matrices of appropriate dimensions. For critic, action, and disturbance networks, the activation functions are hyperbolic tangent functions. The structures are $3-8-1$, $3-8-3$, and $3-8-3$, respectively. The initial weight W

is selected arbitrarily from $(-1,1)$. After 100 time steps, the simulation results are shown in Figs. 7–10. In Figs. 7 and 8, the control and disturbance inputs are given, which are convergent. Under the inputs action, the tracking error trajectories are displayed in Fig. 9, which converge to zero. At last, the closed-loop chaotic system state based on the inputs is demonstrated in Fig. 10. It can be seen that the proposed method is effective to make the chaotic system track the given trajectories.

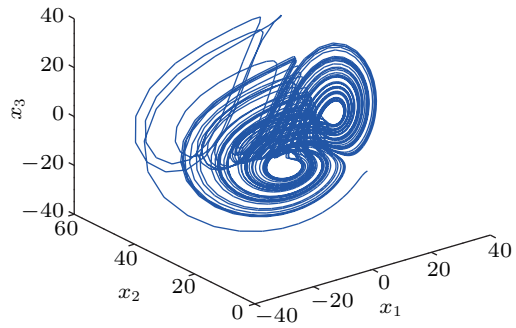


Fig. 6. (color online) Lü chaotic attractor.

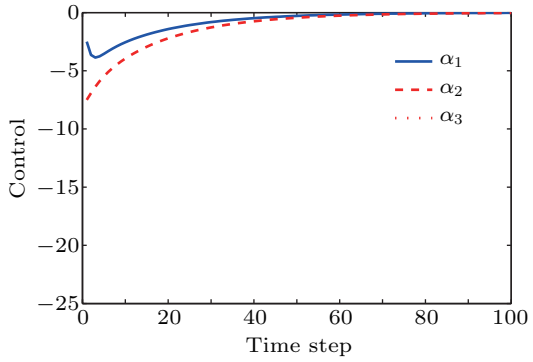


Fig. 7. (color online) Control input trajectories.

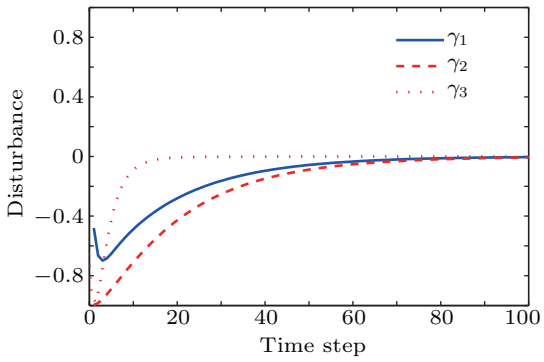


Fig. 8. (color online) Disturbance input trajectories.

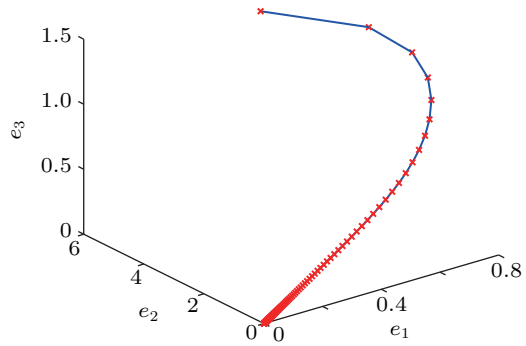


Fig. 9. (color online) Tracking error.

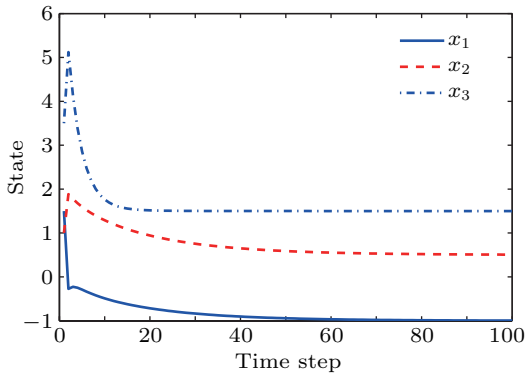


Fig. 10. (color online) State.

6. Conclusion

An optimal tracking control method for a chaotic system with unknown dynamics and disturbances is proposed in this paper. The tracking error dynamics and the generator dynamics make up the augmented system. A new cost function is introduced for the optimal tracking control problem. The PI is introduced to solve the min-max optimization problem. The off-policy learning method is applied to update the iterative cost function using only measured data and without any knowledge about the system dynamics. ANN, DNN, and CNN are used to approximate the cost function, control, and disturbance with convergence analyses. It is proven that the closed-loop tracking error system is convergent. Simulation results are given to show the effectiveness of the proposed synchronous solution method for chaotic system tracking problem. The future research is to use the proposed approach to solve the H_∞ control problem for a class of systems with interconnection term, and to analyze the convergence of PI in ZS games.

References

[1] Shen Z and Li J 2016 *Mathematics and Computers in Simulation*
[2] Wang G, Cai B, Jin P and Hu T 2016 *Chin. Phys. B* **25** 010503
[3] Wang W, Zhang X, Chang Y, Wang X, Wang Z, Chen X and Zheng L 2016 *Chin. Phys. B* **25** 010202
[4] Zhang Z Y, Feng X Q, Yao Z H and Jia H Y 2015 *Chin. Phys. B* **24** 110503
[5] Wu J, Wang L, Chen G and Duan S 2016 *Chaos Soliton. Fract.* **92** 20
[6] Lü J H and Chen G R 2002 *International Journal of Bifurcation and Chaos* **12** 659
[7] Ma T, Zhang H and Fu J 2008 *Chin. Phys. B* **17** 4407
[8] Ma T and Fu J 2011 *Chin. Phys. B* **20** 050511
[9] Yang D 2014 *Chin. Phys. B* **23** 010504
[10] Song R, Xiao W and Wei Q 2014 *Chin. Phys. B* **23** 050504

[11] Wei Q, Song R, Xiao W and Sun Q 2015 *Chin. Phys. B* **24** 090504
[12] Wei Q, Liu D and Lin H 2016 *IEEE Transactions on Cybernetics* **46** 840
[13] Zhang H, Feng T, Yang G H and Liang H 2015 *IEEE Transactions on Cybernetics* **45** 1315
[14] Wei Q, Wang F, Liu D and Yang X 2014 *IEEE Transactions on Cybernetics* **44** 2820
[15] Modares H and Lewis F L 2014 *IEEE Transactions on Automatic Control* **59** 3051
[16] Liu D and Wei Q 2013 *IEEE Transactions on Cybernetics* **43** 779
[17] Nguyen T L 2016 *Neurocomputing*
[18] Song R, Xiao W, Zhang H and Sun C 2014 *IEEE Transactions on Neural Networks and Learning Systems* **25** 1733
[19] Song R, Lewis F L, Wei Q, Zhang H and Jiang Z P 2015 *IEEE Transactions on Neural Networks and Learning Systems* **26** 851
[20] Song R, Lewis F L, Wei Q and Zhang H 2016 *IEEE Transactions on Cybernetics* **46** 1041
[21] Wang W, Li L, Peng H, Xiao J and Yang Y 2014 *Nonlinear Dynamics* **76** 591
[22] Wang W, Li L, Peng H, Xiao J and Yang Y 2014 *Neural Networks* **53** 8
[23] Li L, Li W, Kurths J, Luo Q, Yang Y and Li S 2015 *Chaos Soliton. Fract.* **72** 20
[24] Song R, Xiao W, Sun C and Wei Q 2013 *Chin. Phys. B* **22** 090502
[25] Basar T and Bernard P 1995 *H_∞ Optimal Control and Related Minimax Design Problems* (Boston: Birkhauser)
[26] Basar T and Olsder G J 1999 *Dynamic Noncooperative Game Theory* (2nd Edn.) Vol. 23
[27] Devasia S, Chen D and Paden B 1996 *IEEE Transactions of Automatic Control* **14** 930
[28] Zhang H, Wei Q and Liu D 2011 *Automatica* **47** 207
[29] Wei Q, Song R and Yan P 2016 *IEEE Transactions on Neural Networks and Learning Systems* **27** 444
[30] Liu D and Wei Q 2014 *International Journal of Adaptive Control and Signal Processing* **28** 205
[31] Song R, Lewis F L and Wei Q 2016 *IEEE Transactions on Neural Networks and Learning Systems*
[32] Zhang F, Liao X and Zhang G 2016 *Applied Mathematics and Computation* **284** 332
[33] Chen G and Ueta T 1999 *International Journal of Bifurcation and Chaos* **9** 1465
[34] Leonov G A and Kuznetsov N V 2015 *Applied Mathematics and Computation* **256** 334
[35] Lorenz E 1963 *Journal of the Atmospheric Sciences* **20** 130
[36] Leonov G A, Kuznetsov N V, Korzhemanova N A and Kusakin D V 2016 *Communications in Nonlinear Science and Numerical Simulation* **41** 84
[37] Chua L, Komuro M and Matsumoto T 1986 *IEEE Transactions on Circuits and Systems* **33** 1072
[38] Alkahtani B S T 2016 *Chaos Soliton. Fract.* **89** 547
[39] Wiggins S 1987 *Phys. Lett. A* **124** 138
[40] Simo H and Woafo P 2016 *Optik-International Journal for Light and Electron Optics* **127** 8760
[41] Vargas J A R, Grzeidak E and Gularte K H M 2016 *Neurocomputing* **174** 1038
[42] Lü J, Chen G and Zhang S 2002 *International Journal of Bifurcation and Chaos* **12** 1001
[43] Lü J, Chen G and Zhang S 2002 *Chaos Soliton. Fract.* **14** 669