In this paper, a novel adaptive fuzzy robust controller with a state observer approach based on the hybrid particle swarm optimization-simulated annealing (PSO-SA) technique for a class of multi-input multi-output (MIMO) nonlinear systems with disturbances is proposed. Firstly, particle swarm optimization-simulated annealing (PSO-SA) is used to adjust the fuzzy membership functions, while adaptive laws are used to approximate nonlinear functions and the unknown upper bounds of disturbances, respectively. Secondly, a state observer is applied for estimating all states which are not available for measurement in the system. By using the strictly-positive-real (SPR) stability theorem, the proposed adaptive fuzzy robust controller not only guarantees the stability of a class of MIMO nonlinear systems, but also maintains good tracking performance. Thirdly, we propose a novel auxiliary compensation, the item is designed to suppress the influence of external disturbance and remove fuzzy approximation error, respectively. The intelligence algorithm consists of the adaptive fuzzy robust method, the individual enhancement scheme and particle swarm optimization-simulated annealing structure which generates new optimal parameters for the control scheme. Finally, one simulation example is given to illustrate the effectiveness of the proposed approach.

Keywords: MIMO, Adaptive fuzzy control, Particle swarm optimization, Robust, Simulated annealing, Strictly-positive-real (SPR) stability theorem

1. Introduction. Adaptive fuzzy controllers provide a methodical and powerful framework for nonlinear control problems. In [1], an adaptive fuzzy control scheme has been developed for uncertain nonlinear systems. The stability of uncertain nonlinear systems has been addressed using adaptive fuzzy control approaches [2-14]. However, a significant constraint is that the system state variables must be measured. For many real systems, the state vector is rarely fully measured. There is thus a need to implement a state observer [15-21].

For a class of SISO systems, an adaptive fuzzy controller with a state observer has been proposed in [22,23]. According to the results, many observer-based adaptive fuzzy design schemes have been proposed in [24-32], where the fuzzy (neural network) can be used to approximate any nonlinear function. For example, some observer-based indirect adaptive fuzzy control schemes have been studied in [24,30] while direct control schemes were
addressed in [26,28,30]. However, in actuality, most engineering systems are multiple-input multiple-output (MIMO) systems [33-36]. MIMO systems can be found in nature, such as manipulators and chaotic systems, and are often composed of a set of subsystems [2,6,16,37-39]. Recently, adaptive fuzzy (neural network) control schemes that use universal approximators with adjustable parameters have been developed for MIMO nonlinear systems [4-7,40-44]. For example, several adaptive fuzzy control schemes were developed in [41-43] while an adaptive neural controller was designed in [44] to adjust parameters on-line.

Particle swarm optimization (PSO), proposed by Kennedy and Eberhart [45], is a new evolutionary algorithm that may be used to find optimal or near optimal solutions in a large search space. The PSO algorithms for tuning the membership function parameters of fuzzy logic controllers have been studied extensively in the literature [46-49]. However, for the basic PSO algorithm, all particles are usually trapped into the local minimum, and the optimal value found is thus often a local minimum instead of a global minimum. To overcome the shortcomings of the basic PSO algorithm, our method combines the PSO algorithm with the simulated annealing (SA) algorithm. SA is a kind of stochastic method that is well known for effectively bypassing the local minimum trap.

The control theory for nonlinear multivariable systems has universal applications. The control of multivariable systems is a difficult problem due to the coupling between the control inputs and the subsystems. Meanwhile, the states of multivariable systems are unmeasured; therefore, the control problem becomes more challenging. This paper proposes an observer-based adaptive fuzzy robust control scheme for a class of MIMO nonlinear systems with external disturbances. A state observer is introduced to estimate the unmeasured states. In the controller design, we assume that the systems are controllable and propose a novel auxiliary compensation; the control is designed to suppress the influence of external disturbance and remove the fuzzy approximation error, respectively. Compared with existing designs, the proposed method has four main contributions: (1) an adaptive fuzzy robust tracking control method for a class of MIMO systems is designed; (2) the controller does not require a priori knowledge of the sign of the control coefficient; (3) the PSO-SA algorithm is used to self-adjust the controller’s coefficient for the optimal solution; and (4) a novel auxiliary compensation is introduced to eliminate external disturbance and fuzzy approximation error. All parameter adaptive laws and robust control terms are derived based on Lyapunov’s stability analysis, so that convergence to zero tracking error and boundedness of all signals are guaranteed.

The rest of this paper is organized as follows: in Section 2, the system description and the design of the fuzzy logic system are presented; the adaptive fuzzy robust controller based on the PSO-SA algorithm combined with an observer for a MIMO nonlinear system is described in Section 3; simulation results are presented to validate the proposed control scheme in Section 4; and finally, the conclusions are given in Section 5.

2. Problem Statement and Preliminaries. In this section, we shall describe the plant and some basic concepts of fuzzy set theory and fuzzy logic used in this paper.

2.1. System description. Consider a class of MIMO nonlinear systems as represented by the following differential equations:

\[ \dot{x}_{11}(t) = x_{12}(t), \]
\[ \vdots \]
\[ \dot{x}_{1(n_1-1)} = x_{1n_1}(t), \]
The controller makes the system output track the given desired signal

\begin{equation}
\dot{x}_{1n_1}(t) = f_1(x(t)) + \sum_{j=1}^{p} g_{1j} (x(t)) u_j(t) + \Delta d_1 (x(t)),
\end{equation}

\begin{equation}
\dot{x}_{p1}(t) = x_{p2}(t),
\end{equation}

\vdots

\begin{equation}
\dot{x}_{p(n_p-1)}(t) = x_{pn_p}(t),
\end{equation}

\begin{equation}
\dot{x}_{pn_p}(t) = f_p (x(t)) + \sum_{j=1}^{p} g_{pj} (x(t)) u_j(t) + \Delta d_p (x(t)),
\end{equation}

\begin{equation}
y_1(t) = x_{11}(t),
\end{equation}

\vdots

\begin{equation}
y_p(t) = x_{p1}(t).
\end{equation}

(1a)-(1d) can be rewritten as:

\begin{equation}
\dot{x}(t) = Ax(t) + B [F (x(t)) + G (x(t)) u(t) + \Delta d (x(t))],
\end{equation}

\begin{equation}
y(t) = Cx(t).
\end{equation}

Define:

\begin{equation}
F(x) = [f_1(x), \cdots, f_p(x)]^T, \quad G_i(x) = [g_{1i}(x), \cdots, g_{pi}(x)]^T, \quad i = 1, \ldots, p.
\end{equation}

\begin{equation}
G(x) = [G_1(x), \cdots, G_p(x)],
\end{equation}

\begin{equation}
A_i = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}_{n_i \times n_i}, \quad B_i = \begin{bmatrix}
0 \\
\vdots \\
1
\end{bmatrix}_{n_i \times 1},
\end{equation}

\begin{equation}
C_i = \begin{bmatrix}
1 & 0 & \cdots & 0
\end{bmatrix}_{1 \times n_i},
\end{equation}

\begin{equation}
A = \text{diag} [A_1, \cdots, A_p], \quad B = \text{diag} [B_1, \cdots, B_p], \quad C = \text{diag} [C_1, \cdots, C_p],
\end{equation}

where \( x = [x_{11}, \cdots, x_{1m_1}, \cdots, x_{p1}, \cdots, x_{pm_p}]^T \in \mathbb{R}^n \) is the state vector, which is assumed to be unavailable for measurement, and \( u = [u_1, \cdots, u_p]^T \in \mathbb{R}^p \) and \( y = [y_1, \cdots, y_p]^T \in \mathbb{R}^n \) are the state vector, the control input and the output of the system, respectively, while \( n = n_1 + \cdots + n_p \). \( f_i (x(t)) \) and \( g_{ij} (x(t)) \) are smooth nonlinear functions, and \( \Delta d = [\Delta d_1, \cdots, \Delta d_p]^T \) is the vector of external disturbance. The following assumptions are made for the controller design:

**Assumption 2.1.** [50] The matrix \( G(x) \) as previously defined is nonsingular, i.e., \( G^{-1}(x) \) exists, and is bounded for all \( x \in U \), where \( U \in \mathbb{R}^n \) is a compact set.

**Control objectives.** To design a stabilizing controller for the system described by Equation (2) allowing the tracking error to converge to zero asymptotically. The stabilizing controller makes the system output track the given desired signal \( y_{pd} \).

Then, the derivation of the output can be expressed as:

\begin{equation}
y_i(t) = x_{i1}(t), \quad i = 1, \cdots, p
\end{equation}

\vdots

\begin{equation}
y_{i(n_i-1)}(t) = x_{i(n_i-1)}(t),
\end{equation}

\begin{equation}
y_{i(n_i)}(t) = \dot{x}_{i(n_i)}(t)
\end{equation}
\begin{equation}
= f_i(x(t)) + \sum_{j=1}^{p} g_{ij}(x(t)) u_j(t) + \Delta d_i(x(t)),
\end{equation}

where \( i = 1, \ldots, p \). Let \( y_{id}, \ldots, y_{pd} \) denote the desired bounded output signal that contains finite derivatives up to the \( n \)th order. The tracking errors of the system are defined as follows:
\begin{align}
\dot{e}_{i\ell}(t) &= y_{i\ell}^{(\ell-1)}(t) - y_{id}^{(\ell-1)}(t), \quad i = 1, \ldots, p, \quad \ell = 1, \ldots, n_i, \\
&\vdots \\
\dot{e}_{in}(t) &= y_{in}^{(n-1)}(t) - y_{id}^{(n-1)}(t),
\end{align}

\begin{equation}
= f_i(x(t)) + \sum_{j=1}^{p} g_{ij}(x(t)) u_j(t) - y_{id}^{(n)}(t) + \Delta d_i(x).
\end{equation}

Equation (4) can be shown as:
\begin{equation}
\dot{e}_i(t) = A_i e_i(t) + B_i \left[ f_i(x(t)) + \sum_{j=1}^{p} g_{ij}(x(t)) u_j(t) - y_{id}^{(n)}(t) + \Delta d_i(x) \right],
\end{equation}

Equation (5) can be rewritten as:
\begin{equation}
\dot{e}(t) = A e(t) + B \left[ F(x(t)) + G(x(t)) u(t) - y_{id}^{(n)}(t) + \Delta D(x) \right],
\end{equation}

where \( e_i = [e_{i1}, e_{i2}, \ldots, e_{in}]^T \), \( e = [e_1, \ldots, e_p]^T \), \( e_q = [e_{i1}, e_{i2}, \ldots, e_{ip}]^T \), \( \Delta D = [\Delta d_1, \ldots, \Delta d_p]^T \) and \( y_{id}^{(n)} = [y_{id_{11}}, \ldots, y_{id_{1p}}]^T \).

For later derivations, the following assumption should be satisfied.

**Assumption 2.2.**
\begin{equation}
||\Delta D(x)|| \leq g_d(x),
\end{equation}

where \( g_d(x) \) is the unknown positive smooth continuous function.

If the nonlinear functions \( f_i(x(t)) \) and \( g_{ij}(x(t)) \) are exactly known, and \( G(x(t)) \neq 0 \) based on the certainty equivalent approach, the ideal control law can be obtained as:
\begin{equation}
u(t) = G^{-1}(x(t)) \left[ -F(x(t)) + y_{id}^{(n)} - K_c e(t) + u_a(t) \right],
\end{equation}

where \( K_c = [K_{c1}, \ldots, K_{cp}] \) is a control gain vector and \( u_a(t) \) is the flexible auxiliary compensation, which is designed for compensating the fuzzy approximation errors and the external disturbance.

### 2.2. Fuzzy system description.

The basic configuration of the fuzzy logic system consists of four main components: fuzzy rule base, fuzzy inference engine, fuzzifier and defuzzifier. The fuzzy rule base consists of a collection of fuzzy IF-THEN rules:
\begin{equation}
R^{(l)}: \text{If } x_1 \text{ is } F_{1l}^i \text{ and } x_2 \text{ is } F_{2l}^i \ldots \text{ and } x_n \text{ is } F_{nl}^i, \quad l = 1, 2, \ldots, n,
\text{then } y_s = G_{sl}^i, \quad s = 1, 2, \ldots, p,
\end{equation}

where \( x = [x_1, \ldots, x_n]^T \) is the fuzzy system input; \( y \) is the fuzzy system output; and \( F_{il}^i \) and \( G_{sl}^i \) are fuzzy sets. Through singleton fuzzification, center average defuzzification and
product inference, the output of the fuzzy logic system can be expressed as:

\[ y_s = \theta_s^T \xi(x), \quad s = 1, \ldots, p, \]  \tag{10} 

where \( \theta_s^T = [\theta_1^s, \ldots, \theta_p^s] \), with each variable \( \theta_i^s \) being the point at which the fuzzy membership function of \( G_i^s \) achieves the maximum value; and \( \xi(x) = [\xi_1(x), \ldots, \xi_n(x)]^T \) with each variable \( \xi_i \) being the fuzzy basis function defined as:

\[ \xi_i(x) = \prod_{i=1}^{n} \mu_{F_i}(x_i) \left/ \sum_{i=1}^{N} \left( \prod_{i=1}^{n} \mu_{F_i}(x_i) \right) \right. \]  \tag{11} 

where \( \mu_{F_i}(x_i) \) is the membership function of the fuzzy set. Allow \( \theta^T = \text{diag}[\theta_1^T, \ldots, \theta_p^T] \) and \( \Phi(x) = \text{diag}[\xi(x), \ldots, \xi(x)] \).

MIMO fuzzy systems can be rewritten as:

\[ y = \theta^T \Phi(x), \]  \tag{12} 

Define the nonlinear functions \( f_i(x(t)) \) and \( g_{ij}(x(t)) \) to be approximated by fuzzy logic systems as:

\[
\begin{align*}
    f_i(x|\theta_f^i) &= \theta_{f_i}^T \xi(x), \quad i = 1, 2, \ldots, p, \\
    g_{ij}(x|\theta_g^i, \theta_g^j) &= \theta_{g_{ij}}^T \xi(x), \quad i, j = 1, 2, \ldots, p, \\
    \Phi(x|\theta_f) &= \theta_f^T \xi_f(x), \\
    \Phi(x|\theta_g) &= \theta_g^T \xi_g(x).
\end{align*}
\]  \tag{13} 

Based on approximation theory, fuzzy inference systems with specific operations are universal approximators that can be used to model the nonlinear functions of the systems.

3. Main Results. In this section, the adaptive fuzzy controller with the PSO-SA design method will be described.

3.1. Basic PSO algorithm. PSO is a stochastic optimization algorithm [51-56]. The main concept of PSO is the mathematical modeling and simulation of the food searching activities of a flock of birds. The PSO algorithm includes two parts: (1) Cognitive component; and (2) Social component. First, the cognitive component models the memory of the each particle about its previous best position. Second, the social component models the memory of the each particle about the best position among the particles. From the above discussion, the mathematical model for the PSO algorithm can be written as:

\[
\begin{align*}
    V_i(k+1) &= V_i(k) + c_1 \cdot \text{rand}_1(\cdot) \times (P_{\text{best}_i} - X_i(k)) \\
    &\quad + c_2 \cdot \text{rand}_2(\cdot) \times (G_{\text{best}_n} - X_i(k)), \\
    X_i(k+1) &= X_i(k) + V_i(k+1), \quad i = 1, \ldots, N_{\text{swarm}},
\end{align*}
\]  \tag{15} 

where \( V_i \) is the velocity of particle \( i \); \( X_i \) is the position of particle \( i \); \( P_{\text{best}_i} \) and \( G_{\text{best}_n} \) are the best historical position of particle \( i \) and the best historical position of the neighbor of particle \( i \) with objective values \( F_{p_i} \) and \( F_{g_n} \), respectively. \( \text{rand}_1(\cdot) \) and \( \text{rand}_2(\cdot) \) are random numbers between 0 and 1, while constants \( c_1 \) and \( c_2 \) are the weighting factors. Usually \( G_{\text{best}_n} \) is often replaced by \( G_{\text{best}_g} \), which is the global best position. The method has the drawback of being easily trapped in local minima. The following intelligence algorithm was thus proposed to overcome this shortcoming.
3.2. Simulated annealing. The simulated annealing (SA) algorithm is introduced to prevent particles from becoming trapped in a local optimum and to increase the diversity of the particle swarm. It was first put forward by Metropolis and successfully applied to optimization problems by Kirkpatrick [57].

The most significant character of SA is the probabilistic jumping property; i.e., a bad solution has a probability of being accepted as the new solution. Moreover, by adjusting the temperature, the jumping probability can be controlled. In particular, the probability decreases with decreasing temperature; and when the temperature moves toward zero, the probability approaches zero. In this scenario, better solutions are accepted. It has been proven that the SA algorithm is globally convergent with a probability of 1. In this section, we incorporate the mechanism of SA into PSO to create a hybrid optimization strategy, named PSO-SA.

As just mentioned, in PSO, $G_{best}^g$ is an element of the set of all $P_{best}^i$, which can be considered as a set of local optima. Therefore, we modify the selection of $G_{best}^g$ to overcome premature convergence. We utilize the jumping mechanism of the SA algorithm for the selection of $G_{best}^n$ instead of $G_{best}^g$. When $P_{best}^i$ has better features, it should have a larger probability of being selected for $G_{best}^g$. By employing the mechanism of the SA algorithm, all other $P_{best}^i$ can be considered as solutions worse than $G_{best}^g$.

Meanwhile, we can set $e^{-(F_{P_{best}^i} - F_{G_{best}^g})/t}$ as the fitness function of each $P_{best}^i$ to change $G_{best}^g$ at a temperature $t$. The largest fitness value belongs to $G_{best}^g$ and it’s value is 1, with other fitness values within the interval $(0, 1]$. Hence, the probability of selecting $P_{best}^i$ as $G_{best}^g$ can be described as:

$$\frac{e^{-(F_{P_{best}^i} - F_{G_{best}^g})/t}}{N_{swarm} \sum_{j=1}^{N_{swarm}} e^{-(F_{P_{best}^j} - F_{G_{best}^g})/t}}.$$ (16)

Then, roulette wheel selection can be used to randomly determine which $P_{best}^i$ to select as $G_{best}^g$.

3.3. Optimization of fuzzy logic control with PSO-SA. The fuzzy logic controller has 60 parameters to be adjusted. Now, the particles are chosen as:

$$P_i = [^i\sigma_{11} ^i c_{11} \cdots ^i \sigma_{15} ^i c_{15}], \quad i = 1, 2, 3,$$

where parameters $\sigma$ and $c$ represent the center and deviation of the Gaussian membership functions, respectively. The whole procedure of the PSO-SA algorithm can be written as follows:

Step 1: Let $k = 0$, and randomly initialize $X_i(0)$ and $V_i(0)$, $i = 1, \cdots, N_{swarm}$. Then, evaluate objective function $F_i$ for all particles.

Step 2: Initialize $P_{best}^i$ with a copy of $X_i(0)$, $i = 1, \cdots, N_{swarm}$. Meanwhile, initialize $G_{best}^g$ with a copy of the best $P_{best}^i$, and define the temperature $t(0) = t_0$.

Step 3: Repeat until the stopping criterion is satisfied.

Calculate the fitness:

$$\text{Fitness}_i = \exp\left[-(F_{P_{best}^i} - F_{G_{best}^g})/t(k)\right],$$

where $F_{P_{best}^g} = \frac{1}{N_{swarm}} \sum_{j=1}^{N_{swarm}} \sqrt{e^2(j)}$. (19)

Step 4: For every particle $i$, apply roulette wheel selection from the set of all $P_{best}^i$, and then update $V_i$ and $X_i$ via the following equations:
V_i(k+1) = \chi \times \{ V_i(k) + c_1 \cdot \text{rand}_1(\cdot) \times (P\text{best}_i - X_i(k)) \\
+ c_2 \cdot \text{rand}_2(\cdot) \times (G\text{best}_n - X_i(k)) \},
(20)

X_i(k+1) = X_i(k) + V_i(k+1),

where \( \chi \) is a constriction coefficient that is defined as:

\[
\chi = \frac{2}{2 - C - \sqrt{C^2 - 4C}}, \quad C = c_1 + c_2.
\]

Step 5: Evaluate the new objective value \( F_i \) for all particles, and then update \( P\text{best}_i \) and \( G\text{best}_g \) (including position values and objective values).

Step 6: Set the annealing temperature to obtain \( t(k+1) \), and let \( k = k + 1 \).

Step 7: The best solution \( G\text{best}_g \) is then obtained.

Remark 3.1. The mechanism of the SA algorithm is combined with the PSO algorithm in this paper. The main purpose of employing the SA algorithm is to balance exploration and exploitation; the evolving process may be prolonged as a result of overcoming premature convergence.

Remark 3.2. The initial temperature \( t_0 \) and the method of annealing play important roles in the algorithm, so the initial value affects the performance of the PSO-SA algorithm. The initial temperature can be described as:

\[
t_0 = -F_{G\text{best}_g}/\ln(0.8),
\]

For the annealing algorithm, an exponential annealing function, i.e., \( k = k + 1 = \lambda \cdot t(k) \), is employed, where \( 0 < \lambda < 1 \) denotes the annealing rate.

The controller design procedure is discussed in the next section.

3.4. Observer-based adaptive fuzzy robust controller design. Utilizing fuzzy logic systems to approximate \( f_i(x(t)) \), \( g_{ij}(x(t)) \) and \( g_d(x) \):

\[
\begin{align*}
\hat{F}(x|\theta_f) &= \theta_f^T \xi_f(x), \\
\hat{G}(x|\theta_g) &= \theta_g^T \xi_g(x), \\
\hat{g}_d(x|\theta_{gd}) &= \theta_{gd}^T \xi_{gd}(x).
\end{align*}
\]

The fuzzy control law in Equation (8) can be rewritten as:

\[
u(t) = \hat{G}^{-1}(x(t)|\theta_g) \left[ -\hat{F}(x(t)|\theta_f) + y_d^{(n)} - K_c e(t) + u_o(t) \right],
\]

where \( K_c = [K_{c1}, \cdots, K_{cp}] \) is the control gain vector, and

\[
u_o = -\hat{w}_c - \frac{B_c^T P \hat{e}_c}{\|\hat{e}_c B_c\|^2} \left( \hat{g}_d(\hat{x}|\theta_{gd}) - \frac{B_c^T P \hat{e}_c}{\|\hat{e}_c B_c\|^2} [\hat{e}_c P_2 K_c \hat{e}_1] \right).
\]

The parameter update laws are as follows:

\[
\begin{align*}
\dot{\theta}_f &= \gamma_f \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c, \\
\dot{\theta}_g &= \gamma_g \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c, \\
\dot{\theta}_{gd} &= \gamma_{gd} \|\hat{e}_c^T P_1 B_c\| \xi_{gd}(\hat{x}), \\
\dot{\theta}_c &= \gamma_c \hat{e}_c^T P_1 B_c,
\end{align*}
\]

where \( \gamma_f > 0, \gamma_g > 0, \gamma_{gd} > 0 \) and \( \gamma_c > 0 \) are positive adaptive gains to be determined.
In order to constrain the parameters $\theta_f$, $\theta_g$ and $\theta_{gd}$ within the sets $M_f$, $M_g$ and $M_{gd}$, respectively, the following parameter projection algorithm was used. Therefore, the proposed adaptive laws (28)-(30) can be modified as follows:

$$
\dot{\theta}_f = \begin{cases} 
\gamma_f \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c & \text{if } (\|\theta_f\| \leq M_f) \\
\text{or } (\|\theta_f\| = M_f \text{ and } \hat{e}_c^T P_1 B_c \theta_f^T \xi_f(\hat{x}) \leq 0) \\
P \{ \gamma_f \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c \} & \text{if } (\|\theta_f\| = M_f \text{ and } \hat{e}_c^T P_1 B_c \theta_f^T \xi_f(\hat{x}) > 0)
\end{cases}
$$  \hspace{1cm} (32)

where $P \{ \gamma_f \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c \} = \gamma_f \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c - \gamma_f \frac{\theta_f^T}{\|\theta_f\|} \xi_f(\hat{x}) \hat{e}_c^T P_1 B_c$.

$$
\dot{\theta}_g = \begin{cases} 
\gamma_g \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c & \text{if } (\|\theta_g\| \leq M_g) \\
\text{or } (\|\theta_g\| = M_g \text{ and } \hat{e}_c^T P_1 B_c \theta_g^T \xi_g(\hat{x}) u \leq 0) \\
P \{ \gamma_g \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c \} & \text{if } (\|\theta_g\| = M_g \text{ and } \hat{e}_c^T P_1 B_c \theta_g^T \xi_g(\hat{x}) u > 0)
\end{cases}
$$  \hspace{1cm} (33)

where $P \{ \gamma_g \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c \} = \gamma_g \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c - \gamma_g \frac{\theta_g^T}{\|\theta_g\|} \xi_g(\hat{x}) u \hat{e}_c^T P_1 B_c$.

$$
\dot{\theta}_{gd} = \begin{cases} 
\gamma_{gd} \| \hat{e}_c^T P_1 B_c \| \xi_{gd}(\hat{x}) & \text{if } (\|\theta_{gd}\| \leq M_{gd}) \\
\text{or } (\|\theta_{gd}\| = M_{gd} \text{ and } \| \hat{e}_c^T P_1 B_c \| \theta_{gd}^T \xi_{gd}(\hat{x}) \leq 0) \\
P \{ \| \hat{e}_c^T P_1 B_c \| \xi_{gd}(\hat{x}) \} & \text{if } (\|\theta_{gd}\| = M_{gd} \text{ and } \| \hat{e}_c^T P_1 B_c \| \theta_{gd}^T \xi_{gd}(\hat{x}) u > 0)
\end{cases}
$$  \hspace{1cm} (34)

where $P \{ \gamma_{gd} \xi_{gd}(\hat{x}) \hat{e}_c^T P_1 B_c \} = \gamma_{gd} \xi_{gd}(\hat{x}) \hat{e}_c^T P_1 B_c - \gamma_{gd} \frac{\theta_{gd}^T}{\|\theta_{gd}\|} \xi_{gd}(\hat{x}) \hat{e}_c^T P_1 B_c$.

Hence, the control law can be chosen as:

$$
u(t) = \hat{G}^{-1}(\hat{x}(t) \mid \theta_g) \left[ -\hat{F}(\hat{x}(t) \mid \theta_f) + y_{d}^{(n)} - K_c e(t) + u_a(t) \right].$$  \hspace{1cm} (35)

Using Equations (35) and (6) yields:

$$
\dot{e}(t) = A e(t) - B K_c \hat{e}(t) + B \left\{ F(x(t) - \hat{F}(\hat{x}(t) \mid \theta_f)) + \left[ G(x(t) - \hat{G}(\hat{x}(t) \mid \theta_g)) \right] u(t) + u_{n}(t) + \Delta D(x) \right\},
$$  \hspace{1cm} (36)

$$
e_{q}(t) = C e(t).
$$

In order to estimate the output tracking error vector, we employ the observer as follows:

$$
\dot{\hat{e}}(t) = (A - B K_c) \hat{e}(t) + K_o (e_{q}(t) - \hat{e}_{q}(t)),
$$  \hspace{1cm} (37)

$$
\hat{e}_{q}(t) = C \hat{e}(t).
$$

where $K_o = [K_{o1}, \cdots, K_{op}]$ is the observer gain vector. Define the observation error vector as:

$$
\hat{e}(t) = e(t) - \hat{e}(t).
$$  \hspace{1cm} (38)

By utilizing Equations (36) and (37), then Equation (38) can be written as:

$$
\hat{e} = (A - K_o C) \hat{e} + B \left\{ F(x(t)) - \hat{F}(\hat{x}(t) \mid \theta_f) \right\} + \left[ G(x(t)) - \hat{G}(\hat{x}(t) \mid \theta_g) \right] u(t) + u_{n}(t) + \Delta D(x),
$$  \hspace{1cm} (39)

$$
\hat{e}_{q}(t) = C \hat{e}(t).$$
respectively, can be described as:

\[
\begin{align*}
U_x &= \{ x \in \mathbb{R}^n : \| x \| \leq M_x < \infty \} \\
U_{\hat{x}} &= \{ \hat{x} \in \mathbb{R}^n : \| \hat{x} \| \leq M_{\hat{x}} < \infty \} \\
\Omega_f &= \{ \theta_f \in \mathbb{R}^N : \| \theta_f \| \leq M_f < \infty \} \\
\Omega_g &= \{ \theta_g \in \mathbb{R}^N : \| \theta_g \| \leq M_g < \infty \} \\
\Omega_{gd} &= \{ \theta_{gd} \in \mathbb{R}^N : \| \theta_{gd} \| \leq M_{gd} < \infty \}
\end{align*}
\]

where \( M_x, M_{\hat{x}}, M_f, M_g \) and \( M_{gd} \) are the parameters, and \( N \) is the number of fuzzy logic system rules. Define the optimal parameter vectors \( \theta_f^*, \theta_g^* \) and \( \theta_{gd}^* \) as:

\[
\begin{align*}
\theta_f^* &= \arg \min_{\theta_f \in \Omega_f} \left\{ \sup_{x \in U_x, \hat{x} \in U_{\hat{x}}} \left| F(x) - \hat{F}(\hat{x} | \theta_f) \right| \right\}, \\
\theta_g^* &= \arg \min_{\theta_g \in \Omega_g} \left\{ \sup_{x \in U_x, \hat{x} \in U_{\hat{x}}} \left| G(x) - \hat{G}(\hat{x} | \theta_g) \right| \right\}, \\
\theta_{gd}^* &= \arg \min_{\theta_{gd} \in \Omega_{gd}} \left\{ \sup_{x \in U_x, \hat{x} \in U_{\hat{x}}} \left| g_{gd}(x) - \hat{g}_{gd}(\hat{x} | \theta_{gd}) \right| \right\},
\end{align*}
\]

where \( \theta_f^* - \theta_f, \theta_g^* - \theta_g \) and \( \theta_{gd}^* - \theta_{gd} \) are parameter estimation errors. Therefore, the minimum approximation errors can be denoted as:

\[
w = F'(x(t)) - \hat{F}(\hat{x} | \theta_f^*) + \left( G(x(t)) - \hat{G}(\hat{x} | \theta_g^*) \right) u + g_{gd}(x(t)) - \hat{g}_{gd}(\hat{x} | \theta_{gd}^*). \tag{44}
\]

Substituting Equation (44) into (39) yields:

\[
\begin{align*}
\dot{e} &= (A - K_0 C) \dot{e} + B \left\{ F'(x(t)) - \hat{F}(\hat{x} | \theta_f) + \hat{F}(\hat{x}(t) | \theta_f^*) - \hat{F}(\hat{x}(t) | \theta_f) \right. \\
&\quad + \left[ G(x(t)) - \hat{G}(\hat{x}(t) | \theta_g) + \hat{G}(\hat{x}(t) | \theta_g^*) - \hat{G}(\hat{x}(t) | \theta_g^*) \right] u(t) \\
&\quad + u_a(t) + \Delta D(x) \left\},
\end{align*}
\]

\[
\dot{\theta}_f = (A - K_0 C) \dot{\theta}_f + B \left\{ \theta_f^T \xi_f(\hat{x}) + \theta_g^T \xi_g(\hat{x}) u(t) + F(x(t)) - \hat{F}(\hat{x}(t) | \theta_f^*) \\
&\quad + \left[ G(x(t)) - \hat{G}(\hat{x}(t) | \theta_f^*) \right] u(t) + u_a(t) + \Delta D(x) \right\}, \tag{46}
\]

Then, the output error dynamics of (46) can be described as:

\[
\dot{\epsilon} = H(s) \left[ \dot{\theta}_f^T \xi_f(\hat{x}) + \dot{\theta}_g^T \xi_g(\hat{x}) u(t) + w + u_a(t) + \Delta D(x) \right].
\]

The linear transfer function \( H(s) \) is realized using standard techniques. By using the SPR Lyapunov design method, and the following equation can be obtained:

\[
\begin{align*}
\dot{\epsilon}_q(t) &= H(s)L(s) \left[ L(s)^{-1} \dot{\theta}_f^T \xi_f(\hat{x}) + L(s)^{-1} \dot{\theta}_g^T \xi_g(\hat{x}) u(t) \\
&\quad + L(s)^{-1} (w + u_a(t) + \Delta D(x)) \right], \tag{48}
\end{align*}
\]

where \( L(s) = [L_1(s), \ldots, L_p(s)] \) and \( L_i(s) = s^{m_i} + b_{i1} s^{m_i-1} + \cdots + b_{i_m_i}, m_i = n_i - 1, i = 1, 2, \cdots, p. \) After some further manipulations, Equation (48) can be rewritten as
follows:

\[
\dot{\tilde{e}_c} = (A - K_cC) \tilde{e}_c + B_c \left[ \tilde{\theta}_f^T \xi_{cf}(\hat{x}) + \tilde{\theta}_g^T \xi_{cg}(\hat{x}) u(t) + \omega_c + u_{ca}(t) + \Delta D_c(x) \right],
\]

\[
\dot{\tilde{e}_q}(t) = \mathbf{C} \tilde{e}_c(t),
\]

where \( B_c = \text{diag} \{ B_{c1}, \ldots, B_{cp} \}, B_{ci} = \text{diag} \{ 1, b_{i1}, \ldots, b_{im} \} \), \( \xi_{cf}(\hat{x}) = \mathbf{L}(s)^{-1} \xi_f(\hat{x}) \), \( w_c = \mathbf{L}(s)^{-1} w_i \), \( \Delta D_c(x) = \mathbf{L}(s)^{-1} \Delta \mathbf{D}(x) \) and \( \xi_{cg}(\hat{x}) = \mathbf{L}(s)^{-1} \xi_g(\hat{x}) \). The following lemma will be used to verify our main results.

**Lemma 3.1.** [58] Assuming that \( X \) and \( Y \) are vectors or matrices with corresponding dimensions, the following equation holds:

\[
X^T X + Y^T Y \geq 2X^T Y. \tag{50}
\]

**Theorem 3.1.** Consider the MIMO nonlinear system (2) with Assumptions 2.1 and 2.2. The adaptive fuzzy robust controller is defined by Equations (26) and (27) with adaptation laws given by Equations (28)-(31). For the given positive definite matrices \( Q_1 \) and \( Q_2 \), if there exist symmetric positive definite matrices \( P_1 \) and \( P_2 \) such that the following Lyapunov equations are satisfied:

\[
(A - K_cC)^T P_1 + P_1 (A - K_cC) \leq -Q_1, \tag{51a}
\]

\[
(A - BK_c)^T P_2 + P_2 (A - BK_c) \leq -Q_2, \tag{51b}
\]

then the system (2) is asymptotically stable.

**Proof:** Consider the Lyapunov function as:

\[
V = \frac{1}{2} \tilde{e}_c^T P_1 \tilde{e}_c + \frac{1}{2} \tilde{e}_c^T P_2 \tilde{e}_c + \frac{1}{2} \gamma_f \text{tr} \left( \tilde{\theta}_f^T \tilde{\theta}_f \right) + \frac{1}{2} \gamma_g \text{tr} \left( \tilde{\theta}_g^T \tilde{\theta}_g \right)
\]

\[
+ \frac{1}{2 \gamma_c} \left( \tilde{w}_c^T \tilde{w}_c \right) + \frac{1}{2 \gamma_d} \left( \tilde{w}_c^T \tilde{w}_c \right). \tag{52}
\]

Differentiating Equation (52) with respect to time, yields:

\[
\dot{V} = \frac{1}{2} \tilde{e}_c^T P_1 \dot{\tilde{e}}_c + \frac{1}{2} \tilde{e}_c^T P_1 \dot{\tilde{e}}_c + \frac{1}{2} \tilde{e}_c^T P_2 \dot{\tilde{e}}_c + \frac{1}{2} \tilde{e}_c^T P_2 \dot{\tilde{e}}_c - \frac{1}{\gamma_f} \text{tr} \left( \tilde{\theta}_f^T \tilde{\theta}_f \right) - \frac{1}{\gamma_g} \text{tr} \left( \tilde{\theta}_g^T \tilde{\theta}_g \right)
\]

\[
- \frac{1}{\gamma_d} \text{tr} \left( \tilde{\theta}_d^T \tilde{\theta}_d \right) - \frac{1}{\gamma_c} \left( \tilde{w}_c^T \tilde{w}_c \right). \tag{53}
\]

Then, Equation (53) can be described as:

\[
\dot{V} = \frac{1}{2} \tilde{e}_c^T \left[ (A - K_cC)^T P_1 + P_1 (A - K_cC) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}_c^T \left[ (A - BK_c)^T P_2 + P_2 (A - BK_c) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}_c^T \left[ \tilde{\theta}_f^T \xi_{cf}(\hat{x}) + \tilde{\theta}_g^T \xi_{cg}(\hat{x}) u + \omega_c + u_{ca}(t) + \Delta D_c(x) \right]
\]

\[
+ \frac{1}{\gamma_f} \text{tr} \left( \tilde{\theta}_f^T \tilde{\theta}_f \right) - \frac{1}{\gamma_g} \text{tr} \left( \tilde{\theta}_g^T \tilde{\theta}_g \right)
\]

\[
- \frac{1}{\gamma_d} \text{tr} \left( \tilde{\theta}_d^T \tilde{\theta}_d \right). \tag{54}
\]

\[
\dot{V} \leq \frac{1}{2} \tilde{e}_c^T \left[ (A - K_cC)^T P_1 + P_1 (A - K_cC) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}_c^T \left[ (A - BK_c)^T P_2 + P_2 (A - BK_c) \right] \tilde{e}_c + \tilde{e}_c^T P_1 B_c u_a + \tilde{e}_c^T P_1 B_c \tilde{\theta}_f^T \xi_{cf}(\hat{x})
\]

\[
+ \tilde{e}_c^T P_1 B_c \tilde{\theta}_g^T \xi_{cg}(\hat{x}) u + \tilde{e}_c^T P_1 B_c (\omega_c + \dot{\omega}_c) + \Vert \tilde{e}_c^T P_1 B_c \Vert \cdot \Vert \Delta D_c \Vert \tag{55}
\]
Step 3: According to the design procedure, the design proceeds as follows:

\[ \dot{\mathbf{y}} = K_c \mathbf{e}_c - \frac{1}{\gamma_f} tr \left( \dot{\mathbf{f}}^T \mathbf{f} \right) - \frac{1}{\gamma_g} tr \left( \dot{\mathbf{g}}^T \mathbf{g} \right) \]

According to Lemma 3.1 and Assumption 2.2, Equation (55) can be obtained as:

\[ \dot{V} \leq \frac{1}{2} \dot{\mathbf{e}}^T_c \left( \mathbf{A} - \mathbf{K}_c \mathbf{C} \right)^T \mathbf{P}_1 + \mathbf{P}_1 \left( \mathbf{A} - \mathbf{K}_c \mathbf{C} \right) \mathbf{e}_c + \frac{1}{2} \dot{\mathbf{e}}^T_c \left( \mathbf{A} - \mathbf{B} \mathbf{K}_c \right)^T \mathbf{P}_2 + \frac{1}{2} \dot{\mathbf{e}}^T_c \mathbf{P}_2 \mathbf{B}_c \mathbf{u}_c + \dot{\mathbf{e}}^T_c \mathbf{B}_c \mathbf{Q}_1 \dot{\mathbf{e}}_f + \dot{\mathbf{e}}^T_c \mathbf{B}_c \mathbf{Q}_2 \dot{\mathbf{e}}_g \mathbf{u} \]

After some straightforward manipulations, the following equation can be obtained:

\[ \dot{V} \leq \frac{1}{2} \dot{\mathbf{e}}^T_c \left( \mathbf{A} - \mathbf{K}_c \mathbf{C} \right)^T \mathbf{P}_1 + \mathbf{P}_1 \left( \mathbf{A} - \mathbf{K}_c \mathbf{C} \right) \mathbf{e}_c + \frac{1}{2} \dot{\mathbf{e}}^T_c \left( \mathbf{A} - \mathbf{B} \mathbf{K}_c \right)^T \mathbf{P}_2 + \frac{1}{2} \dot{\mathbf{e}}^T_c \mathbf{P}_2 \mathbf{B}_c \mathbf{u}_c + \dot{\mathbf{e}}^T_c \mathbf{B}_c \left( \mathbf{w}_c + \dot{\mathbf{w}}_c \right) + \| \dot{\mathbf{e}}_c \mathbf{P}_1 \mathbf{B}_c \| \cdot \dot{\mathbf{g}}_d (\dot{\mathbf{x}}) + \| \dot{\mathbf{e}}_c \mathbf{P}_1 \mathbf{B}_c \| \cdot \dot{\mathbf{g}}_d (\dot{\mathbf{x}}) \]

Thus, \( \dot{V} < 0 \), if Equations (51a) and (51b) are satisfied. This completes the proof of the theorem.

4. Example. In the following, we apply one example to verify the effectiveness of the proposed approach. The proposed controller is designed for the tracking problem of two well-known chaotic systems. In this example, the control objective is to force the system output to track the given desired trajectory \( y_{pd} = \sin t \).

Consider the following interconnected Duffing system given by [59]:

\[
\begin{align*}
\dot{x}_{11} &= x_{12}, \\
\dot{x}_{12} &= -0.1x_{12} - x_{11}^3 + 12\cos(t) + x_{21} + x_{22} + u_1 + \Delta d_1, \\
\dot{x}_{21} &= x_{22}, \\
\dot{x}_{22} &= -0.1x_{22} - x_{21}^3 + 12\cos(t) + x_{11} + x_{12} + u_2 + \Delta d_2.
\end{align*}
\]

According to the design procedure, the design proceeds as follows:

Step 1: Specify parameters for PSO-SA: \( c_1 = 1, c_2 = 1, \chi = 0.8 \) and swarm size \( N_{\text{swarm}} = 100 \), maximum evolution generation \( Gen = 200 \) (stopping condition).

Step 2: Denote a positive definite \( Q_1 = \text{diag} [1,1,1,1], Q_2 = \text{diag} [1,1,1,1], \gamma_f = \gamma_g = 0.1, \gamma_{gd} = 0.11, r_c = 0.01 \) and \( \Delta d = \begin{bmatrix} 0.5\sin(t) \\ 0.5\sin(t) \end{bmatrix} \). Then solving the Lyapunov Equations (51a) and (51b), we can get \( P_1 \) and \( P_2 \).

Step 3: Specify the design parameters

\[
K_c = \begin{bmatrix} 145 & 45 & 0 & 0 \\ 0 & 0 & 140 & 35 \end{bmatrix}, \quad K_o = \begin{bmatrix} 45 & 100 & 0 & 0 \\ 0 & 0 & 70 & 85 \end{bmatrix}^T.
\]

Step 4: The membership functions of the fuzzy sets are described as:

\[
\begin{align*}
\mu_{F_1^1}(x_i) &= 1/1 + \exp(5 \times (x_i + 2)), \quad \mu_{F_1^2}(x_i) = \exp\left[ - (x_i + 1)^2 \right], \\
\mu_{F_2^1}(x_i) &= \exp\left[ - x_i^2 \right], \quad \mu_{F_2^2}(x_i) = \exp\left[ - (x_i - 1)^2 \right], \\
\mu_{F_3^1}(x_i) &= 1/1 + \exp(5 \times (x_i - 2)).
\end{align*}
\]
Step 5: Obtain the control law and apply it to the plant. Then, compute the adaptive law to adjust the parameter.

The initial conditions are $\mathbf{x}(0) = [2, 2, 2, 2]^T$, $\dot{\mathbf{e}}(0) = [0.1, 0.1, 0.1, 0.1]^T$, $\mathbf{d}(0) = 0$, $\theta_a(0) = 0.2$ and $\theta_{sa}(0) = 0$. Figures 1-3 show the initial membership functions for the control inputs $u$, $e$ and $\dot{e}$, where $e$ and $\dot{e}$ are mapped to the linguistic variable error and its differentiation by the fuzzification operator. In this, the two subsystems have the same initial membership functions. The fuzzy logic system inputs are composed of the five linguistic terms NB (Negative Big), NM (Negative Medium), Z (Zero), PM (Positive Medium) and PB (Positive Big). In order to compare our stabilization performance of the proposed controller with that of a conventional observer-based fuzzy adaptive robust controller (OFARC), we consider the example used in [59]. Figures 4-6 show the membership functions of $u$, $e$ and $\dot{e}$ after adjustment with the PSO-SA algorithm for $u_1$, while Figures 7-9 show membership functions of $u$, $e$ and $\dot{e}$ after adjustment with PSO-SA algorithm for $u_2$. Figures 10-13 show the response of the nonlinear system under the control law of OFARC and the proposed method with the initial conditions are $\mathbf{x}(0) = [2, 2, 2, 2]^T$, $\dot{\mathbf{e}}(0) = [0.1, 0.1, 0.1, 0.1]^T$, $\mathbf{d}(0) = 0$, $\theta_a(0) = 0.2$ and $\theta_{sa}(0) = 0$. The selection of these parameters affects the convergence rate of the adaptive parameters, settling time and overshoot. It is clear that the proposed controller can stabilize the nonlinear system and show better ability of the tracking reference signals than the OFARC. From the experiment, it is shown that the convergence times of OFARC and proposed scheme are $5.5 \sim 6.0$ seconds and $2.5 \sim 3.0$ seconds, respectively. However, the fast convergence time of proposed scheme renders its performance increased about $50\%$ over that of OFARC. The proposed control scheme can be applied to MIMO chaotic system and can achieve excellent performance for all state variables. In addition, the tracking control problems of the uncertain MIMO system have three different initial conditions $[3, 3, 3, 3]$, $[-3, -3, -3, -3]$ and $[5, -5, 5, -5]$ are considered. The simulation results are depicted in Figures 14-17, and the results show the stabilization performance of the proposed controller. One can find that the proposed scheme can achieve much better tracking performance for all state variables and is robust to external disturbance and internal uncertainty.

In order to evaluate the output control performance, we will also refer to the Integral Square Error (ISE) [60] and the Integral Absolute Error (IAE) [61] to compare with the performance of OFARC and proposed scheme, respectively.

\[
J = ISE = \int_0^\infty e^2(t) dt. \tag{59}
\]

\[
J = IAE = \int_0^\infty |e(t)| dt. \tag{60}
\]

Finally, the performance indices (tracking time, adjustment parameter of membership functions, exploit, IAE and ISE) of the simulation results are tabulated in Table 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
<th>Membership functions</th>
<th>Exploit</th>
<th>ISE</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFARC</td>
<td>$5.5 \sim 6.0$</td>
<td>Fixed</td>
<td>Affected by noise</td>
<td>2.6</td>
<td>6.5</td>
</tr>
<tr>
<td>Proposed method</td>
<td>$2.5 \sim 3.0$</td>
<td>Flexible</td>
<td>Eliminate noise</td>
<td>1.1</td>
<td>2.8</td>
</tr>
</tbody>
</table>
Figure 1. Membership functions of input variables for $u_1$ and $u_2$

Figure 2. Membership functions of $e$ for $u_1$ and $u_2$

Figure 3. Membership functions of $\dot{e}$ for $u_1$ and $u_2$

Figure 4. Membership functions after adjustment of input variables for $u_1$

Figure 5. Membership functions after adjustment of input $e$ variables for $u_1$

Figure 6. Membership functions after adjustment of input $\dot{e}$ variables for $u_1$
Figure 7. Membership functions after adjustment of input variables for $u_2$

Figure 8. Membership functions after adjustment of $e$ for $u_2$

Figure 9. Membership functions after adjustment of $_e$ for $u_2$

Figure 10. Trajectories of $x_{11}$

Figure 11. Trajectories of $x_{12}$

Figure 12. Trajectories of $x_{21}$
Figure 13. Trajectories of $x_{22}$

Figure 14. Trajectories of $x_{11}$ (various initial conditions)

Figure 15. Trajectories of $x_{12}$ (various initial conditions)

Figure 16. Trajectories of $x_{21}$ (various initial conditions)

Figure 17. Trajectories of $x_{22}$ (various initial conditions)
5. Conclusions and Future Work.

5.1. Conclusions. In this paper, an observer-based adaptive fuzzy robust controller design was proposed for a class of MIMO nonlinear systems. Compared with existing designs, the four main advantages of the proposed approach are: (1) an adaptive fuzzy robust tracking control method for a class of MIMO systems is designed; (2) the controller does not require a priori knowledge of the sign of the control coefficient; (3) a PSO-SA algorithm is used to self-adjust the controller’s coefficient for the optimal solution; and (4) a novel auxiliary compensation is introduced to eliminate external disturbance and fuzzy approximation error. The PSO-SA based adaptive fuzzy controller can generate fuzzy rules by transforming the universe of discourse and reducing the dependence on expert knowledge. Furthermore, the stabilization condition of the closed-loop is proposed using the Lyapunov theorem. Simulation results show that the proposed PSO-SA based adaptive fuzzy robust control scheme provides better tracking performance than an existing conventional method.

5.2. Future work. Since the novel auxiliary compensation in this paper can eliminate external disturbance and internal uncertainty, the proposed approach can be extended to uncertain MIMO nonlinear systems with time delays. However, there are some deficiencies when solving this kind of problem in traditional PSO-SA ways. In order to deal with the issues, we will improve the PSO-SA algorithm for these existing frameworks. In addition, the proposed method can be extended the application of the PSO-SA based adaptive fuzzy controller performance in real-world problems. Furthermore, based on the proposed PSO-SA, the algorithm can be improved to deal with more complex external disturbance and internal uncertainty of nonlinear system.

Acknowledgment. This work was supported by the National Science Council of Taiwan, under Grant NSC-100-2221-E-027-017.

REFERENCES


