RESEARCH ARTICLE OPEN ACCESS

Adaptive Fuzzy-Neural Control Utilizing Sliding Mode Based Learning Algorithm for Robot Manipulator

Thuy Van Tran*, YaoNan Wang**
* (College of Electrical and Information Engineering, Hunan University, China;
Faculty of Electrical Engineering, Hanoi University of Industry, Vietnam;
Email: tranthuyvan.haui@gmail.com)
** (College of Electrical and Information Engineering, Hunan University, China;
Email: yaonan@hnu.edu.cn)

ABSTRACT
This paper introduces an adaptive fuzzy-neural control (AFNC) utilizing sliding mode-based learning algorithm (SMBLA) for robot manipulator to track the desired trajectory. A traditional sliding mode controller is applied to ensure the asymptotic stability of the system, and the fuzzy rule-based wavelet neural networks (FWNNs) are employed as the feedback controllers. Additionally, a novel adaptation of the FWNNs parameters is derived from the SMBLA in the Lyapunov stability theorem. Hence, the AFNC approximates parameter variation, unmodeled dynamics, and unknown disturbances without the detailed knowledge of robot manipulator, while resulting in an improved tracking performance. Lastly, in order to validate the effectiveness of the proposed approach, the comparative simulation results of two-degrees of freedom robot manipulator are presented.

Keywords – traditional sliding mode control (TSMC), adaptive fuzzy neural control (AFNC), fuzzy rule-based wavelet neural network (FWNN), sliding mode-based learning algorithm (SMBLA), degrees of freedom robot manipulator (DOFRM)

I. INTRODUCTION

Generally, various uncertainties comprising parameter variation, unmodeled dynamics, and unknown disturbances influence the tracking performances of robot manipulator [1, 2]. In the designing of reference model based control system, it is difficult for determining a mathematical model correctly. Because the traditional controllers (i.e., robust controller [3], sliding mode controller [4]) are time-invariant controllers, this term causes nonlinearities and discontinuities which renders traditional control invalid. So the requirement of the intelligent control approaches (ICAs) is that reducing the impact of the various uncertainties in the design process. During the last decades, the ICAs (i.e., neural network control (NNC) [5], and fuzzy logic control (FLC) [6]) have been largely applied for controlling the motion of robot manipulators [7, 8]. The topical trend of researches is that integrating the traditional control methods with the ICAs for the improvement in the performance of system [9-11]. Besides, based on the combination of the rule reasoning of fuzzy systems and the learning capability of neural networks without the prior knowledge, the fuzzy-neural network control (FNNC) methods are also designed to provide higher robustness than both NNC and FLC [12-14].

In the training of artificial neural networks (ANNs) and fuzzy-neural networks (FNNs), different learning algorithms containing gradient descent-based algorithm (GDBA) [15] and evolutionary computation-based algorithm (ECBA) [16, 17] have been utilized. However, the convergence rate of GDBA is sluggish due to the involvement of partial derivatives, specifically when the solution space is complicated. For the ECBA, the stability and optimal values are difficultly reached by using stochastic operators, and the high calculation is still a burden. It is well-known that sliding mode control (SMC) is a method which can ensure the stability and robustness in both the case of uncertainties and computationally intelligent systems [18]. By using the SMC strategy in the online learning for ANNs and FNNs, sliding mode-based learning algorithm (SMBLA) can guarantee better convergence and more robust than conventional learning approaches [19, 20]. It is different from GDBA in feedback-error learning [21], the network parameters are updated by SMBLA in the way that the learning error is enforcedly satisfied a stable equation.

In this paper, an adaptive fuzzy-neural control (AFNC) using SMBLA is proposed for tracking desired trajectory of robot manipulator. In the proposed control method, the traditional sliding mode controller (TSMC) is applied for guaranteeing the asymptotic stability of the control system, and the fuzzy rule-based wavelet neural networks

(FWNNs) are employed as feedback controllers to approximate the uncertainties. Moreover, a novel SMBLA strategy is suggested to train the FWNNs using wavelet basis membership function (WBMF) [22]. By using Lyapunov theorem to prove the stability of the SMBLA, the fast convergence ability of the FWNNs parameters is ensured, and an adaptive updating law is achieved. Hence, the proposed method approximates the uncertainties without the detailed knowledge of robot manipulator, while resulting in an improved performance. Last of all, the comparative simulation results of two-degrees of freedom (DOF) robot manipulator are presented for validating the effectiveness of the proposed AFNC system.

The remainder of the paper is organized as follows: section 2 represents the preliminaries. In section 3, the AFNC scheme and the SMBLA are presented. Section 4 provides the comparative simulation results of two-DOF robot manipulator. Finally, the conclusion is shown in section 5.

II. PRELIMINARIES

1. Dynamic Model of Robot Manipulator

Consider an n-DOF robot manipulator, the dynamics can be represented in Lagrange formation [23]:

$$\boldsymbol{M}_r(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{V}_r(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{g}_r(\boldsymbol{\theta}) + \boldsymbol{\eta}_e = \boldsymbol{u}_\tau \quad (1)$$

where $\boldsymbol{M}_r(\boldsymbol{\theta}) \in R^{n\times n}$ is the inertial matrix, $\boldsymbol{V}_r(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \in R^{n\times n}$ is the Coriolis-centripetal matrix, $\boldsymbol{g}_r(\boldsymbol{\theta}) \in R^n$ is the gravity vector, $\boldsymbol{\eta}_e \in R^n$ is the vector of unknown disturbances, $\boldsymbol{u}_\tau \in R^n$ is the vector of control torques, and $\boldsymbol{\theta}(t) \in R^n$, $\dot{\boldsymbol{\theta}}(t)$, and $\ddot{\boldsymbol{\theta}}(t)$ are the vectors of joint positions, corresponding velocities, and corresponding accelerations, respectively.

2. FWNN Structure and Fuzzy If-Then Rule

The structure of a five-layer FWNN, as depicted in Figure 1, contains two input neurons, $(p+q)$ membership neurons, $(p\times q)$ rule neurons, $(p\times q)$ normalization neurons, and one output neuron.

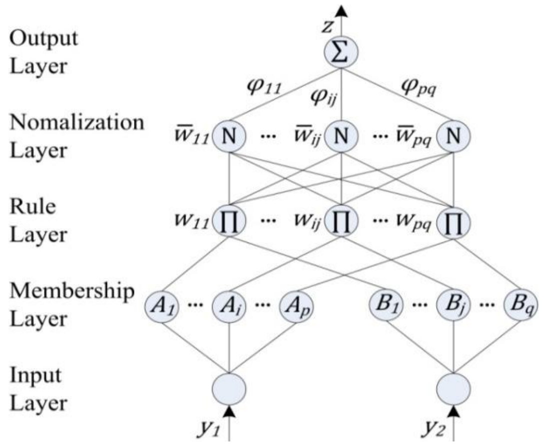

Figure 1: Structure of FWNN

Consider a zeroth-order Takagi-Sugeno-Kang model containing two input variables, the fuzzy If-Then rules is described as follows:

$$r_{ij}: \quad If\ y_1\ is\ A_i\ and\ y_2\ is\ B_j, \quad Then\ \varphi_{ij} = d_{ij} \quad (2)$$

where y_1 and y_2 are the input variables of FWNN, φ_{ij} is a zeroth-order function in the consequent element of the rule r_{ij}, and A_i and B_j denote the fuzzy sets of y_1 and y_2, respectively.

Input Layer (Layer 1): Given a input vector of two crisp variables $\boldsymbol{y} = [y_1, y_2]^T \in R^2$, their values are transmitted to the next layer by the neurons in this layer.

Membership Layer (Layer 2): By using WBMF, the membership neurons map y_1 and y_2 into fuzzified values. These membership neurons have the WBMFs represented by:

$$\begin{cases} \mu_{A_i}(y_1) = \left(1 - \left\{\delta_{A_i}(y_1 - \alpha_{A_i})\right\}^2\right) e^{-\left\{\delta_{A_i}(y_1-\alpha_{A_i})\right\}^2} \\ \mu_{B_j}(y_2) = \left(1 - \left\{\delta_{B_j}\left(y_2 - \alpha_{B_j}\right)\right\}^2\right) e^{-\left\{\delta_{B_j}(y_2-\alpha_{B_j})\right\}^2} \end{cases} \quad (3)$$

where $\mu_{A_i}(y_1)$ and $\mu_{B_j}(y_2)$ are the membership values, α_{A_i} and α_{B_j} are the translation parameters, and δ_{A_i} and δ_{B_j} are the dilation parameters of WBMF for input variables y_1 and y_2, respectively. $i = 1,2,\dots,p$ and $j = 1,2,\dots,q$.

Rule Layer (Layer 3): The output of each rule neuron expresses a firing strength w_{ij} of corresponding rule, and it is calculated by multiplying two incoming signals:

$$w_{ij} = \mu_{A_i}(y_1)\mu_{B_j}(y_2) \quad (4)$$

Normalization Layer (Layer 4): In this layer, the normalization of all of the firing strengths is performed. Then, the normalized value of every neuron can be denoted as:

$$\bar{w}_{ij} = \frac{w_{ij}}{\sum_{i=1}^{p}\sum_{j=1}^{q} w_{ij}} \quad (5)$$

Output Layer (Layer 5): The defuzzification is performed in this layer. The output linguistic variable is computed according to the weighted sum technique of all incoming signals:

$$z = \sum_{i=1}^{p}\sum_{j=1}^{q}\left(\bar{w}_{ij}\,\varphi_{ij}\right) = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q}(w_{ij}\,d_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{q} w_{ij}} \quad (6)$$

III. DESIGN OF AFNC USING SMBLA

1. AFNC Scheme

An AFNC scheme, as illustrated in Figure 2, presents a combination of the sliding mode controller in parallel with the FWNNs.

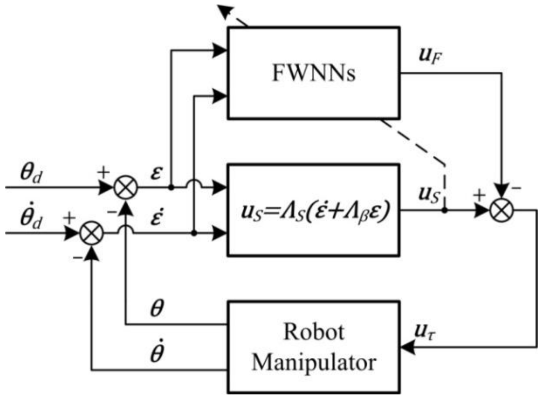

Figure 2: Structure of AFNC system

The first, the sliding mode controller is designed for ensuring the asymptotic stability of the control system. A sliding surface $\boldsymbol{\beta}_\varepsilon$ is specified by:

$$\boldsymbol{\beta}_\varepsilon(\boldsymbol{\varepsilon},\dot{\boldsymbol{\varepsilon}}) = \dot{\boldsymbol{\varepsilon}} + \boldsymbol{\Lambda}_\beta \boldsymbol{\varepsilon} = \dot{\boldsymbol{\theta}}_d - \dot{\boldsymbol{\theta}} + \boldsymbol{\Lambda}_\beta(\boldsymbol{\theta}_d - \boldsymbol{\theta}) = [\beta_\varepsilon^1, \dots \quad \beta_\varepsilon^k, \dots \quad \beta_\varepsilon^n]^T \tag{7}$$

where $\boldsymbol{\Lambda}_\beta$ is a diagonal and positive definite constant matrix defining the sliding surface slope, $k = 1,2,\dots n$, and the vectors of desired positions, desired velocities, feedback position errors, and feedback velocity errors are denoted by $\boldsymbol{\theta}_d = [\theta_d^1, \dots \quad \theta_d^k, \dots \quad \theta_d^n]^T$, $\dot{\boldsymbol{\theta}}_d = [\dot{\theta}_d^1, \dots \quad \dot{\theta}_d^k, \dots \quad \dot{\theta}_d^n]^T$, $\boldsymbol{\varepsilon} = [\varepsilon^1, \dots \quad \varepsilon^k, \dots \quad \varepsilon^n]^T$, and $\dot{\boldsymbol{\varepsilon}} = [\dot{\varepsilon}^1, \dots \quad \dot{\varepsilon}^k, \dots \quad \dot{\varepsilon}^n]^T$, respectively. Then, the sliding control law is defined as follows:

$$\boldsymbol{u}_S = [u_S^1, \dots \quad u_S^k, \dots \quad u_S^n]^T = \boldsymbol{\Lambda}_S \boldsymbol{\beta}_\varepsilon = \begin{bmatrix} \lambda_S^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_S^n \end{bmatrix} [\beta_\varepsilon^1, \dots \quad \beta_\varepsilon^k, \dots \quad \beta_\varepsilon^n]^T \tag{8}$$

where $\boldsymbol{\Lambda}_S$ is a diagonal and positive definite gain matrix, with $\lambda_S^k > 0$.

The second, the FWNNs are used as the feedback controllers to approximate the uncertainties in the system. For the k^{th} FWNN, the two inputs y_1^k and y_2^k are considered as ε^k and $\dot{\varepsilon}^k$, and the output z^k is applied as the output of k^{th} feedback controller. Then, the output vector of feedback controllers is obtained as $\boldsymbol{u}_F = [u_F^1, \dots \quad u_F^k, \dots \quad u_F^n]^T$, where $u_F^k = z^k = \sum_{i=1}^{p}\sum_{j=1}^{q}\left(\bar{w}_{ij}^k d_{ij}^k\right)$.

Thus, the control input vector of the joint torques, $\boldsymbol{u}_\tau$, is determined by:

$$\boldsymbol{u}_\tau = \boldsymbol{u}_S - \boldsymbol{u}_F = [u_\tau^1, \dots \quad u_\tau^k, \dots \quad u_\tau^n]^T \tag{9}$$

2. Sliding Mode-Based Learning Algorithm

Assumption 1: Consider that all of the input signals (i.e., y_1^k and y_2^k) and their time derivatives (i.e., $\dot{y}_1^k$ and $\dot{y}_2^k$) are bounded by:

$$\begin{cases} \left|y_1^k(t)\right| \le b_y; \quad |\dot{y}_1(t)| \le b_{\dot{y}}; \quad \forall t \\ \left|y_2^k(t)\right| \le b_y; \quad |\dot{y}_2(t)| \le b_{\dot{y}}; \quad \forall t \end{cases} \tag{10}$$

with b_y and $b_{\dot{y}}$ are known positive constants.

Assumption 2: Suppose that all of the control input torques and their time derivatives are bounded by:

$$|u_\tau^k(t)| \le b_u; \quad |\dot{u}_\tau^k(t)| \le b_{\dot{u}}; \quad \forall t \tag{11}$$

with b_u and $b_{\dot{u}}$ are known positive constants.

Definition 1: By utilizing the SMC theory in [24], $\boldsymbol{u}_S$ can be defined as a time-varying sliding surface:

$$\boldsymbol{\beta}_u(\boldsymbol{u}_\tau, \boldsymbol{u}_F) = \boldsymbol{u}_S(t) = \boldsymbol{u}_F(t) + \boldsymbol{u}_\tau(t) = 0 \tag{12}$$

Definition 2: A sliding motion sustains on (12) after finite time t_u, if the satisfaction of the inequality $[\boldsymbol{\beta}_u(t)]^T \dot{\boldsymbol{\beta}}_u(t) < 0$ is achieved for all time t in some non-trivial semi-open sub-interval of a form as $[t, t_u) \subset (-\infty, t_u)$.

Theorem 1: Based on the above Assumptions and Definitions, given an initial value $\boldsymbol{u}_S(0)$, the convergence of the learning error $\boldsymbol{u}_S(t)$ to zero within t_u can be guaranteed, if the adaptive learning laws for the parameters of FWNNs are designed as:

$$\begin{cases} \dot{\alpha}_{A_i}^k = \dot{y}_1^k + c_{A_i}^k \vartheta sgn(u_S^k) \\ \dot{\alpha}_{B_j}^k = \dot{y}_2^k + c_{B_j}^k \vartheta sgn(u_S^k) \\ \dot{\delta}_{A_i}^k = \left(\delta_{A_i}^k + \dfrac{h_{A_i}^k}{\delta_{A_i}^k \left\{c_{A_i}^k\right\}^2}\right) \vartheta sgn(u_S^k) \\ \dot{\delta}_{B_j}^k = \left(\delta_{B_j}^k + \dfrac{h_{B_j}^k}{\delta_{B_j}^k \left\{c_{B_j}^k\right\}^2}\right) \vartheta sgn(u_S^k) \\ \dot{\varphi}_{ij}^k = -\dfrac{\bar{w}_{ij}^k}{[\bar{w}^k]^T \bar{w}^k} \vartheta sgn(u_S^k) \end{cases} \tag{13}$$

where $c_{A_i}^k = y_1^k - \alpha_{A_i}^k$, $c_{B_j}^k = y_2^k - \alpha_{B_j}^k$, $h_{A_i}^k = \left(1 - \left\{\delta_{A_i}^k c_{A_i}^k\right\}^2\right) / \left(2 - \left\{\delta_{A_i}^k c_{A_i}^k\right\}^2\right)$, $h_{B_j}^k = \left(1 - \left\{\delta_{B_j}^k c_{B_j}^k\right\}^2\right) / \left(2 - \left\{\delta_{B_j}^k c_{B_j}^k\right\}^2\right)$, $\bar{\boldsymbol{w}}^k = \left[\bar{w}_{11}^k, \dots \quad \bar{w}_{ij}^k, \dots \quad \bar{w}_{pq}^k\right]^T$, $sgn(.)$ is the sign function, and the learning speed ϑ is a sufficiently large positive constant which is designed for satisfying the condition $b_{\dot{u}} < \vartheta$.

Proof of Theorem 1:

From (3), the time derivatives of the membership functions in the k^{th} FWNN are written as follows:

$$\begin{cases} \dot{\mu}_{A_i}^k(y_1^k) = -2\dfrac{\sigma_{A_i}^k \dot{\sigma}_{A_i}^k}{h_{A_i}^k} \mu_{A_i}^k(y_1^k) \\ \dot{\mu}_{B_j}^k(y_2^k) = -2\dfrac{\sigma_{B_j}^k \dot{\sigma}_{B_j}^k}{h_{B_j}^k} \mu_{B_j}^k(y_2^k) \end{cases} \tag{14}$$

where:

$$\begin{cases} \sigma_{A_i}^k = \delta_{A_i}^k c_{A_i}^k = \delta_{A_i}^k\left(y_1^k - \alpha_{A_i}^k\right) \\ \sigma_{B_j}^k = \delta_{B_j}^k c_{B_j}^k = \delta_{B_j}^k\left(y_2^k - \alpha_{B_j}^k\right) \end{cases} \tag{15}$$

By differentiating (15), yields:

$$\begin{cases} \dot{\sigma}_{A_i}^k = \dot{\delta}_{A_i}^k\left(y_1^k - \alpha_{A_i}^k\right) + \delta_{A_i}^k\left(\dot{y}_1^k - \dot{\alpha}_{A_i}^k\right) \\ \dot{\sigma}_{B_j}^k = \dot{\delta}_{B_j}^k\left(y_2^k - \alpha_{B_j}^k\right) + \delta_{B_j}^k\left(\dot{y}_2^k - \dot{\alpha}_{B_j}^k\right) \end{cases} \tag{16}$$

The time derivative of w_{ij}^k is expressed as:

$$\dot{w}_{ij}^k = \dot{\mu}_{A_i}^k \mu_{B_j}^k + \dot{\mu}_{B_j}^k \mu_{A_i}^k = -\dot{s}_{ij}^k w_{ij}^k \tag{17}$$

where:

$$\dot{s}_{ij}^k = 2\left(\frac{\sigma_{A_i}^k \dot{\sigma}_{A_i}^k}{h_{A_i}^k} + \frac{\sigma_{B_j}^k \dot{\sigma}_{B_j}^k}{h_{B_j}^k}\right) \quad (18)$$

According to (17) and (18), $\dot{\bar{w}}_{ij}^k$ is determined as

$$\dot{\bar{w}}_{ij}^k = -\bar{w}_{ij}^k \dot{s}_{ij}^k + \bar{w}_{ij}^k \sum_{i=1}^p \sum_{j=1}^q \left(\bar{w}_{ij}^k \dot{s}_{ij}^k\right) \quad (19)$$

From (13), (15) and (16), it can be obtained that

$$\begin{cases} \frac{\sigma_{A_i}^k \dot{\sigma}_{A_i}^k}{h_{A_i}^k} = \frac{\sigma_{A_i}^k}{h_{A_i}^k}\left\{\dot{\delta}_{A_i}^k c_{A_i}^k + \delta_{A_i}^k\left(\dot{y}_1^k - \dot{\alpha}_{A_i}^k\right)\right\} = \vartheta sgn(u_S^k) \\ \frac{\sigma_{B_j}^k \dot{\sigma}_{B_j}^k}{h_{B_j}^k} = \frac{\sigma_{B_j}^k}{h_{B_j}^k}\left\{\dot{\delta}_{B_j}^k c_{B_j}^k + \delta_{B_j}^k\left(\dot{y}_2^k - \dot{\alpha}_{B_j}^k\right)\right\} = \vartheta sgn(u_S^k) \end{cases} \quad (20)$$

Take a Lyapunov function as follows:

$$\mathcal{L}_1(t) = \frac{1}{2}[\boldsymbol{\beta}_u(t)]^T \boldsymbol{\beta}_u(t) = \frac{1}{2}[\boldsymbol{u}_S]^T \boldsymbol{u}_S = \frac{1}{2}\sum_{k=1}^n \{u_S^k\}^2 \quad (21)$$

By differentiating (21) with respect to time yields:

$$\dot{\mathcal{L}}_1(t) = [\boldsymbol{\beta}_u(t)]^T \dot{\boldsymbol{\beta}}_u(t) = \sum_{k=1}^n \{\dot{u}_S^k u_S^k\} = \sum_{k=1}^n \{(\dot{u}_F^k + \dot{u}_\tau^k) u_S^k\} \quad (22)$$

By using (13), (18), (19), and (20), $\dot{u}_F^k$ is obtained as follows:

$$\begin{aligned} \dot{u}_F^k &= \sum_{i=1}^p \sum_{j=1}^q \left(\bar{w}_{ij}^k \dot{\varphi}_{ij}^k + \dot{\bar{w}}_{ij}^k \varphi_{ij}^k\right) = \\ &\sum_{i=1}^p \sum_{j=1}^q \Bigg[-\bar{w}_{ij}^k \frac{\bar{w}_{ij}^k}{[\bar{\boldsymbol{w}}^k]^T \bar{\boldsymbol{w}}^k} \vartheta sgn(u_S^k) + \Bigg(-2\bar{w}_{ij}^k \left\{\frac{\sigma_{A_i}^k \dot{\sigma}_{A_i}^k}{h_{A_i}^k} + \frac{\sigma_{B_j}^k \dot{\sigma}_{B_j}^k}{h_{B_j}^k}\right\} + \\ &\bar{w}_{ij}^k \sum_{i=1}^p \sum_{j=1}^q \left\{2\bar{w}_{ij}^k \left[\frac{\sigma_{A_i}^k \dot{\sigma}_{A_i}^k}{h_{A_i}^k} + \frac{\sigma_{B_j}^k \dot{\sigma}_{B_j}^k}{h_{B_j}^k}\right]\right\}\Bigg) \varphi_{ij}^k\Bigg] = \\ &\sum_{i=1}^p \sum_{j=1}^q \Bigg[-\bar{w}_{ij}^k \frac{\bar{w}_{ij}^k}{[\bar{\boldsymbol{w}}^k]^T \bar{\boldsymbol{w}}^k} \vartheta sgn(u_S^k) + \left(-4\bar{w}_{ij}^k \vartheta sgn(u_S^k) + \bar{w}_{ij}^k \sum_{i=1}^p \sum_{j=1}^q \left\{4\bar{w}_{ij}^k \vartheta sgn(u_S^k)\right\}\right) \varphi_{ij}^k\Bigg] = \\ &\vartheta sgn(u_S^k) \sum_{i=1}^p \sum_{j=1}^q \Bigg[-\bar{w}_{ij}^k \frac{\bar{w}_{ij}^k}{[\bar{\boldsymbol{w}}^k]^T \bar{\boldsymbol{w}}^k} + \left(-4\bar{w}_{ij}^k + 4\bar{w}_{ij}^k \sum_{i=1}^p \sum_{j=1}^q \{\bar{w}_{ij}^k\}\right) \varphi_{ij}^k\Bigg] = \\ &\vartheta sgn(u_S^k) \sum_{i=1}^p \sum_{j=1}^q \left[-\bar{w}_{ij}^k \frac{\bar{w}_{ij}^k}{[\bar{\boldsymbol{w}}^k]^T \bar{\boldsymbol{w}}^k}\right] = -\vartheta sgn(u_S^k) \end{aligned} \quad (23)$$

Based on (23), the Assumptions and the condition $b_{\dot{u}} < \vartheta$, it is concluded that $\dot{\mathcal{L}}(t)$ must be lower than zero for satisfying the stability of the learning:

$$\dot{\mathcal{L}}_1(t) = \sum_{k=1}^n \{(-\vartheta sgn(u_S^k) + \dot{u}_\tau^k) u_S^k\} \le \sum_{k=1}^n \{-\vartheta |u_S^k| + |\dot{u}_\tau^k||u_S^k|\} \le \sum_{k=1}^n \{-\vartheta |u_S^k| + b_{\dot{u}} |u_S^k|\} < 0, \quad \forall u_S^k \ne 0 \quad (24)$$

This completes the proof.

Assumption 3: Assume that the desired position vector $\boldsymbol{\theta}_d(t)$ is uniformly continuous and differentiable, and the vectors $\boldsymbol{\theta}_d(t)$, $\dot{\boldsymbol{\theta}}_d(t)$ and $\ddot{\boldsymbol{\theta}}_d(t)$ are bounded.

Theorem 2: Consider a dynamics system as (1), under all of Assumptions and Definitions, if an AFNC law is defined as (9), and an online adaptation strategy for the parameters of FWNNs is designed as (13), then the convergence of tracking errors and the stability of proposed control system can be ensured.

Proof of Theorem 2:

By using (8) and (12), a relation between $\boldsymbol{\beta}_\varepsilon$ and $\boldsymbol{\beta}_u$ is presented as follows:

$$\boldsymbol{\beta}_\varepsilon = [\boldsymbol{\Lambda}_S]^{-1} \boldsymbol{\beta}_u = [\boldsymbol{\Lambda}_S]^{-1} \boldsymbol{u}_S = \begin{bmatrix} \frac{1}{\lambda_S^1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\lambda_S^n} \end{bmatrix} [u_S^1, \dots \; u_S^k, \dots \; u_S^n]^T \quad (25)$$

For analyzing the tracking performance of the control system, a Lyapunov function is considered as follows:

$$\mathcal{L}_2(t) = \frac{1}{2}[\boldsymbol{\beta}_\varepsilon(t)]^T \boldsymbol{\beta}_\varepsilon(t) \quad (26)$$

Based on (25) and Theorem 1, the negative-definiteness of the time derivative of $\mathcal{L}_2(t)$ can be guaranteed:

$$\dot{\mathcal{L}}_2(t) = [\boldsymbol{\beta}_\varepsilon(t)]^T \dot{\boldsymbol{\beta}}_\varepsilon(t) = \sum_{k=1}^n \left\{\frac{1}{(\lambda_S^k)^2} \dot{u}_S^k u_S^k\right\} \le \sum_{k=1}^n \left\{\frac{1}{(\lambda_S^k)^2}\left(-\vartheta |u_S^k| + b_{\dot{u}} |u_S^k|\right)\right\} < 0, \quad \forall u_S^k \ne 0 \quad (27)$$

This completes the proof.

IV. COMPARATIVE SIMULATION RESULTS

Consider a two-DOF robot manipulator with the dynamics parameters as follows:

$$\boldsymbol{M}_r(\boldsymbol{\theta}) = \begin{bmatrix} l_1^2 m_1 + (l_1^2 + l_2^2) m_2 + \xi_m, & l_2^2 m_2 + \xi_m \\ l_2^2 m_2 + \xi_m, & l_2^2 m_2 \end{bmatrix} \quad (28)$$

$$\boldsymbol{V}_r(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} -l_1 l_2 m_2 \dot{\theta}_2 sin(\theta_2), & -l_1 l_2 m_2 (\dot{\theta}_1 + \dot{\theta}_2) sin(\theta_2) \\ l_1 l_2 m_2 \dot{\theta}_1 sin(\theta_2), & 0 \end{bmatrix} \quad (29)$$

$$\boldsymbol{g}_r(\boldsymbol{\theta}) = 9.81 \begin{bmatrix} l_1(m_1 + m_2) cos(\theta_2) + l_2 m_2 cos(\theta_1 + \theta_2) \\ l_2 m_2 cos(\theta_1 + \theta_2) \end{bmatrix} \quad (30)$$

where $\xi_m = 2 l_1 l_2 m_2 cos(\theta_2)$.

In order to demonstrate the robustness and the superior control performance of the proposed AFNC, both the AFNC system and the proportional differential control (PDC) system [2] are simulated for comparison.

The PDC system is illustrated in Figure 3, and the PDC law is defined as

$$\boldsymbol{u}_{pd} = \boldsymbol{K}_p \boldsymbol{\varepsilon} + \boldsymbol{K}_d \dot{\boldsymbol{\varepsilon}} \quad (31)$$

where the gain matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are derived from the tuning rules of Ziegler Nichols [25] by a compromise between the control performance and the stability:

$$\boldsymbol{K}_p = \begin{bmatrix} 1100 & 0 \\ 0 & 600 \end{bmatrix};\ \boldsymbol{K}_d = \begin{bmatrix} 45 & 0 \\ 0 & 30 \end{bmatrix} \qquad (32)$$

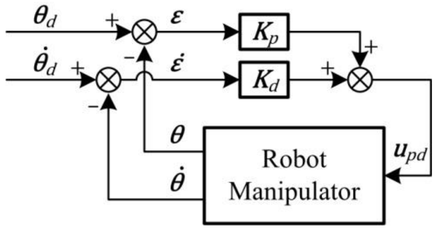

Figure 3: Structure of PDC system

In the AFNC system, all values of $\delta_{A_i}^k$, $\alpha_{A_i}^k$, $\delta_{B_j}^k$, and $\alpha_{B_j}^k$ are randomly initialised in the range of $[-0.1,\ \ 0.1]$, and other detailed parameters are given as follows:

$$p = q = 5;\ \vartheta = 0.01;\ \boldsymbol{\Lambda}_\beta = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix};\ \boldsymbol{\Lambda}_S = \begin{bmatrix} 60 & 0 \\ 0 & 60 \end{bmatrix} \qquad (33)$$

The nominal parameters of the robot system are given as in Tables 1.

Table 1: The nominal parameters of the robot system

DOF	DOF 1	DOF 2
Mass (kg)	$m_1 = 3$	$m_2 = 1.5$
Length (m)	$l_1 = 0.5$	$l_2 = 0.9$
Initial position (rad)	$\theta_1(0) = 0.8$	$\theta_2(0) = 0.8$
Initial velocity (rad/s)	$\dot{\theta}_1(0) = 0$	$\dot{\theta}_2(0) = 0$
Desired trajectory (rad)	$\theta_{d_1}(t) = e^{-t}$	$\theta_{d_2}(t) = e^{-2t}$

Herein, the simulation is implemented in two cases as follows:

Case 1: Have no the parameter variation, and consider the external disturbances term as:

$$\boldsymbol{\eta}_e = [4e^{-0.6t},\ \ 6e^{-0.4t}]^T \qquad (34)$$

Case 2: $\boldsymbol{\eta}_e$ as in (34), and the parameter variation (i.e., a tip load, 1 (kg), on DOF 2) is considered.

Besides, the root mean square error (RMSE) method is utilized to record the individual performance of control systems:

$$RMSE_k = \sqrt{\frac{1}{T_\omega}\sum_{\omega=1}^{T_\omega}[\theta_d^k(\omega) - \theta^k(\omega)]^2} \qquad (35)$$

where $\theta_d^k(\omega)$ is the ω^{th} element of θ_d^k, $\theta^k(\omega)$ is the ω^{th} element of θ^k, T_ω is the total sampling instants, and $k = 1, 2$.

The simulation results of the PDC system and the AFNC system in two cases, which comprise joint position, tracking error, and control torque, are depicted in Figures 4-7, respectively. Moreover, the values of RMSEs in both the PDC system and the AFNC system are expressed in Table 2.

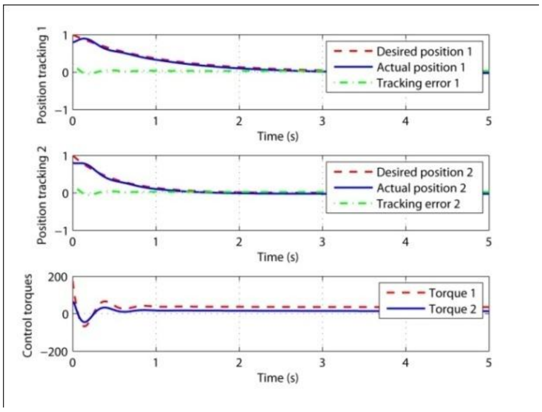

Figure 4: The simulation of PDC in Case 1

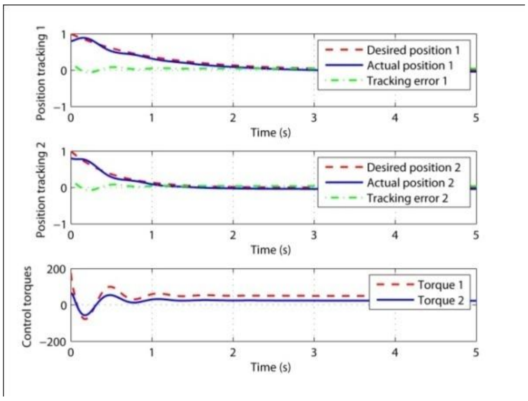

Figure 5: The simulation of PDC in Case 2

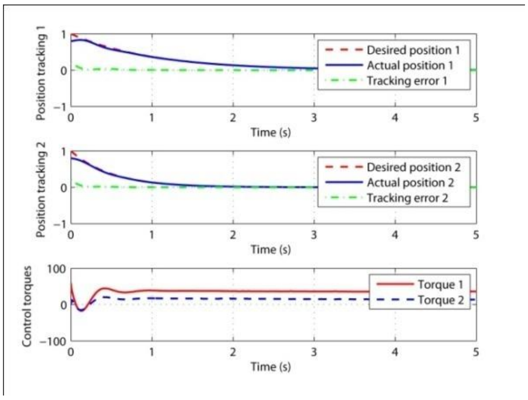

Figure 6: The simulation of AFNC in Case 1

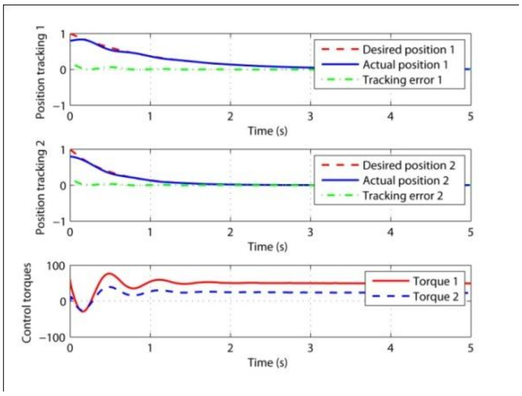

Figure 7: The simulation of AFNC in Case 2

Table 2: RMSEs of PDC system and AFNC system in two cases

RMSE (rad)	Case 1		Case 2	
	PDC	AFNC	PDC	AFNC
$RMSE_1$	0.0517	0.0279	0.0609	0.0298
$RMSE_2$	0.0478	0.0264	0.0578	0.0276

From Figures 4 and 5, the tracking performances of the PDC system are good. However, the convergences of tracking errors are still slow. In Figures 6 and 7, the joint positions can closely track the desired trajectories under the existence of the uncertainties, and the tracking errors are regularly reduced because of the learning ability of FWNNs.

The simulation results in Figures 4-7 and Table 2 show that the proposed AFNC system reaches the control performance improvement than that of the PDC system, while the convergence of tracking errors as well as the RMSEs of the proposed AFNC method is better than ones of the PDC method.

V. CONCLUSION

This paper has successfully applied an AFNC approach utilizing SMBLA for tracking the desired trajectory of robot manipulator. The AFNC scheme represents a parallel combination of FWNNs and TSMC, which not only approximates the various uncertainties but also guarantees the stability of the whole system. Additionally, the parameters of FWNNs are updated by a novel SMBLA that its convergence is proven by employing Lyapunov theorem. Hence, the proposed control system resulting in a robust and improved tracking performance without the detailed knowledge of robot manipulator. The comparative simulation results of two-DOF robot manipulator demonstrate that the tracking errors of the proposed AFNC method converge faster than ones of the PDC method.

VI. ACKNOWLEDGEMENTS

The authors would like to thank the editor and the reviewers for their valuable comments.

REFERENCES

[1]. M.W. Spong, Robot Dynamics and Control (New York: Wiley-Interscience, 1989).
[2]. J.-J.E. Slotine and W. Li, Applied Nonlinear Control (Englewood Cliffs, NJ: Prentice-Hall, 1991).
[3]. K. Liu and F.L. Lewis, Robust Control Techniques for General Dynamic Systems, Journal of Intelligent and Robotic Systems, 6 (1), 1992, 33-49.
[4]. C. Edwards and S.K. Spurgeon, Sliding Mode Control: Theory and Applications (London: Taylor and Francis, 1998).
[5]. O. Omidvar and D.L. Elliott, Neural Systems for Control (New York: Academic Press, 1997).
[6]. L.X. Wang, A Course in Fuzzy Systems and Control (Englewood Cliffs, NJ: Prentice-Hall, 1997).
[7]. S.M. Prabhu and D.P. Garg, Artificial neural network based robot control: An overview, Journal of Intelligent and Robotic Systems, 15 (4), 1996, 333-365.
[8]. S.-J. Huang and J.-S. Lee, A stable self-organizing fuzzy controller for robotic motion control, IEEE Transactions on Industrial Electronics, 47 (2), 2000, 421-428.
[9]. S.S. Ge, C.C. Hang, and L.C. Woon, Adaptive Neural Network Control of Robot Manipulators in Task Space, IEEE Transactions on Industrial Electronics, 44 (6), 1997, 746-752.
[10]. B.K. Yoo and W.C. Ham, Adaptive control of robot manipulator using fuzzy compensator, IEEE Transactions on Fuzzy Systems, 8 (2), 2000, 186-199.
[11]. S. Mahjoub, F. Mnif, N. Derbel, and M. Hamerlain, Radial-Basis-Functions Neural Network Sliding Mode Control for Underactuated Mechanical Systems, International Journal of Dynamics and Control, 2 (1), 2014, 1-9.
[12]. C.T. Lin and C.S.G. Lee, Neural Fuzzy Systems (Englewood Cliffs, NJ: Prentice-Hall, 1996).
[13]. R.J. Wai, Y.C. Huang, Z.W. Yang, and C.Y. Shih, Adaptive fuzzy-neural-network velocity sensorless control for robot manipulator position tracking, IET Control Theory & Applications, 4 (6), 2010, 1079-1093.
[14]. [H. Chaudhary, V. Panwar, R. Prasad, and N. Sukavanam, Adaptive neuro fuzzy based hybrid force/position control for an industrial robot manipulator, Journal of Intelligent Manufacturing, 2014, 1-10.
[15]. S. Beyhan and M. Alci, Extended fuzzy function model with stable learning methods for online system identification, International Journal of Adaptive Control and Signal Processing, 25 (2), 2011, 168-182.
[16]. C.-F. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, IEEE Transactions on Fuzzy Systems, 10 (2), 2002, 155-170.

[17]. R.A. Aliev, B.G. Guirimov, B. Fazlollahi, and R.R. Aliev, Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: Recurrent fuzzy neural networks, Fuzzy Sets and Systems, 160 (17), 2009, 2553-2566.
[18]. O. Kaynak, K. Erbatur, and M. Ertugnrl, The fusion of computationally intelligent methodologies and sliding-mode control-a survey, IEEE Transactions on Industrial Electronics, 48 (1), 2001, 4-17.
[19]. M.O. Efe, O. Kaynak, and B.M. Wilamowski, Stable training of computationally intelligent systems by using variable structure systems technique, IEEE Transactions on Industrial Electronics, 47 (2), 2000, 487-496.
[20]. G.G. Parma, B.R. Menezes, and A.P. Braga, Sliding mode algorithm for training multilayer artificial neural networks, Electronics Letters, 34 (1), 1998, 97-98.
[21]. H. Gomi and M. Kawato, Neural network control for a closed-loop System using Feedback-error-learning, Neural Networks, 6 (7), 1993, 933-946.
[22]. Q. Zhang and A. Benveniste, Wavelet networks, IEEE Transactions on Neural Networks, 3 (6), 1992, 889-898.
[23]. F.L. Lewis, C.T. Abdallah, and D.M. Dawson, Control of Robot Manipulators (New York: Macmillan Publishing Company, 1993).
[24]. Z. Yi and E. Meng Joo, An Evolutionary Approach Toward Dynamic Self-Generated Fuzzy Inference Systems, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 38 (4), 2008, 963-969.
[25]. K.J. Astrom and B. Wittenmark, Adaptive Control, 2nd ed. (Reading, MA: Addison-Wesley, 1995).