An Application of Backtracking Search Algorithm for Chaos Control Using Adaptive Fuzzy PID Controller with H∞ tracking performance

Maryam Khodabandeh[1], Alireza Alfi[2]*

Received:2015/6/7 Accepted:2015/7/11

Abstract

This paper introduces a technique for controlling a class of uncertain chaotic systems using an adaptive fuzzy Proportional-Integrator-Derivative (PID) controller with H∞ tracking performance. The purpose of this work is to achieve optimal tracking performance of the controller using Backtracking Search Algorithm (BSA). BSA, which is a novel heuristic algorithm, has an easy structure with single control parameter. In BSA, three basic genetic operators (selection, mutation and crossover) are utilized to generate trial individuals. To this reason, the control problem in hand is considered as an optimization problem by defining an appropriate objective function. Stability analysis of the control scheme is provided based on Lyapunov theory and modified Riccati-like equation, where the robustness of the closed-loop system is guaranteed by H∞ tracking performance for any predefined level. To evaluate the performance of the proposed control method, it is employed for tracking control of Duffing uncertain chaotic system. Simulation results show the capability of the proposed controller.

Keywords: Chaos Control, Adaptive Fuzzy PID, Stability, Optimization, Backtracking Search Algorithm

1. Introduction

Chaotic systems have complex dynamical behaviours that possess distinctive aspects like excessive sensitivity to initial conditions [1]. Since chaotic phenomenon exists in many engineering and scientific fields, controlling chaos has attracted much interest during the last decades. Different control methodologies have been introduced in the literature [2-12]. Proportional-Integrator-Derivative (PID) controllers are one of the most well-known control methods yet. They endeavour to minimize the error between a desired input and the output of the process by providing an appropriate signal that can control the process accordingly.

PID controllers have been successfully applied for controlling of the processes because of interesting aspects such as simple realization, easy implementation and appropriate reliability [13]. In spite of the interest of PID controllers, there exist some reasons in chaos control which can deteriorate the performance of them: (1) the parameters of the system are not precisely known (or unknown) in which can be perturbed during the process, (2) the existence of the external disturbances which is unavoidable. This has led to an intense interest in the development of robust tracking control of chaotic systems.

Since Fuzzy Logic (FL) is universal approximator [14], FL control systems offer an operational method to handle nonlinear systems, particularly against incomplete knowledge of the plant. Although many FL-based control schemes have been developed for unknown nonlinear systems, the key disadvantage of the FL control systems is the lack of any systematic approaches where stability analysis of such a system is not easy. In addition, the tuning of the parameter is usually a time-consuming process, because of the nonlinear and multi-parametric nature of fuzzy systems [15]. From the universal approximation aspect of the fuzzy systems, the adaptive control methods that incorporate the FL techniques are utilized to control the nonlinear dynamic system [16]. Recently, these systems with H∞ tracking performance have been introduced for nonlinear systems [17-22]. These techniques are thoroughly related to their robust stability and performance providing satisfactory results to the trajectory tracking problem using a small amount of the fuzzy inference mechanisms.

Based on aforementioned, an adaptive fuzzy PID controller with H∞ tracking performance is introduced for controlling uncertain chaotic systems. By using Lyapunov stability theory, the proposed controller can assure the robust stability of the system with H∞ tracking performance for

1. Faculty of Electrical and Robotic Engineering, University of Shahrood.
2 . Faculty of Electrical and Robotic Engineering, University of Shahrood. a_alfi@shahroodut.ac.ir

any predefined level. While applying the control manner into the system, an error dynamic of closed-loop system can be achieved. The significant problem is to choose an arbitrary set of the poles of the error dynamic which are directly related to the controller performance. Trials–errors procedure is generally applied for tuning until a required behaviour of the system is achieved in which can be became more delicate and hard as the complexity of the controlled plant increases. To overcome this deficiency, the problem can be considered as an optimization problem by defining an appropriate objective function. Here, it is required to develop an optimal tuning methodology of the controller to determine a set of the poles simultaneously. Accordingly, different types of conventional techniques have been employed like Gradient Descents [23]. However, most of them have some fundamental drawbacks such as differentiability of the objective function. Evolutionary Algorithms (EAs) are also one of the most qualified methodologies for finding the optimal solution via cooperation and competition between the individuals of the population. These algorithms have been successfully applied in different areas such as clustering [24-27], Synchronization of bilateral teleoperation systems [28-30], Optimal setting of TCSCs in power systems [31], dynamic optimisation problems [32,33], system identification [34,35], design optimization [36], optimal controller design [37,38] and job shop scheduling problem [39]. In this paper, Backtracking Search optimization Algorithm (BSA) originally proposed by Civicioglu [40] is employed to calculate the proper location of the poles of the error dynamic. BSA has an easy structure with single control parameter which has three basic genetic operators (selection, mutation and crossover) to generate trial individuals. The preference of the BSA compared to other algorithms can be found in [40].

The reminder of paper is organized as follows. The problem formulation is given in section 2. Section 3 presents an adaptive fuzzy PID control with H∞ tracking performance in details. Section 4 demonstrates the corresponding optimal tracking performance problem. Simulation results are given in section 5 to show the efficiency of the proposed approach. Section 6 outlines the main conclusions.

2. Problem Formulation

In this paper, a dynamical equation of an n*th* order nonlinear system is given as

$$x^{(n)} = f(x, \dot{x}, \ldots, x^{(n-1)}) + g(x, \dot{x}, \ldots, x^{(n-1)})u + d \quad (1)$$
$$y = x$$

where $\boldsymbol{x} = [x, \dot{x}, \ldots, x^{(n-1)}]^T = [x_1, x_2, \ldots, x_n]^T \in \Re^n$ is a state vector, $y \in \Re$ is the system output, $u \in \Re$ is the control input, d is an external disturbance which is bounded, and $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are the unknown but bounded nonlinear functions. Here, all state variables $\boldsymbol{x}$ are measurable and the input gain is strictly positive, i.e. $0 < g_L \le g(\boldsymbol{x}) < \infty$.

In chaos control problem, the output of the system must track the reference signal $\boldsymbol{y}_{\mathbf{r}}$. Accordingly, the tracking error vector is defined as $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{y}_{\mathbf{r}} = [e, \dot{e}, \ldots, e^{(n-1)}]^T \in \Re^n$, where $\boldsymbol{y}_r = [y_r, \dot{y}_r, \ldots, y_r^{(n-1)}]^T \in \Re^n$ and $\boldsymbol{y} = [y, \dot{y}, \ldots, y^{(n-1)}]^T \in \Re^n$.

If the nonlinear functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are entirely known, then we have

$$e^{(n)} + \boldsymbol{k}^T \boldsymbol{e} = g(\boldsymbol{x})(u - u^*) + d \quad (2)$$

where

$$u^* = \frac{-1}{g(\boldsymbol{x})}(f(\boldsymbol{x}) + y_r^{(n)} - \boldsymbol{k}^T \boldsymbol{e}) \quad (3)$$

in which $\boldsymbol{k} = [k_n, \ldots, k_2, k_1]^T \in \Re^n$ is a Hurwitz polynomial. Consequently, the error dynamic can be described as

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\left[g(x)\left(u - u^*\right) + d\right] \quad (4)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -k_1 & -k_2 & \cdots & \cdots & -k_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

In this case, the system is free of external disturbance d and $u = u^*$. Afterward, we have $\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e}$ and $\lim_{t \to \infty} \boldsymbol{e} = \boldsymbol{0}$. However, $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are commonly unknown and there exists the external disturbance d so that u^* is unknown. To overcome this problem, an adaptive PID-type control law is given by

$$u = u_{PID} + u \qquad (5)$$

where

$$u_{PID} = k_P e(t) + k_I \int e(t)dt + k_D \dot{e}(t) \qquad (6)$$

performs as an approximator of u^* in which k_P, k_I, and k_D are the proportional, integral, and derivative parameters of the PID controller, respectively. Finally, u is an H∞ compensator is defined by

$$u = -\frac{1}{l}\boldsymbol{B}^T \boldsymbol{P} \boldsymbol{e} g^{-1}(\boldsymbol{x}) \qquad (7)$$

where l is a positive constant to be determined namely learning rate and $\boldsymbol{P}$ is a positive definite matrix which is a solution of the Riccati-like equation as follows [41].

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \frac{2}{l}\boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P} + \frac{1}{r^2}\boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P} = 0, \qquad 2r^2 \ge l \qquad (8)$$

From Eq. (6), the PID-type controller can be represented in the matrix form as

$$u_{PID}(z|q) = q^T z(e) \qquad (9)$$

where $q = [k_P, k_I, k_D]^T$ and $z = [e_1, e_2, e_3]^T = [e, e_I, e_D]^T$ in which $e_I = k_I \int e(t)dt$ and $e_D = k_D \dot{e}(t)$.

The PID-type controller given in Eq. (9) can uniformly approximate the controller u^* (Chang et al., 2002). Therefore, there is an optimal gain vector q^* such that the approximation error between u and u^* can be bounded by a predefined constant level d, i.e., $u = u^* = u_{PID}(z|q^*) + d(z)$ where $|d(z)| \le d, \forall t$.

Define the following minimum approximation error

$$w = g(\boldsymbol{x})[u_{PID}(z|q^*) - u^*] + d \qquad (10)$$

which can be rewritten as

$$d = w - g(\boldsymbol{x})[u_{PID}(z|q^*) - u^*] \qquad (11)$$

Substituting Eqs. (5) and (11) into (4) yields

$$\begin{aligned}\dot{\boldsymbol{e}} &= \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}(g(\boldsymbol{x})u_{PID}(z|q) + g(\boldsymbol{x})u - g(\boldsymbol{x})u^* + w - g(\boldsymbol{x})u_{PID}(z|q^*) + g(\boldsymbol{x})u^*) \\ &= \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}g(\boldsymbol{x})(u_{PID}(z|q) - u_{PID}(z|q^*) + u) + \boldsymbol{B}w\end{aligned} \qquad (12)$$

3. Adaptive fuzzy PID controller with H∞ tracking performance

Because of the existing unknown parameters of the controlled plant given in Eq. (1), the goal is to design a fuzzy PID controller $u_{PID}(z|q)$ using the update laws to adjust the adaptable parameter vector q such that the tracking performance is satisfied.

The elementary structure of the fuzzy system contains a set of fuzzy rules and inference engine. The fuzzy inference engine utilized the fuzzy rules to execute a mapping from an input linguistic vector z to an output linguistic variable $z(\boldsymbol{\xi}) \in \Re$. It is worth mentioning that there is no direction in fuzzy set theory to determine the best shapes of the fuzzy sets, thus different shapes need to be utilized to accomplish an optimum solution for various ranges of the system states. Here, the general shape of membership functions is considered for the inputs e_1, e_2, e_3 as follows.

$$m_N(e_k) = \begin{cases} 1 & \text{if } e_k \le -1 \\ 1 - 2(e_k + 1)^2 & \text{if } -1 \le e_k \le -0.5 \\ 2e_k & \text{if } -0.5 \le e_k \le 0 \\ 0 & \text{if } e_k \ge 0 \end{cases} \qquad (13)$$

$$m_Z(e_k) = e^{(-e^2/(2s)^2)}, s = 0.3$$

$$m_P(e_k) = \begin{cases} 0 & \text{if } e_k \le 0 \\ 2e_k & \text{if } 0 \le e_k \le 0.5 \\ 1 - 2(e_k - 1)^2 & \text{if } 0.5 \le e_k \le 1 \\ 1 & \text{if } e_k \ge 1 \end{cases}$$

where P, Z and N are the linguistic meanings of the membership functions and $k = 1,2,3$.

From [42], a fuzzy system is a multi-input single-output (MISO) system including $N = \prod_{k=1}^{3} N_k$ rules in the following form.

$R_{i_1 i_2 i_3}$: **IF** e_1 **is** $G^1_{i_1}$ **AND** e_2 **is** $G^2_{i_2}$ **AND** e_3 **is** $G^3_{i_3}$ **THEN** $z(x)$ **is** $C_{i_1 i_2 i_3}$,

$i_1 = 1,\ldots,N_1,\ i_2 = 1,\ldots,N_2,\ i_3 = 1,\ldots,N_3$

where e_k, $k = 1,2,3$, and $z(x)$ stand for the linguistic variables associated with the inputs and output of the fuzzy system, respectively, and $G_{i_k}^k$ and $C_{i_1 i_2 i_3}$ are linguistic values of linguistic variables z and $z(z)$ in the universes of discourse $U \in \Re^3$ and $\Re$, respectively.

The key components of a fuzzy logic controller are the fuzzifier, the inference engine and the defuzzifier. The fuzzifier transforms the numeric into fuzzy sets called fuzzification and the inference performs all of the logic manipulations in a fuzzy controller. Finally, the results of the inference process are transformed into a numeric value using the defuzzifier, so- called defuzzification. In designing the fuzzy controller, using singleton fuzzifier, product inference engine and the centre average defuzzifier, the following crisp output can be obtained.

$$z(z) = \sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\sum_{i_3=1}^{N_3} \frac{\prod_{k=1}^{3} m_k(e_k)}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\sum_{i_3=1}^{N_3}\prod_{k=1}^{3} m_k(e_k)} c_{i_1 i_2 i_3} = \sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\sum_{i_3=1}^{N_3} a_{i_1 i_2 i_3}(z) c_{i_1 i_2 i_3} \quad (14)$$

where $c_{i_1 i_2 i_3}$'s are the center of the i_kth fuzzy set and $a_{i_1 i_2 i_3}(z)$ is the nonlinear mapping defined as

$$a_{i_1 i_2 i_3}(z) = \frac{\prod_{k=1}^{3} m_k(e_k)}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\sum_{i_3=1}^{N_3}\prod_{k=1}^{3} m_k(e_k)}, \quad (15)$$

In Eq. (15), we have $\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\sum_{i_3=1}^{N_3} a_{i_1 i_2 i_3}(z) = 1$ and $a_{i_1 i_2 i_3}(z)$ can be considered as a weighting function. Thus, the above fuzzy inference can be written as

$$q = \mathrm{X}\boldsymbol{c} \quad (16)$$

where $c_{i_1 i_2 i_3}$'s are adaptable parameters and $\mathrm{X} = \mathrm{diag}\{\boldsymbol{a}^T, \boldsymbol{a}^T, \boldsymbol{a}^T\}$ with dimension $3\times(3N)$ for which $\boldsymbol{a} = (a_{111} \ \ a_{112} \ \ \ldots \ \ a_{11N_3} \ \ \ldots \ \ a_{N_1 N_2 N_3})^T$ is an $N\times 1$ fuzzy basis function vector and $a_{i_1 i_2 i_3}$'s are defined in Eq. (16), and $\boldsymbol{c} = (\boldsymbol{c}_1 \ \boldsymbol{c}_2 \ \boldsymbol{c}_3)^T$ in which $\boldsymbol{c}_l = (c_{111l} \ \ c_{112l} \ \ \ldots \ \ c_{11\ldots N_3 l} \ \ \ldots \ \ c_{N_1 N_2 N 1 l} \ \ \ldots \ \ c_{N_1 N_2 N_3 l})^T$, $l = 1,2,3.$

Now, the fuzzy system is employed to approximate the controller u^* given in Eq. (3) using the update control law for tuning the adaptable parameter vector q. To this reason, consider $q = \mathrm{X}\hat{\boldsymbol{c}}$ be the estimate of q^* because of $\hat{\boldsymbol{c}}$ and the vector of error given by $\tilde{\boldsymbol{c}} = \hat{\boldsymbol{c}} - \boldsymbol{c}^*$. From this, Eq. (3) can be rewritten as

$$u_{PID}(z\,|\,q) = z^T \mathrm{X}\hat{\boldsymbol{c}} \quad (17)$$

Substituting Eq. (17) into Eq. (12) yields

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}g(\boldsymbol{x})(\boldsymbol{W}\tilde{\boldsymbol{c}} + u) + \boldsymbol{B}w \quad (18)$$

where $\boldsymbol{W} = z^T \mathrm{X}$.

On the basis of the above discussion, the following theorem is given to demonstrate the stability of the control structure.

Theorem 1: Consider the system Eq. (1) with unknown bounded nonlinear functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$. If the adaptive fuzzy PID control law given in Eq. (5) is used with H_∞ tracking compensator given in Eq. (7) in which the weight vector is adaptively tuned according to

$$\dot{\hat{\boldsymbol{c}}} = \dot{\tilde{\boldsymbol{c}}} = -g\boldsymbol{W}^T g(\boldsymbol{x})\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{e} \quad (19)$$

Then, for any $t \ge t_0$, $\boldsymbol{e}(t)$ and $\tilde{\boldsymbol{c}}(t)$ are uniformly ultimately bounded (UUB) and the H_∞ tracking performance for the system satisfies the following relationship:

$$\int_0^T \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e}\,dt \le \boldsymbol{e}^T(0)\boldsymbol{P}\boldsymbol{e}(0) + \frac{1}{g}\tilde{\boldsymbol{c}}(0)^T\tilde{\boldsymbol{c}}(0) + r^2\int_0^T w^T \quad (20)$$

Proof: Consider the following Lyapunov function.

$$V = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{P}\boldsymbol{e} + \frac{1}{2g}\tilde{\boldsymbol{c}}^T\tilde{\boldsymbol{c}} \quad (21)$$

Differentiate Eq. (21) with respect to time along the trajectory (18), then

$$\begin{aligned}
\dot{V} &= \frac{1}{2}\dot{\boldsymbol{e}}^T \boldsymbol{P}\boldsymbol{e} + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{P}\dot{\boldsymbol{e}} + \frac{1}{g}\tilde{c}^T\dot{\tilde{c}} \\
&= \frac{1}{2}(\boldsymbol{e}^T \boldsymbol{A}^T + \tilde{c}^T \boldsymbol{W}^T g(\boldsymbol{x})\boldsymbol{B}^T + u\, g(\boldsymbol{x})\boldsymbol{B}^T + w^T \boldsymbol{B}^T)\boldsymbol{P}\boldsymbol{e} \\
&\quad + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{P}(\boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}g(\boldsymbol{x})\boldsymbol{W}\tilde{c} + \boldsymbol{B}g(\boldsymbol{x})u + \boldsymbol{B}w) + \frac{1}{g}\tilde{c}^T\dot{\tilde{c}} \\
&= \frac{1}{2}(\boldsymbol{e}^T \boldsymbol{A}^T \boldsymbol{P}\boldsymbol{e} + \tilde{c}^T \boldsymbol{W}^T g(\boldsymbol{x})\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + u\, g(\boldsymbol{x})\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} \\
&\quad + w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{A}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}g(\boldsymbol{x})\boldsymbol{W}\tilde{c} \\
&\quad + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}g(\boldsymbol{x})u + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w + \frac{1}{g}\tilde{c}^T\dot{\tilde{c}} \\
&= \frac{1}{2}[\boldsymbol{e}^T \boldsymbol{A}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{A}\boldsymbol{e} + 2\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}g(\boldsymbol{x})u + w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} \\
&\quad + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w] + \tilde{c}^T \boldsymbol{W}^T g(\boldsymbol{x})\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \frac{1}{g}\tilde{c}^T\dot{\tilde{c}}
\end{aligned} \tag{22}$$

From Eqs. (19) and (7), Eq. (22) can be rewritten as

$$\begin{aligned}
\dot{V} &= \frac{1}{2}[\boldsymbol{e}^T \boldsymbol{A}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{A}\boldsymbol{e} + 2\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}g(\boldsymbol{x})(-\frac{1}{l}g^{-1}(\boldsymbol{x})\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e}) \\
&\quad + w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w \\
&= \frac{1}{2}[\boldsymbol{e}^T \boldsymbol{A}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{A}\boldsymbol{e} - \frac{2\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e}}{l} \\
&\quad + w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w]
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2}\boldsymbol{e}^T[\boldsymbol{A}^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \frac{2}{l}\boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T \boldsymbol{P}]\boldsymbol{e} \\
&\quad + \frac{1}{2}[w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w]
\end{aligned} \tag{23}$$

Inserting (8) and (19) into (23), we have

$$\begin{aligned}
\dot{V} &= +\frac{1}{2}\boldsymbol{e}^T(-\boldsymbol{Q} - \frac{1}{r^2}\boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T \boldsymbol{P})\boldsymbol{e} + \frac{1}{2}(w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w) \\
&\quad - \frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} - \frac{1}{2}(\frac{1}{r^2}\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} - w^T \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} - \boldsymbol{e}^T \boldsymbol{P}\boldsymbol{B}w \\
&\quad + r^2 w^T w) + \frac{1}{2}r^2 w^T w \\
&= -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} - \frac{1}{2}(\frac{1}{r}\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} - rw)^T(\frac{1}{r}\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{e} - rw) \\
&\quad + \frac{1}{2}r^2 w^T w \\
&\le -\frac{1}{2}\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} + \frac{1}{2}r^2 w^T w
\end{aligned} \tag{24}$$

Since r is the constant prescribed attenuation level, from Eq. (24), it can be concluded that $\boldsymbol{e}$, $\boldsymbol{c}$, and q are UUB, for any $t \ge t_0$. Also, by integrating both sides of Eq. (24), after some manipulation, it concludes

$$V(t_f) - V(0) \le -\frac{1}{2r}\int_0^{t_f} \mathbf{e}^T \mathbf{Q}\mathbf{e}dt + \frac{1}{2}r\int_0^{t_f} w^2 dt \tag{25}$$

From there

$$\int_0^{t_f} \mathbf{e}^T \mathbf{Q}\mathbf{e}dt \le 2rV(0) - 2rV(t_f) + r^2\int_0^{t_f} w^2 dt \tag{26}$$

Since $V(t_f) \ge 0$, Eq. (26) can be rewritten as

$$\int_0^{t_f} \mathbf{e}^T \mathbf{Q}\mathbf{e}dt \le 2rV(0) + r^2\int_0^{t_f} w^2 dt \tag{27}$$

Substituting Eq. (28) into Eq. (27), the H_∞ tracking performance given in Eq. (20) is satisfied. This completes the proof.

4. Optimal Tracking Performance

When applying the controller into the system, the closed-loop error dynamic given in Eqs. (18) is provided corresponding to vector $\boldsymbol{k} = [k_1, k_2, ..., k_n]^T \in \Re^n$. The simplest method to choose the appropriate values for the vector $\boldsymbol{k}$ is trial and error technique, which is a time-consuming procedure. This type of tuning method becomes difficult and delicate without a systematic design method. To overcome this problem, it can be considered as an optimization problem. Here, to achieve an optimal performance, BSA is employed to determine the appropriate poles location by choosing a proper vector $\boldsymbol{k}$. In the following, first the BSA is explained briefly. Afterward, the optimization problem in hand is described.

4.1. Overview of BSA

BSA is a population-based iterative EA designed to be a global minimizer with five processes including selection-I, mutation, crossover and selection-II. Three basic genetic operators (initialization, selection, mutation, and crossover) are used to generate trial individuals. BSA has a random mutation strategy that utilizes only one direction individual for each target individual and randomly selects the direction individual from individuals of a randomly chosen previous generation. BSA utilizes a non-uniform crossover strategy that is much more complex than traditional crossover strategies. The procedure of BSA can be described briefly as follows. BSA initializes the population P as

$$P_{i,j} \approx U(low_j, up_j) \tag{28}$$

for $i = 1,2,3,...,N$ and $j = 1,2,3,...,D$, where N and D are the population size and the problem dimension, respectively, U is the uniform distribution and each P_i is a target individual in the population P.

Selection-I stage evaluates the population and determines the historical population *oldP* according to the obtained fitness value

$$if \quad a < b \quad then \quad oldP := P \big| a,b \approx U(0,1) \tag{29}$$

where := is the update operation.

Eq. (29) insures that BSA designates a population belonging to a randomly selected previous generation as the historical population and remembers this historical population until it is changed. Therefore, BSA has a memory.

During each generation, BSAs mutation process generates the initial form of the trial population using

$$Mutant = P + F.(oldP - P) \tag{30}$$

where F is the amplitude control factor that controls the amplitude of the search-direction matrix. Here, we use the value $F = 3.rand$, where $rndn \approx N(0,1)$ (N is the standard normal distribution).

After *oldP* is determined, the following equation is utilized to randomly alter the order of the individuals in *oldP*:

$$oldP := permuting(oldP) \tag{31}$$

where the *permuting* function is a random shuffling function.

Because the historical population is used in the calculation of the search direction matrix, the trial population is generated by taking partial advantage of its experiences from previous generations. After the new mutant operation is finished, the crossover process generates the final form of the trial population T .

The initial value of the trial population is Mutant, which has been set in the mutation process. Individuals with better fitness values for the optimization problem are utilized to progress the target population. The first step of the crossover process calculates a binary integer-valued matrix (map) of size $N.D$ that indicates the individuals of T to be manipulated using the relevant individuals of P . Then T is updated with $T_{n,m} := P_{n,m}$ if $map_{n,m} = 1$, where $n \in \{1,2,3,...,N\}$ and $m \in \{1,2,3,...,D\}$.

4.2. Procedure of poles location

To handle the optimization problem, first of all, two significant subjects must be defined. The first one is how to define a suitable objective function. The objective function assesses the location of the poles and returns a fitness value by considering the performance requirements. Here, the Sum of Squared Errors (SSE) is considered as

$$SSE = \sum_{k=1}^{N} [y(k) - y_d(k)]^2 = \sum_{k=1}^{N} e^2(k) \tag{32}$$

where e is the tracking error, N is the number of given sampling steps, $y(k)$ and $y_d(k)$ are the system output and reference input signal, respectively.

The second one is that how the pole locations should be represented as gens. As mentioned earlier, the poles are directly related to the elements of vector $\boldsymbol{k}$. As each gen can be considered as a solution, the position of each gen can be described as an n-dimensional vector including a set of all closed-loop poles for the error dynamic.

5. Simulations

To evaluate the performance of proposed approach, simulation is performs for tracking control of the chaotic Duffing forced-oscillation system. The dynamic equation of such system is given by [43]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -0.1x_2 - x_1^3 + 12\cos(t) + u + d \end{aligned} \tag{34}$$

This system is chaotic without control as shown in Fig. 1. The goal is to force the Duffing system to fulfil the H_∞ tracking performance with a desired output $y_d(t)$ in presence of external disturbance d and system uncertainty. To this end, it is assumed that the state of system x_1 follow the desired output $y_d(t) = \frac{p}{10}\sin(t)$ asymptotically whereas the initial state is $(x_1(0), x_2(0)) = (0.05, 0)$. In all the simulations, a system uncertainty term is set to $0.01\sin(x_1)$ and the external disturbance d is considered a square wave with amplitude ± 0.5 and period $2p$. A Gaussian noise is also inserted at the system with mean zero and variance of 0.005.

Simulation results are performed in two general parts. In the first part, to show the effect of poles location on the performance of the controller, three different scenarios are considered. In the second part, the BSA is employed to further improve the performance of the controller. In the first part, to show the different tracking related to the elements of vector $\boldsymbol{k}$, an arbitrary set of stable poles are considered for each scenario as follows.

$k_1 = (-5 \quad -5)^T$, $k_2 = (-10 \quad -5)^T$ and $k_3 = (-2 \quad -2)^T$ for scenario 1, 2 and 3, respectively corresponding to $\mathbf{A_1} = \begin{bmatrix} 0 & 1 \\ -5 & -5 \end{bmatrix}$, $\mathbf{A_2} = \begin{bmatrix} 0 & 1 \\ -10 & -5 \end{bmatrix}$ and $\mathbf{A_3} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}$ for scenario 1, 2 and 3, respectively. To do a fair comparison, a positive definite matrix Q = diag [2, 2], the desired attenuation level $r = 0.1$, the learning rate $l = 0.02$, and the tuning rate of the update law $g = 130$ are used in all scenarios. From Eq. (8), the positive matrices are obtained as $\boldsymbol{P}_1 = \begin{pmatrix} 2.2 & 0.2 \\ 0.2 & 0.24 \end{pmatrix}$, $\boldsymbol{P}_2 = \begin{pmatrix} 2.7 & 0.1 \\ 0.1 & 0.22 \end{pmatrix}$ and $\boldsymbol{P}_3 = \begin{pmatrix} 2.5 & 0.5 \\ 0.5 & 0.75 \end{pmatrix}$ for scenario 1, 2 and 3, respectively. Figs. 2–4 illustrate the trajectory of the reference input and output signals for scenario 1, 2 and 3, respectively. Fig. 5 shows the trajectory of errors for each scenario. The final controller parameters values are as $(k_{p1}, k_{i1}, k_{d1}) = (58.14, 59.66, 98.74)$, $(k_{p2}, k_{i2}, k_{d2}) = (33.03, 93.14, 131.42)$ and $(k_{p3}, k_{i3}, k_{d3}) = (71.50, 49.62, 56.12)$ for scenario 1, 2 and 3, respectively. Referring to these figures, it is apparent that the tracking performance of the controller relays certainly on the poles location of error dynamics.

In this point of view, to achieve the optimal tracking performance of the controller, BSA is employed to determine the appropriate pole locations of error dynamic as described in section 4.2. In BSA, we set $Mixrate = 0.03$ [31]. The BSA is coded in Matlab 7.0 and the simulations are run on a laptop computer with Corei3 2.1 GHz speed processor and 4 GB memory capacity. The population size is se to 20. The BSA runs 20 times independently and we record the SSE defined as Eq. (32). The statistical results including the best, worse, mean and standard deviation (Std.) are listed in Table 1. This table demonstrates that the std. of the solutions obtained by BSA is small which indicates the robustness of the algorithm in solving these problems. Applying this method, we obtain $k = (-0.1645 \quad -20)^T$ and $\boldsymbol{P} = \begin{pmatrix} 121.6628 & 6.0802 \\ 6.0802 & 0.3540 \end{pmatrix}$. The corresponding results are shown in Figs. 6 and 7 for the best objective function value. Fig. 6 represents the trajectory of the output x_1. Fig. 7 depicts the control effort signal. The trajectories of the control parameter (k_p, k_i and k_d) are also shown in Fig. 8 where the initial value the PID parameters is $[k_p, k_i, k_d] = [5, 5, 5]$. As this figure shows, the steady-state values of the parameters are $k_p = 26.44, k_i = 21.22$ and $k_d = 18.6$. The corresponding cost function is also shown in Fig. 9, respectively. Totally, it is obvious that the system has optimal performance with a continuous control effort while is robust in presence of the uncertainties and the external disturbance.

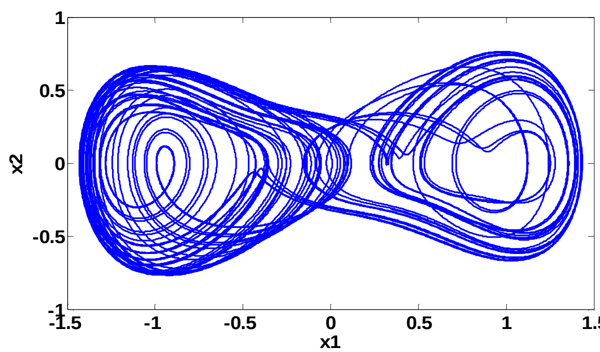

Fig. 1. Phase plane of the Duffing system.

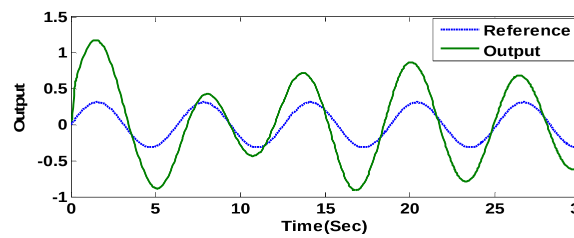

Fig 1. The output trajectory x_1 for scenario 1.

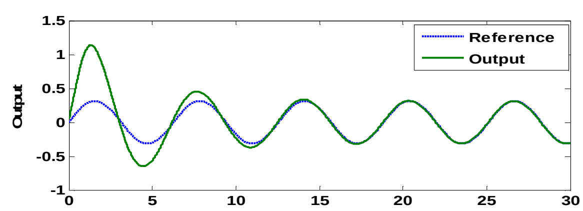

Fig 3. The output trajectory x_1 for scenario 2.

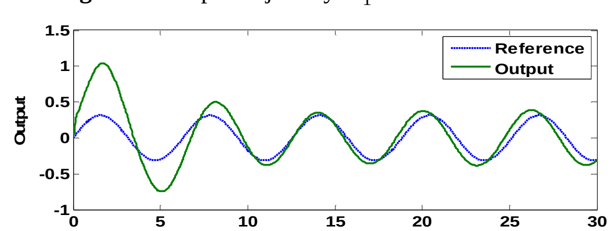

Fig. 4. The output trajectory x_1 for scenario 3.

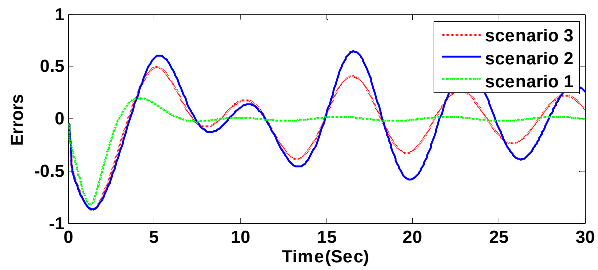

Fig. 5. The errors trajectory for all scenarios.

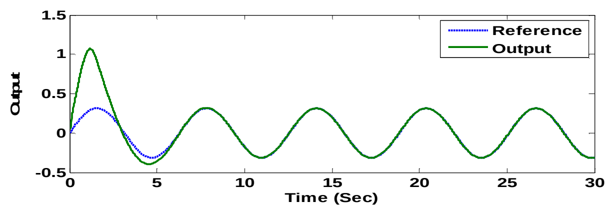

Fig. 6. The output trajectory x_1 using BSA.

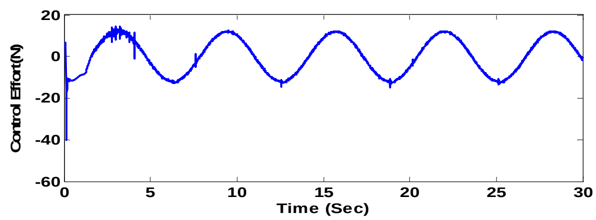

Fig. 7. The control effort signal using BSA.

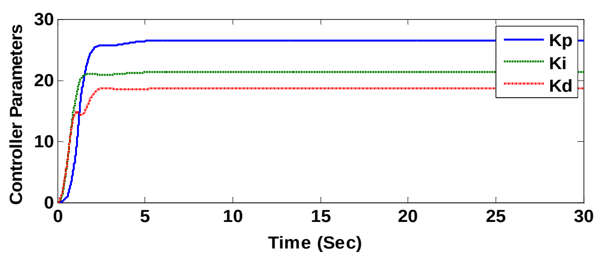

Fig. 8. The controller parameters trajectories using BSA.

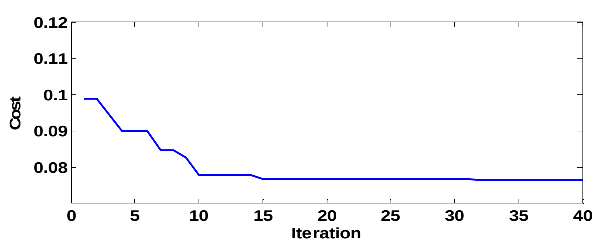

Fig. 9. The objective function using BSA.

Table 1. Statistical results obtained by BSA

Statistical results	Worst	Mean	Best	Std.
	0.0768731	0.0765174	0.0762264	0.0394541

7. Conclusions

This paper proposed the optimal controller design for controlling a class of chaotic systems in presence of the uncertainties and external disturbance. To achieve optimal tracking performance, BSA was employed to determine the proper poles of the dynamic of the error. Stability of the proposed control method was provided using Lyapunov theorem and the H_∞ tracking performance was guaranteed for a prescribed attenuation level. The capability of the controller was demonstrated by applying on the Duffing-oscillator system.

References

[1] Chen, G. 1999. Controlling chaos and bifurcations in engineering systems, Boca Raton, FL: CRC Press

[2] Kemih, K., Halimi, M., Ghanes, M., Fanit, H., Salit, H., 2014. Control and synchronization of chaotic attitude control of satellite with backstepping controller. The Eur. Phys. J. Spec. Top. 233, 1579–1587.

[3] Liu, L., Pu, J., Song, X., Fu, Z., Wang, X. 2014. Adaptive sliding mode control of uncertain chaotic systems with input nonlinearity. Nonlinear Dyn, 76, 1857–1865.

[4] Yu, S.H., Kang, H.S., Kim, Y.T., Hyun, C.H., Park, M. 2014. Fuzzy adaptive modular design of uncertain chaotic Duffing oscillators. Int. J. Control Autom. Syst. 12, 188–194.

[5] Zhao, J., Wu, Y., Liu, Q. 2014. Chaos synchronization between the coupled systems on network with unknown parameters. Appl. Math. Comput. 229, 254–259.

[6] Alfi, A., Akbarzadeh Kalat, A., Khooban, M.H. 2014. Adaptive fuzzy sliding mode control for synchronization of uncertain non-identical chaotic systems using bacterial foraging optimization. J. Intell. Fuzzy Syst. 26, 2567–2576.

[7] Chen, D., Zhao, W., Sprott, J.C., Ma, X. 2013. Application of Takagi–Sugeno fuzzy model to a class of chaotic synchronization and anti-synchronization. Nonlinear Dyn. 73, 1495–1505.

[8] Farivar, F., Aliyari Shoorehdeli, M.M., Nekoui, A., Teshnehlab, M. 2012. Chaos control and generalized projective synchronization of heavy symmetric chaotic gyroscope systems via Gaussian radial basis adaptive variable structure control. Chaos Solitons Fractals. 45, 80–97.

[9] Pourdehi, S., Karimaghaee, P., Karimipour, D. 2011. Adaptive controller design for lag-synchronization of two non-identical time-delayed chaotic systems with unknown parameters. Phys. Lett. A. 375, 1769–1778.

[10] Li, J., Li, W., Li, Q. 2012. Sliding mode control for uncertain chaotic systems with input nonlinearity. Commun. Nonlinear Sci. Numer. Simul. 17, 341–348.

[11] Poursamad, A., Davaie-Markazi, A. 2009. Robust adaptive fuzzy control of unknown chaotic systems. Appl. Soft Comput. 9, 970–976.

[12] Yahyazadeh, M., Ranjbar, Noei A., Ghaderi, R. 2011. Synchronization of chaotic systems with known and unknown parameters using a modified active sliding mode control. ISA Trans. 50, 262–267.

[13] Bouallegue, S., Haggege, J., Ayadi, M., Benrejeb, M. 2012. PID-type fuzzy logic controller tuning based on particle swarm optimization. Eng. Appl. Artif. Intell. 25 (2012) 484–493.

[14] Wang, L.X. 1997. A Course in Fuzzy System and Control. Prentice Hall, New Jersey.

[15] Wang, L.X. 1994. Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall, New Jersey.

[16] Taylor, D.G., Kokotovic, P.V., Marino, R., Kanellakopoulos, I. 1991. Adaptive regulation of nonlinear systems with unmodeled dynamics. IEEE Trans. Autom. Control. 36,1241–53.

[17] Ho, H.F., Wang Y.K., Rad A.B. 2003. Adaptive PID controller for nonlinear systems with H_∞ tracking performance, Proc. IEEE Phys. Con., pp. 1315–19.

[18] Gaun, X., Chen, C. 2003. Adaptive fuzzy control for chaotic systems with H_∞ tracking performance. Fuzzy Sets Syst. 139, 81–93.

[19] Alfi, A., 2012. Chaos suppression on a class of uncertain nonlinear chaotic systems using an H∞ adaptive PID controller. Chaos Solitons Fractals. 45, 351–357.

[20] Kung, C.C., Chen, T.H. 2007. H∞ tracking-based adaptive fuzzy sliding mode controller design for nonlinear systems, IET Control Theory Appl. 1, 82–89.

[21] Yu, W.S. 2006. Adaptive fuzzy PID control for nonlinear systems with H_∞ tracking performance. Proc IEEE Int. Conf. Fuzzy Syst., pp. 1010–15.

[22] Jiang, W., Guo-Dong, Q., Bin, D. 2005. H∞ variable universe adaptive fuzzy control for chaotic system. Chaos Solitons Fractals. 24, 1075–1086.

[23] Snyman, J.A. 2005. Practical mathematical optimization: an Introduction to basic optimization theory and classical and new gradient-based algorithms, Springer Publishing, ISBN 0-387-24348-8

[24] Kociecki, M., Adeli, H. 2014. Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvature. Eng. Appl.Artif.Intell. 32, 218–227.

[25] Kociecki, M., Adeli, H. 2015. Shape optimization of free-form steel space-frame roof structures with complex geometries using evolutionary computing. Eng. Appl.Artif.Intell. 38, 168–182.

[26] Kuila, P., Jana, P.K. 2014. Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach. Eng. Appl. Artif. Intell. 33, 127–140.

[27] Niknam, T., TaherianFard, E., Pourjafarian, N., Rousta, A. 2011. An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. Eng. Appl. Artif. Intell. 24, 306–317.

[38] Shokri-Ghaleh, H., Alfi, A. 2014. A comparison between optimization algorithms applied to synchronization of bilateral teleoperation systems against time delay and modeling uncertainties. Appl. Soft Comput. 24, 447–456.

[29] Shokri-Ghaleh, H., Alfi, A. 2014. Optimal synchronization of teleoperation systems via cuckoo optimization algorithm. Nonlinear Dyn. 78, 2359–2379.

[30] Alfi, A., Khosravi, A., Lari, A. 2013. Swarm-based structure-specified controller design for bilateral transparent teleoperation systems via m synthesis. IMA J. Math. Control Inf. 1–26, DOI.10.1093/imamci/dnt005.

[31] Jordehi, A.R. 2015. Optimal setting of TCSCs in power systems using teaching–learning-based optimisation algorithm, Neural Comput. Appl. DOI.10.1007/s00521-014-1791-x, 2015.

[32] Jordehi, A.R. 2014. A chaotic-based big bang–big crunch algorithm for solving global optimisation problems, Neural Comput. Appl. 25 (6), 1329–1335.

[33] Jordehi, A.R. 2014. Particle swarm optimisation for dynamic optimisation problems: a review, Neural Comput. Appl. 25 (7-8), 1507–1516.

[34] Alfi, A., Modares, H. 2011. System identification and control using adaptive particle swarm optimization, Appl. Math. Model. 35, 1210–1221.

[35] Darabi, A., Alfi, A., Kiumarsi, B., Modares, H. 2012. Employing adaptive PSO algorithm for parameter estimation of an exciter machine. ASME J. Dyn. Syst. Meas. Control. 134, http://dx.doi.org/10.1115/1.4005371.

[36] Elsayed, S.M., Sarker, R.A., Essam, D.L., 2014. A new genetic algorithm for solving optimization problems. Eng. Appl. Artif. Intell. 27, 57–69.

[37] Khooban, M.H., Alfi, A., Nazari Maryam Abadi, D. 2013. Teaching-learning-based optimal interval type-2 fuzzy PID controller design: A nonholonomic wheeled mobile robots. Robotica. 31, 1059–1071.

[38] Arab, A. Alfi, A. 2015. An adaptive gradient descent-based local search in memetic algorithm

applied to optimal controller design, Inf. Scien. 299, 117–142.
[39] Qing-dao-er-jia, R., Wangb, Y. 2012. A new hybrid genetic algorithm for job shop scheduling problem. Comput. Operat. Res. 39, 2291–2299.
[40] Civicioglu, P. 2013. Backtracking search optimization algorithm for numerical optimization problems. Appl. Math. Comput. 219, 8121–8144.
[41] Anderson, B.D.O, Moore J.B. 1990. Optimal Control: Linear Quadratic Methods. Engelwood Cliffs, NJ: Prentice-Hall.
[42] Golea, N., Golea, A., Benmahammed, K. 2002. Fuzzy model reference adaptive control. IEEE Trans. Fuzzy Syst. 10, 436–444.
[43] Chang, W.D., Hwang, R.C., Hsleh, J.G. 2002. A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach. J. Proc. Control. 12, 233–242.