

Observer-Based Adaptive FNN Control of Robot Manipulators: PSO-SA Self Adjust Membership Approach

Kai-Shiuan, Shih

Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan 701, R.O.C.
N2894115@ccmail.ncku.edu.tw

Tzue-Hseng S., Li

Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan 701, R.O.C.
thsli@mail.ncku.edu.tw

Shun-Hung, Tsai

Graduate Institute of Automation
Technology
National Taipei University of
Technology
Taipei, Taiwan, R.O.C.
shhsai@ntut.edu.tw

Abstract—In this paper, a novel observer-based adaptive fuzzy-neural network (FNN) control scheme for robotic systems is proposed for tracking performance and to suppress the effects caused by uncertainties, and disturbances. A PSO-SA based adaptive FNN system is used to approximate an unknown system from the manipulation of the model following tracking errors. The proposed scheme uses an observer, which allows for identifying the state of an unknown state in the system, simultaneously. It is shown that the proposed control scheme can guarantee the better tracking performance and suppress internal uncertainties or external disturbance. Simulations are given to show the validity and confirm the performance of the proposed scheme.

Keywords— FNN, Robust, PSO-SA, MIMO

I. INTRODUCTION

Recently, adaptive fuzzy control methods have been proposed to deal with nonlinear systems [1-5]. Based on the universal approximation theorem [1], adaptive fuzzy control schemes have been developed for a class of nonlinear uncertain systems [2]. In above schemes, the stability of the closed-loop system is established according to Lyapunov's theory. In order to process approximation errors and external disturbances, these adaptive fuzzy controllers are added by a robust compensator. However, in such schemes are based on the assumption that the states of the systems are measured directly.

Hence, the design of the observer for an uncertain nonlinear system is a very important issue. Direct and indirect adaptive fuzzy control schemes have been developed based on state or error observers [6-9]. In [8,10], the authors developed indirect and direct adaptive fuzzy controllers based on an observer for a class of single-input single-output (SISO) and multi-input multi-output (MIMO) nonlinear systems, respectively. In above schemes, the adaptive fuzzy control scheme can ensure the stability of the closed-loop system and achieve a prescribed tracking performance. However, most of adaptive fuzzy control scheme are usually operated based on human

knowledge. Thus, they pose additional difficulties to the stabilization control design and, especially, the tracking control design. In order to overcome shortcomings in the design of a controller, various methods have been developed in the last decades. In this, a successful scheme is fuzzy logic control with tuning capability in neural weights. Recently, adaptive FNN control has attracted increasing attention, essentially because it can provide a powerful learning technique for complex unknown plants to perform complex tasks in highly nonlinear dynamical environments. Furthermore, it can make available information from repetitive adjustment of the system with better performance than those of adaptive fuzzy schemes with constant rules [11,12]. However, mathematical expressions of FNN must be derived for each network layer of a network topology, and it is very difficult to find the optimum solution. In order to overcome these drawbacks, the particle swarm optimization (PSO) based method is proposed.

Particle swarm optimization (PSO) is a relatively novel evolutionary algorithm that may be used to find optimal or near optimal solutions in multi-dimensional search spaces [13]. However, for the basic PSO algorithm, all particles are prone to be trapped in the local minimum of the later phase of convergence. The optimal value found is thus often a local minimum instead of a global minimum. To overcome the shortcomings of the basic PSO algorithm, we use a method which combines the PSO algorithm (PSO) with the simulated annealing (SA) algorithm.

The rest of this paper is organized as follows. In Section 2, the system description and the design of the FNN system are presented. The adaptive FNN controller based on PSO-SA and combined with observer is described in Section 3. Numerical simulations are given and the results show that the designed observers and controllers are feasible and valid in Section 4. Finally, conclusions are given in Section 5.

II. PROBLEM STATEMENT AND PRELIMINARIES

Consider the dynamical equation of an n -link robotic manipulator with external disturbances. The class of uncertain nonlinear systems to be studied in this paper is described by the following set of time-varying differential equations:

$$\mathbf{M}(q)\ddot{q} + c(q, \dot{q}) + g(q) + \tau_d(q, \dot{q}) + \tau_c(q, \dot{q}) = u, \quad (1)$$

where $\mathbf{M}(q) \in \mathbf{R}^{n \times n}$ is the bounded positive definite inertia matrix; $c(q, \dot{q}) \in \mathbf{R}^n$ is the vector expressing centrifugal and Coriolis effects; $g(q)$ is the vector representing gravitational torques; $\tau_c(q, \dot{q}) \in \mathbf{R}^n$, $\tau_d(q, \dot{q}) \in \mathbf{R}^n$ are the vectors which represent the dynamic effects as nonlinear frictions, and bounded torque disturbances.

$u \in \mathbf{R}^n$ is the vector of joint torques; $q \in \mathbf{R}^n$ the vector of joint positions; $\dot{q} \in \mathbf{R}^n$ the vector of joint velocities; and $\ddot{q} \in \mathbf{R}^n$ is the vector of joint accelerations. Here $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbf{R}^n$, is the state vector which is assumed to be unavailable for measurement, the robot model (1) can be rewritten as:

$$\dot{\mathbf{x}} = A\mathbf{x} + B[F(q, \dot{q}) + G(q)u + \Delta d(q, \dot{q})], \quad (2)$$

where

$$\begin{aligned} F(q, \dot{q}) &= [f_1(q, \dot{q}), \dots, f_n(q, \dot{q})]^T, \quad G(q) = [G_1(q), \dots, G_n(q)], \\ G_i(q) &= [g_{i1}(q), \dots, g_{in}(q)]^T, \quad A = \text{diag}[A_1, \dots, A_n], \\ B &= \text{diag}[B_1, \dots, B_n], \quad \Delta d(q, \dot{q}) = [\Delta d_1(q, \dot{q}), \dots, \Delta d_n(q, \dot{q})]^T. \end{aligned}$$

The following assumption is made for the controller design:

Assumption 2.1. [14] The matrix $G(q)$, as previously defined, is nonsingular, i.e., $G^{-1}(q)$ exists, and bounded for all $q \in U$, where $U \in \mathbf{R}^n$ is a compact set.

Based on Assumption 2.1, if $F(q, \dot{q})$ and $G(q)$ are exact known, with $G^{-1}(q) \neq 0$ and based on the equivalent approach, then the control law can be represented as belows:

$$u = G^{-1}(q)[-F(q, \dot{q}) + y_d^{(n)} - K_c e(t) + u_a(t)], \quad (3)$$

where K_c is a control gain vector and u_a is designed for compensating the approximation errors and the eliminate disturbance.

Control Objectives:

The tracking control problem can be stated as: for given bounded reference trajectories, the control input torques such as the robot's states follow the reference trajectories, with all involved signals in the closed loop remaining bounded.

Let $y_{1d}, \dots, y_{nd}$ be the given bounded desired output signal that contains finite derivatives up to the n th order. The tracking errors of the system are found at:

$$\begin{aligned} \dot{e}_{m_i}(t) &= y_i^{(n_i)}(t) - y_{id}^{(n_i)}(t) \\ &= f_i(q, \dot{q}) + \sum_{j=1}^n g_{ij}(q)u_j - y_{id}^{(n)}(t) + \Delta d_i(q, \dot{q}), \end{aligned} \quad (4)$$

where $i = 1, \dots, n$. After some simple manipulations, we can obtain the error equation:

$$\dot{e}(t) = Ae(t) + B[F(q, \dot{q}) + G(q)u - y_d^{(n)}(t) + \Delta D(q, \dot{q})], \quad (5)$$

where $e = [e_1, \dots, e_p]^T$, $\Delta D = [\Delta d_1, \dots, \Delta d_n]^T$.

A. Fuzzy-neural network structure

The fuzzy rule base consists of a collection of fuzzy IF-THEN rules:

$$R^{(l)}: \text{If } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \dots \text{ and } x_n \text{ is } F_n^l, \quad l = 1, 2, \dots, n.$$

$$\text{then } y_s \text{ is } G_s^l, \quad s = 1, 2, \dots, p, \quad (6)$$

where $x = [x_1, \dots, x_n]^T$ is the fuzzy system input; y is the fuzzy system output; and F_n^l and G_s^l are fuzzy sets. The output of the fuzzy logic system can then be expressed as:

$$y_s = \theta_s^T \xi(x), \quad s = 1, \dots, p, \quad (7)$$

where $\theta_s^T = [\theta_s^1, \dots, \theta_s^N]$, with each variable θ_s^l being the point at which the fuzzy membership function of G_s^l achieves the maximum value; and $\xi(x) = [\xi_1(x), \dots, \xi_N(x)]^T$ with each variable ξ_l being the fuzzy basis function defined as:

$$\xi_l(x) = \prod_{i=1}^n \mu_{F_i^l}(x_i) / \sum_{l=1}^N \left(\prod_{i=1}^n \mu_{F_i^l}(x_i) \right), \quad (8)$$

where $\mu_{F_i^l}(x_i)$ is the membership function of the fuzzy set.

The inputs are then fed into the fuzzy-neural network (FNN) shown in Figure 1. The outputs (layer IV) of the fuzzy-neural system are expressed as in (9). The fuzzy system based on the neural networks can then be established [15]. The approximator has four layers. At layer I, input nodes represent vector $x^T = [x_1, \dots, x_n]$, while at layer II, they represent the values of the membership function. Each node in layer II performs a membership function value. At layer III, nodes are the values of the fuzzy basis vector ξ , with each node performing a fuzzy rule. The links between layer III and layer IV are fully connected by the weighting factors, i.e., the adjusted parameters $\theta^T = [\theta^1, \dots, \theta_p^T]$. At layer IV, the outputs represent the values of the output y .

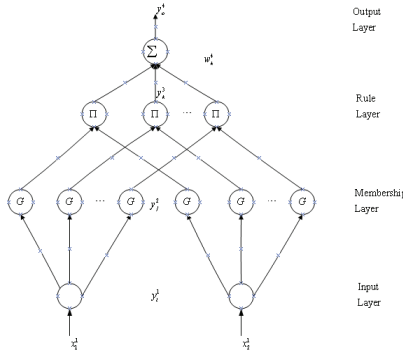


Figure 1. Structure is of a fuzzy-neural network.

Denote $\theta^T = \text{diag}[\theta_1^T, \dots, \theta_p^T]$ and $\Phi(x) = \text{diag}[\xi(x), \dots, \xi(x)]$. The FNN can be designed as:

$$y = \theta^T \Phi(x), \quad (9)$$

Let the nonlinear functions approximated by FNN be rewritten as:

$$\hat{F}((q, \dot{q})|\theta_f) = \theta_f^T \xi_f(x), \quad (10)$$

$$\hat{G}(q, \theta_g) = \theta_g^T \xi_g(x). \quad (11)$$

III. CONTROLLER DESIGN

In this section, we aim to design an observer-based adaptive fuzzy-neural network controller with PSO-SA for the robot system..

A. Basic PSO Algorithm

The concept of the PSO is the mathematical modeling and simulation of the food searching activities of a flock of birds. In multi-dimensional space, each particle moves toward the optimal point by changing its position according to a velocity. The cognitive component models the memory of the bird about its previous best position. The social component models the memory of the bird about the best position among the particles [16,17]. The mathematical model for PSO can be written as:

$$\begin{aligned} V_i(k+1) &= V_i(k) + c_1 \cdot \text{rand}_1(\cdot) \times (Pbest_i - X_i(k)) \\ &\quad + c_2 \cdot \text{rand}_2(\cdot) \times (Gbest_n - X_i(k)) \\ X_i(k+1) &= X_i(k) + V_i(k+1), \quad i = 1, \dots, N_{\text{swarm}}, \end{aligned} \quad (12)$$

where V_i is the velocity of particle i ; X_i represents the position of particle i ; $Pbest_i$ and $Gbest_n$ are the best historical position of particle i and the best historical position of the neighbor of particle i , respectively; while $\text{rand}_1(\cdot)$ and $\text{rand}_2(\cdot)$ are random numbers between 0 and 1, constants c_1 and c_2 are the weighting factors of the stochastic acceleration terms, which pull each particle towards $Pbest_i$ and $Gbest_n$. Usually $Gbest_n$ is often replaced by $Gbest_g$, which is the

global best position with an objective value. However, the method has the drawback of being easily trapped in the local minima; therefore, the following intelligence algorithm was proposed to overcome this shortcoming.

B. Simulated Annealing

The SA algorithm is introduced to prevent particles from becoming trapped in a local minimum and to increase the diversity of the particle swarm. Kirkpatrick [18] applied it to combinatorial optimization including the traveling salesman problem (TSP).

As mentioned before, in PSO, $Gbest_g$ is an element of the set of all $Pbest_i$, which can be regarded as a set of local optima. Thus, we modify the selection of $Gbest_g$ to overcome premature convergence. In particular, we employ the jumping mechanism of the SA for the selection of $Gbest_n$ instead of $Gbest_g$. $Pbest_i$ has better quality should have a larger probability of being selected. Borrowing the mechanism of the SA, all other $Pbest_i$ can be regarded as special solutions worse than $Gbest_g$. Meanwhile, we can set $e^{-(F_{P_i} - F_{G_g})/t}$ as the fitness value of each $Pbest_i$ to replace $Gbest_g$ at a certain temperature t . The largest fitness belongs to $Gbest_g$ and equals 1, with other fitness values within the interval $(0,1]$. Thus, the probability of selecting $Pbest_i$ as $Gbest_g$ can be calculated as follows:

$$\left(e^{-(F_{P_i} - F_{G_g})/t} \right) / \sum_{j=1}^{N_{\text{swarm}}} e^{-(F_{P_j} - F_{G_g})/t}, \quad (13)$$

Then, roulette wheel selection can be applied to randomly determine which $Pbest_i$ to select as $Gbest_g$.

C. Optimization of FNN with PSO-SA

For the FNN controller that has 3 membership functions (15 fuzzy sets) and 25 rules, there are a total of 60 parameters to be adjusted. Now the particles are manipulated as follows:

$$P_i = [\sigma_{11}, c_{11}, \dots, \sigma_{55}, c_{55}], \quad i = 1, 2, 3 \quad (14)$$

where parameters σ and c represent the center and deviation of the Gaussian membership functions, respectively.

The strategies of the PSO-SA are described as:

Step1: Let $k=0$, and randomly initialize $X_i(0)$ and $V_i(0)$, $i=1, \dots, N_{\text{swarm}}$. Then, evaluate objective value F_i for all particles.

Step2: Initialize $Pbest_i$ with a copy of $X_i(0)$; initialize $Gbest_g$ with a copy of the best $Pbest_i$; and initialize the temperature $t(0) = t_0$.

Step3: Repeat until the stopping criterion is satisfied.

Calculate the fitness:

$$\text{Fitness}_i = \exp\left[-(F_{p_i} - F_{p_g}) / t(k)\right], \quad (15)$$

$$\text{where } F_p = \frac{1}{N_{\text{swarm}}} \sum_{i=1}^{N_{\text{swarm}}} \sqrt{e^2(i)}. \quad (16)$$

Step4: For every particle i , apply roulette wheel selection from the set of all $Pbest_i$, and then update V_i and X_i via the following equations.

$$V_i(k+1) = \chi \times \{V_i(k) + c_1 \cdot \text{rand}_1(\cdot) \times (Pbest_i - X_i(k)) + c_2 \cdot \text{rand}_2(\cdot) \times (Gbest_n - X_i(k))\}$$

$$X_i(k+1) = X_i(k) + V_i(k+1), \quad (17)$$

where χ is called a constriction coefficient.

$$\chi = \frac{2}{2 - C - \sqrt{C^2 - 4C}}, \quad C = c_1 + c_2, \quad (18)$$

Step5: Evaluate the new objective value F_i for all particles, and then update $Pbest_i$ and $Gbest_g$ (including position values and objective values).

Step6: Set the annealing temperature to obtain $t(k+1)$, and let $k = k+1$.

Step7: The solution is then $Gbest_g$.

The biggest probability is selected as the fitness value of all $Pbest_i$ solutions as $Gbest_g$.

Remark 1. The initial temperature t_0 affects the performance of the PSO-SA. In these, the initial temperature can be written as:

$$t_0 = -F_{p_g} / \ln(0.8). \quad (19)$$

D. Observer-Based Adaptive Fuzzy-Neural Robust Controller Design

Generally, an unknown numerical can be described as follows:

$$\hat{F}((q, \dot{q})|\theta_f) = \theta_f^T \xi_f(x). \quad (20)$$

$$\hat{G}(q|\theta_g) = \theta_g^T \xi_g(x). \quad (21)$$

$$\hat{g}_d((q, \dot{q})|\theta_{g_d}) = \theta_{g_d}^T \xi_{g_d}(x). \quad (22)$$

The following technical assumption is necessary for the proof of the main statement.

Assumption 3.1: $\|\Delta D(x)\| \leq g_d(x)$, (23)

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

In short, the control law (3) can be rewritten as:

$$u = \hat{G}^{-1}(q|\theta_g)[- \hat{F}((q, \dot{q})|\theta_f) + y_d^{(n)} - K_c e(t) + u_a(t)], \quad (24)$$

where K_c is the control gain vector, and u_a is designed for compensating the approximation errors and the uncertainty.

Where

$$u_a = -\hat{w}_c - \frac{B_c^T P \tilde{e}_c}{\|\tilde{e}_c^T P_1 B_c\|} (\hat{g}_d(q, \dot{q})|\theta_{g_d}) - \frac{B_c^T P \tilde{e}_c}{\|\tilde{e}_c^T P_1 B_c\|^2} [\hat{e}^T P_2 K_o \tilde{e}_1]. \quad (25)$$

Furthermore, the parameter update laws are chosen as follows:

$$\dot{\theta}_f = \gamma_f \xi_f(\hat{x}) \tilde{e}_c^T P_1 B_c, \quad (26)$$

$$\dot{\theta}_g = \gamma_g \xi_g(\hat{x}) u \tilde{e}_c^T P_1 B_c, \quad (27)$$

$$\dot{\theta}_{g_d} = \gamma_{g_d} \|\tilde{e}_c^T P_1 B_c\| \xi_{g_d}(\hat{x}), \quad (28)$$

$$\dot{\hat{w}}_c = \gamma_c \tilde{e}_c^T P_1 B_c, \quad (29)$$

where $\gamma_f > 0$, $\gamma_g > 0$, $\gamma_{g_d} > 0$, and $\gamma_c > 0$ are positive adaptive gains to be determined.

In order to constrain the parameters θ_f , θ_g , and θ_{g_d} within the sets M_f , M_g , and M_{g_d} , respectively, the proposed adaptive laws (26)-(28) can be modified as follows:

$$\dot{\theta}_f = \begin{cases} \gamma_f \xi_f(\hat{x}) \tilde{e}_c^T P_1 B_c, & \text{if } \left(\|\theta_f\| \leq M_f \right) \\ \text{or } \left(\|\theta_f\| = M_f \right. \\ \quad \left. \text{and } \tilde{e}_c^T P_1 B_c \theta_f^T \xi_f(\hat{x}) \leq 0 \right) \\ P\{\gamma_f \xi_f(\hat{x}) \tilde{e}_c^T P_1 B_c\}, & \text{if } \left(\|\theta_f\| = M_f \right. \\ \quad \left. \text{and } \tilde{e}_c^T P_1 B_c \theta_f^T \xi_f(\hat{x}) > 0 \right) \end{cases}, \quad (30)$$

where

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

$$\dot{\theta}_g = \begin{cases} \gamma_g \xi_g(\hat{x}) u \tilde{e}_c^T P_1 B_c, & \text{if } \left(\|\theta_g\| \leq M_g \right) \\ \text{or } \left(\|\theta_g\| = M_g \right. \\ \quad \left. \text{and } \tilde{e}_c^T P_1 B_c \theta_g^T \xi_g(\hat{x}) u \leq 0 \right) \\ P\{\gamma_g \xi_g(\hat{x}) u \tilde{e}_c^T P_1 B_c\}, & \text{if } \left(\|\theta_g\| = M_g \right. \\ \quad \left. \text{and } \tilde{e}_c^T P_1 B_c \theta_g^T \xi_g(\hat{x}) u > 0 \right) \end{cases}, \quad (31)$$

where

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

$$\dot{\theta}_{g_d} = \begin{cases} \gamma_{g_d} \|\tilde{e}_c^T P_1 B_c\| \xi_{g_d}(\hat{x}), & \text{if } (\|\theta_{g_d}\| \leq M_{\theta_{g_d}}) \\ \text{or } \begin{pmatrix} \|\theta_{g_d}\| = M_{\theta_{g_d}} \\ \text{and } \|\tilde{e}_c^T P_1 B_c\| \cdot \theta_{g_d}^T \xi_{g_d}(\hat{x}) \leq 0 \end{pmatrix} \\ P\{\|\tilde{e}_c^T P_1 B_c\| \gamma_{g_d} \xi_{g_d}(\hat{x})\}, & \text{if } \begin{pmatrix} \|\theta_{g_d}\| = M_{\theta_{g_d}} \\ \text{and } \|\tilde{e}_c^T P_1 B_c\| \theta_{g_d}^T \xi_{g_d}(\hat{x}) u > 0 \end{pmatrix} \end{cases} \quad (32)$$

where

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

In this, the control input can be arranged as:

$$u = \hat{G}^{-1}(q|\theta_g) [-\hat{F}((q, \dot{q})|\theta_f) + y_d^{(n)} - K_c \hat{e}(t) + u_a(t)]. \quad (33)$$

By (6), (33) can be arranged:

$$\dot{e}(t) = Ae(t) - BK_c \hat{e}(t) + B\{F((q, \dot{q})) - \hat{F}((q, \dot{q})|\theta_f) + [G(q) - \hat{G}(q|\theta_g)]u(t) + u_a(t) + \Delta D((q, \dot{q}))\}. \quad (34)$$

The output tracking error vector is redefined as follows:

$$\dot{\tilde{e}}(t) = (A - BK_c) \tilde{e}(t) + K_o(e(t) - \hat{e}(t)), \quad (35)$$

where K_o is the observer gain vector.

Define the observation error vector as:

$$\tilde{e}(t) = e(t) - \hat{e}(t), \quad (36)$$

In short, one can obtain:

$$\dot{\tilde{e}} = (A - K_o C) \tilde{e} + B\{[F(q, \dot{q}) - \hat{F}((q, \dot{q})|\theta_f)] + [G(q) - \hat{G}(q|\theta_g)]u(t) + u_a(t) + \Delta D((q, \dot{q}))\}, \quad (37)$$

Define the optimal parameter vectors θ_f^* , θ_g^* , and $\theta_{g_d}^*$ as:

$$\theta_f^* = \arg \min_{\theta_f \in \Omega_f} \left\{ \sup_{x \in U_x, \dot{x} \in U_{\dot{x}}} |F(q, \dot{q}) - \hat{F}((q, \dot{q})|\theta_f)| \right\}, \quad (38)$$

$$\theta_g^* = \arg \min_{\theta_g \in \Omega_g} \left\{ \sup_{x \in U_x, \dot{x} \in U_{\dot{x}}} |G(q) - \hat{G}(q|\theta_g)| \right\}, \quad (39)$$

$$\theta_{g_d}^* = \arg \min_{\theta_{g_d} \in \Omega_{g_d}} \left\{ \sup_{x \in U_x, \dot{x} \in U_{\dot{x}}} |g_d((q, \dot{q})) - \hat{g}_d((q, \dot{q})|\theta_{g_d})| \right\}, \quad (40)$$

where $\tilde{\theta}_f = \theta_f^* - \theta_f$, $\tilde{\theta}_g = \theta_g^* - \theta_g$, and $\tilde{\theta}_{g_d} = \theta_{g_d}^* - \theta_{g_d}$ are parameter estimation errors.

The errors can be defined as:

$$w = F(q, \dot{q}) - \hat{F}((q, \dot{q})|\theta_f^*) + (G(q) - \hat{G}(q|\theta_g^*))u + g_d(q, \dot{q}) - \hat{g}_d((q, \dot{q})|\theta_{g_d}^*). \quad (41)$$

The linear transfer function $H(s)$ is realized using standard techniques by $H(s) = C(sI - (A - K_o C))^{-1} B$, and it

used SPR-Lyapunov design approach, and after some simple manipulations, we can obtain the equation:

$$\tilde{e}_q(t) = H(s)L(s) \left[L(s)^{-1} \tilde{\theta}_f^T \xi_f(\hat{x}) + L(s)^{-1} \tilde{\theta}_g^T \xi_g(\hat{x}) u(t) + L(s)^{-1} (w + u_a(t) + \Delta D(q, \dot{q})) \right], \quad (42)$$

where $L(s) = [L_1(s), \dots, L_n(s)]$ and

$$L_i(s) = s^{m_i} + b_{i1}s^{m_i-1} + \dots + b_{im_i}, \quad (m_i = n_i - 1), \quad i = 1, 2, \dots, p.$$

Hence, the state-space realization can be rewritten as follows:

$$\dot{\tilde{e}}_c = (A - K_o C) \tilde{e}_c + B_c [\tilde{\theta}_f^T \xi_{cf}(\hat{x}) + \tilde{\theta}_g^T \xi_{cg}(\hat{x}) u(t) + w_c + u_{ca}(t) + \Delta D_c(q, \dot{q})], \quad (43)$$

where $B_c = \text{diag}[B_{c1}, \dots, B_{cp}]$, $B_{ci} = \text{diag}[1, b_{i1}, \dots, b_{im_i}]$,

$$\xi_{cf}(\hat{x}) = L(s)^{-1} \xi_f(\hat{x}), \quad w_c = L(s)^{-1} w,$$

$$\Delta D_c(q, \dot{q}) = L(s)^{-1} \Delta D(q, \dot{q}), \text{ and } \xi_{cg}(\hat{x}) = L(s)^{-1} \xi_g(\hat{x}).$$

Lemma 3.1 [14] Assuming that X and Y are vectors or matrices with corresponding dimensions, the following equation holds:

$$X^T X + Y^T Y \geq 2X^T Y. \quad (44)$$

Theorem 3.1: For the given positive definite matrices Q_1 and Q_2 , the augmented control system (2) containing the observer (35) is asymptotically stabilized via the controller (24) if there exist symmetric positive definite matrices P_1 and P_2 such that the following Lyapunov equations are satisfied.

$$\begin{aligned} (A - K_o C)^T P_1 + P_1 (A - K_o C) &\leq -Q_1, \\ (A - BK_c)^T P_2 + P_2 (A - BK_c) &\leq -Q_2. \end{aligned} \quad (45)$$

Proof of Theorem 3.1

Consider the Lyapunov function as:

$$V = \frac{1}{2} \tilde{e}_c^T P_1 \tilde{e}_c + \frac{1}{2} \tilde{e}^T P_2 \tilde{e} + \frac{1}{2\gamma_f} \text{tr}(\tilde{\theta}_f^T \tilde{\theta}_f) + \frac{1}{2\gamma_g} \text{tr}(\tilde{\theta}_g^T \tilde{\theta}_g) + \frac{1}{2\gamma_c} (\tilde{w}^T \tilde{w}) + \frac{1}{2\gamma_{g_d}} \text{tr}(\tilde{\theta}_{g_d}^T \tilde{\theta}_{g_d}), \quad (46)$$

Then, the time derivative of the Lyapunov (46) is:

$$\dot{V} = \frac{1}{2} \tilde{e}_c^T P_1 \dot{\tilde{e}} + \frac{1}{2} \tilde{e}^T P_1 \dot{\tilde{e}} + \frac{1}{2} \dot{\tilde{e}}^T P_2 \tilde{e} + \frac{1}{2} \tilde{e}^T P_2 \dot{\tilde{e}} - \frac{1}{\gamma_f} \text{tr}(\tilde{\theta}_f^T \dot{\tilde{\theta}}_f) - \frac{1}{\gamma_g} \text{tr}(\tilde{\theta}_g^T \dot{\tilde{\theta}}_g) - \frac{1}{\gamma_c} (\tilde{w}^T \dot{\tilde{w}}) - \frac{1}{\gamma_{g_d}} \text{tr}(\tilde{\theta}_{g_d}^T \dot{\tilde{\theta}}_{g_d}). \quad (47)$$

After some simple manipulations, we can obtain as:

$$\begin{aligned} \dot{V} \leq & \frac{1}{2} \tilde{e}_c^T \left[(A - K_o C)^T P_1 + P_1 (A - K_o C) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}^T \left[(A - BK_c)^T P_2 \right. \\ & \left. + P_2 (A - BK_c) \right] \tilde{e} + \tilde{e}_c^T P_1 B_c u_a + \tilde{e}_c^T P_1 B_c \tilde{\theta}_f^T \xi_f(\hat{x}) \\ & + \tilde{e}_c^T P_1 B_c \tilde{\theta}_g^T \xi_g(\hat{x}) u + \tilde{e}_c^T P_1 B_c (\tilde{w} + \hat{w}) + \left\| \tilde{e}_c^T P_1 B_c \right\| \cdot \left\| \Delta D_f \right\| \\ & + \tilde{e}^T P_2 K_o \tilde{e}_1 - \frac{1}{\gamma_f} \text{tr}(\tilde{\theta}_f^T \dot{\tilde{\theta}}_f) - \frac{1}{\gamma_g} \text{tr}(\tilde{\theta}_g^T \dot{\tilde{\theta}}_g) - \frac{1}{\gamma_c} (\tilde{w}^T \dot{\tilde{w}}) \\ & - \frac{1}{\gamma_{g_d}} \text{tr}(\tilde{\theta}_{g_d}^T \dot{\tilde{\theta}}_{g_d}). \end{aligned} \quad (48)$$

By using Lemma 3.1 and Assumption 2.1-3.1, we further have:

$$\begin{aligned} \dot{V} \leq & \frac{1}{2} \tilde{e}_c^T \left[(A - K_o C)^T P_1 + P_1 (A - K_o C) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}^T \left[(A - BK_c)^T P_2 \right. \\ & \left. + P_2 (A - BK_c) \right] \tilde{e} + \tilde{e}_c^T P_1 B_c u_a + \tilde{e}_c^T P_1 B_c \tilde{\theta}_f^T \xi_f(\hat{x}) \\ & + \tilde{e}_c^T P_1 B_c \tilde{\theta}_g^T \xi_g(\hat{x}) u + \tilde{e}_c^T P_1 B_c (\tilde{w} + \hat{w}) + \left\| \tilde{e}_c^T P_1 B_c \right\| \\ & \times \hat{g}_d((q, \dot{q}) | \theta_{g_d}) + \left\| \tilde{e}_c^T P_1 B_c \right\| \cdot \tilde{\theta}_{g_d}^T \xi_{g_d}(\hat{x}) + \tilde{e}^T P_2 K_o \tilde{e}_1 \\ & - \frac{1}{\gamma_f} \text{tr}(\tilde{\theta}_f^T \dot{\tilde{\theta}}_f) - \frac{1}{\gamma_g} \text{tr}(\tilde{\theta}_g^T \dot{\tilde{\theta}}_g) - \frac{1}{\gamma_c} (\tilde{w}^T \dot{\tilde{w}}) \\ & - \frac{1}{\gamma_{g_d}} \text{tr}(\tilde{\theta}_{g_d}^T \dot{\tilde{\theta}}_{g_d}). \end{aligned} \quad (49)$$

Then, (49) can be written as:

$$\begin{aligned} \dot{V} \leq & \frac{1}{2} \tilde{e}_c^T \left[(A - K_o C)^T P_1 + P_1 (A - K_o C) \right] \tilde{e}_c + \frac{1}{2} \tilde{e}^T \left[(A - BK_c)^T P_2 \right. \\ & \left. + P_2 (A - BK_c) \right] \tilde{e}. \end{aligned} \quad (50)$$

Finally:

$$\dot{V} \leq -\frac{1}{2} \tilde{e}_c^T Q_1 \tilde{e}_c - \frac{1}{2} \tilde{e}^T Q_2 \tilde{e}.$$

Q.E.D

IV. SIMULATION

This section presents the simulation results of the proposed observer-based adaptive fuzzy-neural network robust controller with PSO-SA for robot systems to illustrate that the stability for the closed-loop system. In order to demonstrate the effectiveness of the proposed approach, we apply it to a two-link plane robot system.

Example:

$$\mathbf{M}(q)\ddot{q} + c(q, \dot{q})\dot{q} + g(q) + \tau_d(q, \dot{q}) + \tau_c(q, \dot{q}) = u, \quad (51)$$

where

$$\mathbf{M} = \begin{bmatrix} (m_1 + m_2)l_1^2 & m_2 l_1 l_2 (s_1 s_2 + c_1 c_2) \\ m_2 l_1 l_2 (s_1 s_2 + c_1 c_2) & m_2 l_2^2 \end{bmatrix},$$

$$c = \begin{bmatrix} -m_2 l_1 l_2 (c_1 s_2 - s_1 c_2) \dot{q}_2^2 \\ -m_2 l_1 l_2 (c_1 s_2 - s_1 c_2) \dot{q}_1^2 \end{bmatrix}, \quad g = \begin{bmatrix} -g(m_1 + m_2) s_1 l_1 \\ -g m_2 l_2 s_2 \end{bmatrix}.$$

In the simulation, the parameters are chosen as:

$$\begin{aligned} c_1 &= \cos(q_1), \quad c_2 = \cos(q_2), \quad s_1 = \sin(q_1), \quad \text{and} \\ s_2 &= \sin(q_2). \text{ The robot physical parameters are: } l_1 = l_2 = 1\text{m}, \\ m_1 &= m_2 = 1\text{kg}, \quad g = 9.8\text{m/s}^2. \\ \Delta d(q, \dot{q}) &= \begin{bmatrix} \sin(2q_1) + 2\text{sign}(\dot{q}_1) \\ \sin(q_2) + \text{sign}(\dot{q}_2) \end{bmatrix}. \end{aligned}$$

The design algorithm is as follows:

Step1: Let parameters for PSO-SA at $c_1 = 1$, $c_2 = 1$, and $N_{\text{swarm}} = 100$.

Step2: Specify a positive definite $\mathbf{Q}_1 = \text{diag}[1, 1, 1, 1]$, and $\mathbf{Q}_2 = \text{diag}[1, 1, 1, 1]$ $\gamma_f = \gamma_g = 0.1$, $\gamma_{g_d} = 0.1$, $r_c = 0.01$,

Step3: Specify the design parameters $K_c = \begin{bmatrix} 4 & 4 & 0 & 0 \\ 0 & 0 & 4 & 4 \end{bmatrix}$,

$$K_o = \begin{bmatrix} 90 & 185 & 0 & 0 \\ 0 & 0 & 90 & 185 \end{bmatrix}^T.$$

Step4: The membership functions of the fuzzy controller are given by:

$$\mu_{F_1^1}(x_i) = 1/1 + \exp(5 \times (x_i + 2)),$$

$$\mu_{F_1^2}(x_i) = \exp[-(x_i + 1)^2],$$

$$\mu_{F_1^3}(x_i) = \exp[-x_i^2],$$

$$\mu_{F_1^4}(x_i) = \exp[-(x_i - 1)^2],$$

$$\mu_{F_1^5}(x_i) = 1/1 + \exp(5 \times (x_i - 2)).$$

Step5: Obtain the control law.

The initial conditions are $\mathbf{x}(0)=[2,2,2,2]^T$, $\theta_f(0)=0$, $\theta_g(0)=0.2$, and $\theta_{gd}(0)=0$. The initial membership functions for u are depicted as Figure 2 and Figure 3. Then, Figure 4 and Figure 5 show membership functions of u after self-adjustment with PSO-SA for u_1 and u_2 . In order to compare the stabilization and tracking performance of the proposed controller with that of a conventional observer-based fuzzy adaptive robust controller, we consider the example [19]. Figures 6-9 show the response of the system under the traditional control law and the proposed method for the same initial conditions. It is clear that the proposed controllers can stabilize the nonlinear system and show faster tracking responses than traditional control law.

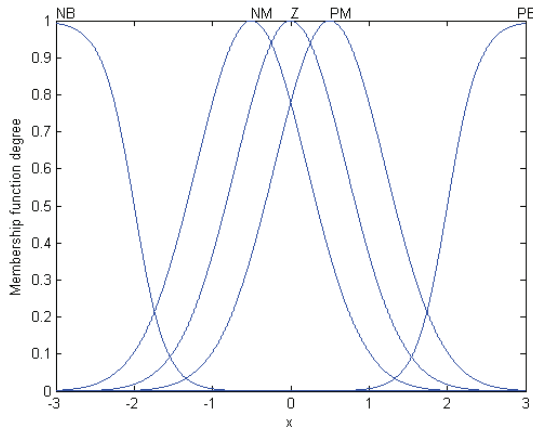


Figure 2. Membership functions of u_1 .

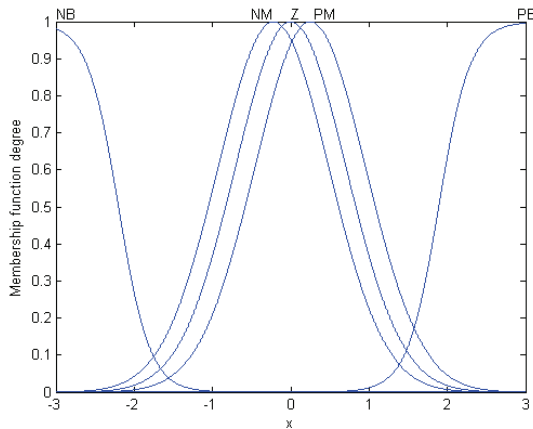


Figure 3. Membership functions of u_2 .

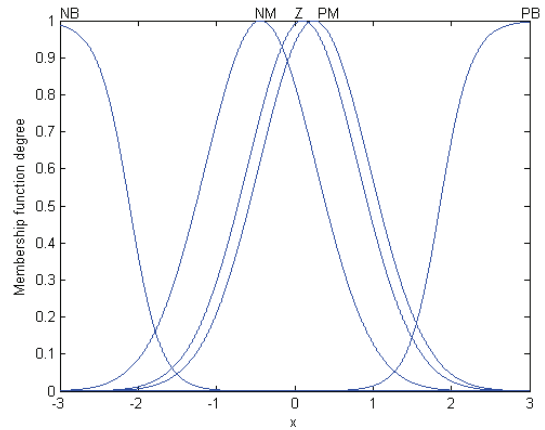


Figure 4. Membership functions after training of input variables for u_1 .

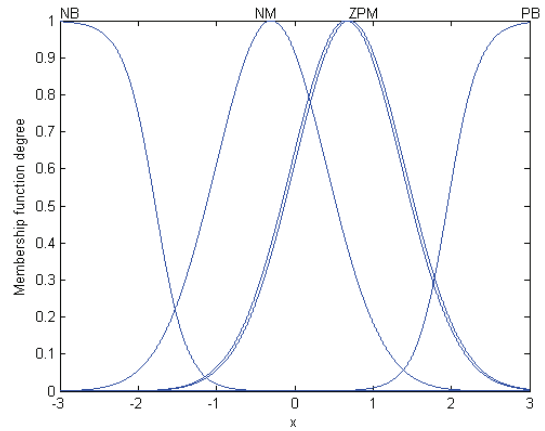


Figure 5. Membership functions after training of input variables for u_2 .

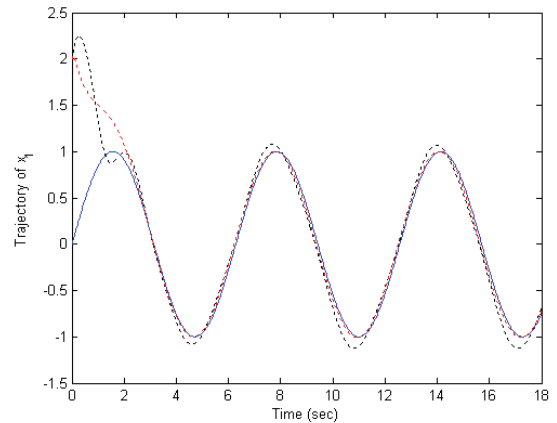


Figure 6. Trajectories of q_1 . (Red: Proposed controller; Black: Traditional control [19])

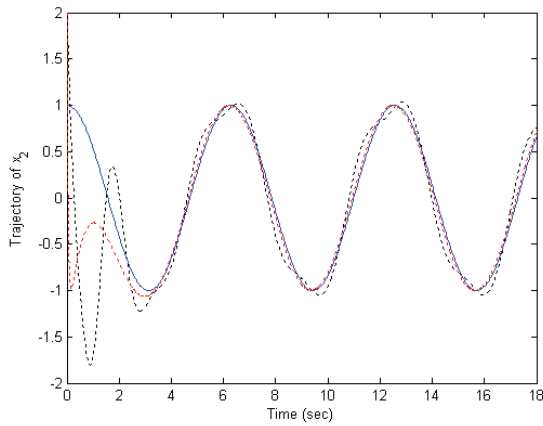


Figure 7. Trajectories of $\dot{q}_1$. (Red: Proposed controller; Black: Traditional control [19])

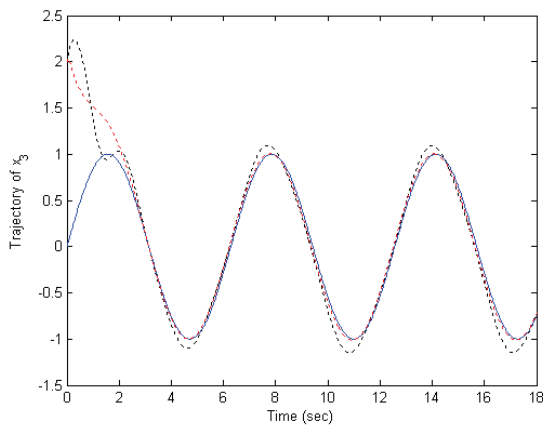


Figure 8. Trajectories of q_2 . (Red: Proposed controller; Black: Traditional control [19])

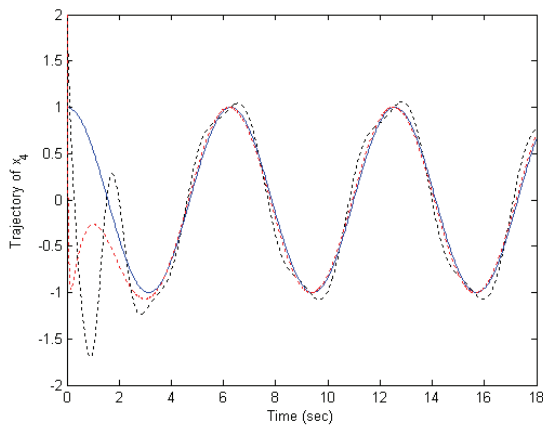


Figure 9. Trajectories of $\dot{q}_3$. (Red: Proposed controller; Black: Traditional control [19])

V. CONCLUSION

An observer-based adaptive FNN controller design was proposed for robot systems. The main contributions of this paper are as follows: Firstly, a robust term is introduced to eliminate disturbance and the controller doesn't require a priori knowledge of the sign of the control coefficient. Secondly, PSO-SA is used to self-adjust membership function

for the sub-optimal solution. Thirdly, the PSO-SA based adaptive FNN controller can generate a reduction in the dependence on expert knowledge, while the stabilization condition of the closed-loop was proposed using the Lyapunov theorem. Finally, stability conditions can be efficiently checked using dedicated optimization software.

REFERENCES

- [1] X. Wang, S. Li, and W. Cai, "Multi-model direct adaptive decoupling control with application to the wind tunnel system," *ISA Transactions*, vol. 44, pp. 131-143, 2005.
- [2] S. Alavandar and M. J. Nigam, "New hybrid adaptive neuro-fuzzy algorithms for manipulator control with uncertainties-Comparative study," *ISA Transactions*, vol. 48, pp. 497-502, 2009.
- [3] T. Bouarar, K. Guelton, and N. Manamanni, "Robust fuzzy Lyapunov stabilization for uncertain and disturbed Takagi-Sugeno descriptors," *ISA Transactions*, vol. 49, pp. 447-461, 2010.
- [4] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, NJ: Englewood Cliffs, 1994.
- [5] J. Yoneyama, "Design of H^∞ control for fuzzy time-delay systems," *Fuzzy Sets and Systems*, vol. 19, pp. 224-256, 2004.
- [6] Y. Zhao, E. G. C. Jr, and D. A. Cartes, "Self-tuning adaptive control for an industrial weigh belt feeder," *ISA Transactions*, vol. 42, pp. 437-350, 2003.
- [7] J. W. Macki, P. Nistri, and P. Zecca, "Mathematical models for hysteresis," *SIAM Review*, vol. 35, pp. 94-123, 1993.
- [8] I. D. Mayergoyz, *The Preisach Model for Hysteresis*, Springer-Verlag. Berlin: Germany, 1991.
- [9] N. J. Ahmad and F. Khorrami, "Adaptive control of systems with backlash hysteresis at the input," *Proceedings of American Control Conference*, vol. 1, pp. 3018-3022, 1999.
- [10] J. Zhou, C. Zhang, and C. Wen, "Robust adaptive output control of uncertain nonlinear plants with unknown backlash nonlinearity," *IEEE Transactions on Automatic Control*, vol. 52, pp. 503-509, 2007.
- [11] T. P. Zhang and S. S. Ge, "Adaptive neural control of MIMO nonlinear state time-varying delay systems with unknown dead-zones and gain signs," *Automatica*, vol. 43, pp. 1021-1033, 2007.
- [12] Y. J. Liu and N. Zhou, "Observer-based adaptive fuzzy-neural control for a class of uncertain nonlinear systems with unknown dead-zone input," *ISA Transactions*, vol. 49, pp. 462-469, 2010.
- [13] Kennedy and J., Eberhart, R. (1995), 'Particle swarm optimization', In: *Proc. IEEE Internat. Conf. on Neural Networks*, pp. 1942-1948.
- [14] P. A. Ioannou, J. Sun, *Robust Adaptive Control*, Englewood Cliffs, NJ: 1996.
- [15] M. J. Er and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE Transactions on Fuzzy System*, vol. 9, pp. 578-594, 2001.
- [16] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Trans. Power Systems*, vol.18, pp.60-68, 2003.
- [17] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Inform. Process. Lett.*, vol.85, pp.317-325, 2003.
- [18] Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol.220, pp.671-680, 1983.
- [19] N. Gole'a, A. Gole'a, K. Barra, and T. Bouktir, "Observer-based adaptive control of robot manipulators: Fuzzy systems approach," *Applied Soft Computing* 8 778-787, 2008.