A Novel Robust Adaptive Intelligent and Compound Control of an Adaptive Neural Network, SMC, FLC and PI for Robot Manipulators

Duc Ha Vu\textsuperscript{1,2}, Shoudao Huang\textsuperscript{1} and Thi Diep Tran\textsuperscript{1,2}

\textsuperscript{1}College of Electrical and Information Engineering, Hunan University, Hunan, P.R. China
\textsuperscript{2}Faculty of Electrical Engineering, Sao Do University, Hai Duong, Vietnam

Corresponding author’s e-mail address: vuhadhsd@hnu.edu.cn

Abstract. In this paper, a new compound control scheme is proposed for robot manipulators (RMs) based on radial basis function neural network (RBFNNs), sliding mode control (SMC), fuzzy logic control (FLC) and proportional–integral (PI) controller. In this control scheme, the filtered tracking error is the input of the RBFNNs update laws, SMC, FLC and PI controller. The RBFNNs uses three-layer to approximate uncertain nonlinear manipulator dynamics. A robust sliding function is selected as a second controller to guarantee the stability and robustness under various environments. The FLC as the third controller completely removes the chattering signal caused by the sign function in the SMC. By using additional PI controllers, the goal of RMs tuning is to minimize tracking performance and overshoot can be realized. Simulation results highlight performance of the controller to compensate the approximate errors and its simplicity in the adaptive parameter tuning process. To be concluded, the controller is suitable for robust adaptive intelligent control and can be used as supplementary of traditional neural network (NN) controllers.

1. Introduction
Recently the development of intelligent controller based on NN for RMs has received considerable attention \cite{1, 2}. The properties of NN are learning, nonlinear mapping, and generalization abilities, parallelism of computation, vitality. Because of these characteristics, NN becomes important application in areas such as identification, optimization, intelligence control, robotics, etc. NN can provide online learning algorithms and deal with unmodeled unknown dynamics in robot model. These algorithms are designed based on the Lyapunov stability theorem. In \cite{3}, the authors proposed a control compensator using a neural network to compensate for the control errors of the control system. This controller has proposed a new command control based on tracking error of the desired position trajectories and actual position system when the parameter changes and large disturbances. In \cite{4}, the authors researched the tracking problem based on the SBLF and adaptive Neural Networks. The proposed controller was combined backstepping and adaptive feedback approximation method to improve the control performance. In \cite{5}, Bin Xu, Chengguang, and Zhongke Shi were proposed an adaptive critic based NN controller to solve the nonlinear systems. Both the tracking error and strategic utility function were improved by using the action neural network to approximate the nonlinear functions. In \cite{6}, an adaptive NN controller was suggested for an uncertain robot with unknown dynamics. The authors employed the adaptive neural networks to approximate the unknown model of the robot, shut the door on the violation...
of the constraints, and compensate for the unknown of the dynamic structure of the robot. In [7], the authors have been studied the tracking problem of 3-DOF robotic based on an adaptive neural network controller. This controller has been considered both output feedback and state control schemes by using two neural networks. A neural network was employed to approximate the dynamic of the robot and another neural network uses to approximate the unknown hysteresis non-linearity. In [8], a robust adapted controller on the basis of neural networks was introduced to control the SCARA robot arm. In [9] an adaptive output feedback control method uses the proposed RBFNNs to adaptive compensate for the tracking output of continuous nonlinear systems. In [10] an adaptive tracking control scheme based on the neural network is proposed for a class of nonlinear systems. In which RBFNNs is used to adaptive learn the uncertainty bounded of the system according to the Lyapunov stability theorem, and the output of the neural network is used as the parameters of the controller to compensate uncertainty impacts of the system. In the above documents have inherited advantages of the neuron controller, which is the ability to approximate and learning the rules online during the controller work. However, the neuron controller cannot avoid approximate errors. To solve the disadvantages of the neuron controller, it is necessary to provide a combination controller between the neuron controller and the sliding controller [11, 12]. In which, the authors have combined the neuron controller with the sliding controller to control industrial robot manipulators. The neuron controller with fast learning algorithm and good approximation ability and the sliding controller with robust compensating effect acts as a secondary controller to ensure stability and sustainability under the different environments. These two studies both achieve the correct tracking performance and high stability of the controller. However, the control signal in [11, 12] still has chattering phenomenon. To overcome the above disadvantages, in the author's paper, not only does the combination SMC with the neuron network (NN) to compensate the approximation error, but it also incorporates FLC and PI controller to removes chattering signal, improve tracking performance and overshoot. Therefore different with existing intelligent control methods, this method combines the RBFNNs, SMC, FLC and the PI to improve the robust ability, control performance and thus can be seemed like a robust adaptive intelligent controller. As shown in the simulation results, when applying this controller to control a two-link robot manipulator compared with the RBFNNs [12], PD the performance of the proposed control is improved so much.

2. Dynamics of manipulators

In general, the dynamic of an n-link robot manipulator may be expressed in the Lagrange as follows:

\[
M(q)\ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) = \tau - T_d
\]  

(1)

Where \((q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n\times1}\) are the vectors of joint position, angle velocity and angle acceleration, respectively. \(M(q) \in \mathbb{R}^{n\times n}\) is the symmetric inertial matrix. \(C(q, \dot{q}) \in \mathbb{R}^{n\times n}\) is the vector of Coriolis and Centripetal forces. \(G(q) \in \mathbb{R}^{n\times1}\) expresses the gravity vector. \(F(\dot{q}) \in \mathbb{R}^{n\times1}\) represents the vector of the frictions \(\tau \in \mathbb{R}^{n\times1}\) is vector of input torques. \(T_d \in \mathbb{R}^{n\times1}\) is vector of disturbance torques. For the purpose of designing controller, several properties of the manipulator model (1) have been assumed as follows:

Property 1: The inertial matrix \(M(q)\) is a positive symmetric matrix and is defined by:

\[
m_1\|x\|^2 \leq x^T M(q) x \leq m_2\|x\|^2, \forall x \in \mathbb{R}^n
\]

(2)

With \(m_1\) and \(m_2\) are known positive constants and they depend on the mass of the manipulators.

Property 2: \(M(q) - 2C(q, \dot{q})\) is skew symmetry matrix, in which

\[
x^T \left[ M(q) - 2C(q, \dot{q}) \right] x = 0
\]

(3)

Property 3: \(C(q, \dot{q})\dot{q}, F(\dot{q})\) and \(G(q)\) are bounded as follows:

\[
\|C(q, \dot{q})\dot{q}\| \leq C_k \|\dot{q}\|^2, \|F(\dot{q})\| \leq F_k \|\dot{q}\| + F_0, \|G(q)\| \leq G_k
\]

(4)

with \(C_k, G_k, F_k, F_0\) are positive constants.
3. Control strategy

Control strategy based on design of RBFNNs controller, SMC, FLC and the PI controller. The RBFNNs controller is designed based on the structure of radial basis function (RBF) neural networks. After analyzing the structure of RBF neural networks and the dynamics of manipulators, a new compound control scheme is proposed. Finally, demonstrate stability and capability of eliminating chattering signal of the proposed controller is shown.

3.1. The structure of RBF neural networks

RBF neural network is a type of network capable of approximation any continuous nonlinear functions with arbitrary accuracy by the number of neurons in the hidden layer. Furthermore, RBFNNs controllers have been successfully deal with manipulators control problems mentioned by many researchers. In this report, the RBF neural network proposed in [11] is selected as the network controller for its Lyapunov stability. Assume that $M(q), C(q, \dot{q}), G(q)$ and $F(\dot{q})$ the output values of ideal RBFNNs and determined, respectively as

$$M(q) = M_\alpha(q) + \varepsilon_M = W_M^T \cdot h_M(q) + \varepsilon_M$$

$$C(q, \dot{q}) = C_R(q, \dot{q}) + \varepsilon_C = W_C^T \cdot h_C(q, \dot{q}) + \varepsilon_C$$

$$G(q) = G_R(q) + \varepsilon_G = W_G^T \cdot h_G(q) + \varepsilon_G$$

$$F(\dot{q}) = F_R(\dot{q}) + \varepsilon_F = W_F^T \cdot h_F(\dot{q}) + \varepsilon_F$$

With $W_M, W_C, W_G$ and $W_F$ are ideal optimum weight value of RBF; $h_M, h_C, h_G$ and $h_F$ are outputs of hidden layer, $\varepsilon_M, \varepsilon_C, \varepsilon_G$ and $\varepsilon_F$ are modeling error of $M(q), C(q, \dot{q}), G(q)$ and $F(\dot{q})$, respectively. $\hat{M}_R(q), \hat{C}_R(q, \dot{q}), \hat{G}_R(q)$ and $\hat{F}_R(\dot{q})$ are the estimated values of the $M_R(q), C_R(q, \dot{q}), G_R(q)$ and $F_R(\dot{q})$, respectively. They are described by RBF as follows:

$$\hat{M}_R(q) = \hat{W}_M^T \cdot h_M(q)$$

$$\hat{C}_R(q, \dot{q}) = \hat{W}_C^T \cdot h_C(q, \dot{q})$$

$$\hat{G}_R(q) = \hat{W}_G^T \cdot h_G(q)$$

$$\hat{F}_R(\dot{q}) = \hat{W}_F^T \cdot h_F(\dot{q})$$

Where $\hat{W}_M, \hat{W}_C, \hat{W}_G$ and $\hat{W}_F$ are estimates of $W_M, W_C, W_G$ and $W_F$ respectively.

3.2. Compound control scheme

Robotic manipulator dynamics with multi-joints are highly nonlinear, highly coupling and form an uncertain model system. The RBFNNs controller has many advantages, which is the ability to approximate and learning the rules online during the controller work. However, the neuron controller cannot avoid approximate errors. As a result, a compound control method based on RBFNNs, SMC, FLC and PI is introduced for the position-tracking control of robot manipulators. The SMC with robust compensating effect acts as a secondary controller to ensure stability and sustainability under the different environments. The FLC completely removes chattering signal caused by the sign function in the SMC. The PI with the ability minimize tracking performance and overshoot. The architecture of the manipulator control system is shown in figure 1.

Determine a tracking error vector $e(t)$ and the filtered tracking error $s(t)$ as the following equations:

$$e(t) = q_d - q$$

$$s(t) = \hat{e} - \lambda e$$

Where $\lambda = \lambda^T > 0$, differentiating $s(t)$ and using (1), the manipulators dynamics may be written in terms of the sliding mode functions as:

$$M\dot{s} = -Cs + M(\dot{q}_d + \lambda \dot{e}) + C(\dot{q}_d + \lambda e) + G + F + T_d - \tau$$

Since equation (5) - (8), (15) becomes
\[ M\mathbf{s} = -Cs + f(x) - \tau + T_d + \varepsilon \]  
(16)

Where \( f(x) \) is defined as 
\[ f(x) = M_R(q) + C_R(q, \dot{q}) + G_R(q) + F_R \]
and 
\[ \varepsilon = e_d(q) + \lambda \dot{e} + e_c(q) + e_g + T_d \]

For the dynamics of an n-link robot manipulator (1), the compound control law is proposed as:
\[ \tau = \hat{f}(x) + \tau_s + \tau_{pl} \]  
(17)

Where:
\[ \tau_{pl} = K_p s + K_f \int_0^t s dt \]
\[ K_p = \text{diag}(K_{p1}, K_{p2}, ..., K_{pn}) \]
\[ K_f = \text{diag}(K_{f1}, K_{f2}, ..., K_{fn}) \]  
(18)

\( K_p, K_f \) is the positive definite matrix, \( \tau_s \) is a SMC robust term that is used to suppress the effects of uncertainties and approximation errors, and \( \hat{f}_s \) is the approximation of the adaptive function \( f_s \) and is defined as:
\[ \hat{f}(x) = \hat{M}_R(q) + \hat{C}_R(q, \dot{q}) + \hat{G}_R(q) + \hat{F}_R \]  
(19)

Figure 1. Proposed control scheme.

Follow above analysis, we propose a SMC robust term \( \tau_s \) as:
\[ \tau_s = K_s \text{sgn}(s) \]  
(20)

Where \( K_s = \text{diag}(K_{s1}, K_{s2}, ..., K_{sn}) \), and \( K_s > \| \varepsilon \| \).

Substituting (17) into (16), we have:
\[ M(q)s = -C(q, \dot{q})s + \hat{f}(x) - \tau_{pl} - \tau_s + T_d + \varepsilon \]  
(21)

Where 
\[ \hat{f}(x) = f(x) - \hat{f}(x) = \hat{M}_R(q) + \hat{C}_R(q, \dot{q}) + \hat{G}_R(q) + \hat{F}_R \]

Substituting (18), (19) and (20) into (17), we obtain:
\[ \tau = \hat{M}_R(q) + \hat{C}_R(q, \dot{q}) + \hat{G}_R + \hat{F}_R + K_s \text{sgn}(s) + K_p s + K_f \int_0^t s dt \]  
(22)

In control rule of the RMs system represented by equation (22) with \( \text{sgn}(s) \) function, this is the main cause of chattering in the system. A method of removing the chattering signal is to replace the fixed parameter in equation (22) by a variable value through the fuzzy logic controller. Value \( K_f \) will change under the extent of the filtered tracking error \( s \). When \( s \) is extremely small, namely the state variables move closer to zero then \( K_f \) will also decrease to zero to make the \( \text{sgn}(s) \) function no longer affect the \( \tau \) control signal. We have:
\[
\lim_{K_s \to 0} sgn(s) = 0 \quad (23)
\]

However, if is \( K_s \) small from the beginning, the uncontrolled signal will move very slowly towards the equilibrium position. But if \( K_s \) from the beginning is extremely large, the state variables of the system will quickly advance to the equilibrium position, but at equilibrium position, the system will fluctuate greatly. Therefore, the value \( K_s \) initial should be large enough so that \( \tau \) can pull the system to equilibrium position. When the system is in equilibrium then the smaller \( K_s \) is the better it is. To implement the above idea, the author changes the value of \( K_s \) based on value of the filtered tracking error \( s \).

We will compute \( s \) through a fuzzy controller, the input of the fuzzy controller is the value of \( s \). The If-Then rule base of this fuzzy logic controller is designed as:

1. \( R_1^f \): If \( s \) is A Then \( K_s^1 = A \);
2. \( R_2^f \): If \( s \) is B Then \( K_s^2 = B \);
3. \( R_3^f \): If \( s \) is C Then \( K_s^3 = C \);
4. \( R_4^f \): If \( s \) is D Then \( K_s^4 = D \);
5. \( R_5^f \): If \( s \) is E Then \( K_s^5 = E \);
6. \( R_6^f \): If \( s \) is F Then \( K_s^6 = F \);
7. \( R_7^f \): If \( s \) is G Then \( K_s^7 = G \);

The membership functions of linguistic labels A, B, C, D, E, F, G for the term \( s \) are shown in figure 2. The membership functions of linguistic labels A, B, C, D, E, F, G for the term \( K_s \) are shown in figure 3.

![Figure 2. Membership function of each input.](image)

![Figure 3. Membership function of each output.](image)

By applying the adaptive control law equation (22) to the dynamic equation (1), using the SMC function equation (20), and using the PI controller equation (18), the online RBF neural networks adaptive update laws are designed as:

\[
\begin{align*}
\dot{W}_M &= k_M \Xi_M s \cdot \Xi_M h_M s^T \\
\dot{W}_C &= k_c \Xi_c s \cdot \Xi_c h_c s^T
\end{align*}
\]

\[
\dot{W}_G = k_G \Xi_G s \cdot \Xi_G h_G s^T \\
\dot{W}_F = k_f \Xi_f s \cdot \Xi_f h_f s^T
\]

Where \( \Xi_M, \Xi_c, \Xi_G, \Xi_f \) are symmetric positive definite constant matrices. \( k_M, k_c, k_G, k_f \) are positive adaptation rates.

### 3.3. Demonstrate stability and capability of eliminating chattering signal of the proposed controller

Two theorems will be proved in this section. Theorem 1 is to analyze the asymptotic stability of the closed-loop system in figure 1 which is described by equation (22). Theorem 2 involves analyzing the ability of eliminating chattering signal of the proposed controller.

Theorem 1: Consider an n-link robot manipulator represented by (1). If the RBFNNs adaptive update laws are designed as (24), the SMC is given by (20), and PI controller (18). The adaptive control law
designed in (22), then the tracking error and the convergence of all the system parameters can approached to zero asymptotically.

Proof: Consider the following candidate Lyapunov function:

\[
V(t) = \frac{1}{2} \left[ s^T M s + \left( \int_0^t s dt \right)^T K_i \left( \int_0^t s dt \right) + \text{tr}(\ddot{W}_M^T \dot{\dot{W}}_M) + \text{tr}(\ddot{W}_C^T \dot{\dot{W}}_C) + \text{tr}(\ddot{W}_G^T \dot{\dot{W}}_G) + \text{tr}(\ddot{W}_F^T \dot{\dot{W}}_F) \right]
\]  

(25)

Where

\[
\ddot{W}_M = \dot{W}_M - \dot{\dot{W}}_M, \ddot{W}_C = \dot{W}_C - \dot{\dot{W}}_C, \ddot{W}_G = \dot{W}_G - \dot{\dot{W}}_G, \ddot{W}_F = \dot{W}_F - \dot{\dot{W}}_F.
\]

The derivative of \( V(t) \) along to time, we have:

\[
\dot{V}(t) = s^T \left( M s + K_i \int_0^t s dt + \text{tr}(\dot{W}_M^T \dot{W}_M) + \text{tr}(\dot{W}_C^T \dot{W}_C) + \text{tr}(\dot{W}_G^T \dot{W}_G) + \text{tr}(\dot{W}_F^T \dot{W}_F) \right)
\]  

(26)

Submitting (21) into (26), yields

\[
\dot{V}(t) = -s^T K_p s - \frac{1}{2} s^T (M - 2C) s + s^T \dot{e} - s^T \tau_e + \text{tr}(\dot{W}_M^T \dot{W}_M) + \text{tr}(\dot{W}_C^T \dot{W}_C) + \text{tr}(\dot{W}_G^T \dot{W}_G) + \text{tr}(\dot{W}_F^T \dot{W}_F)
\]

By using property 2, and since \( \ddot{W} = -\dot{W}, s^T \dot{W} h(x) = tr(\dot{W}^T h(x)s^T) \), and from adaptive law (24), (26) becomes:

\[
\dot{V}(t) = -s^T K_p s + s^T \dot{e} - s^T \tau_e + s^T \dot{W}_M(k_m \dot{W}_M) + s^T \dot{W}_C(k_c \dot{W}_C) + s^T \dot{W}_G(k_g \dot{W}_G) + s^T \dot{W}_F(k_f \dot{W}_F)
\]

By using \( tr(W - \ddot{W}) = (\ddot{W}, W) \leq \| \ddot{W} \| \| W \| - \| \ddot{W} \| \) and \( \| W \| \leq W_{\text{max}} \)

We have

\[
\dot{V}(t) \leq -s^T K_p s + s^T \dot{e} - s^T \dot{W}_M + k_m s (\| \ddot{W}_M \| (\| W_M \| - \| \ddot{W}_M \|)) + k_c s (\| \ddot{W}_C \| (\| W_C \| - \| \ddot{W}_C \|)) + k_g s (\| \ddot{W}_G \| (\| W_G \| - \| \ddot{W}_G \|)) + k_f s (\| \ddot{W}_F \| (\| W_F \| - \| \ddot{W}_F \|))
\]

\[
\leq -s^T K_p s + s^T \dot{e} - s^T \dot{W}_M + k_m s (\| \ddot{W}_M \| (W_{\text{max}} - \| \ddot{W}_M \|)) + k_c s (\| \ddot{W}_C \| (W_{\text{max}} - \| \ddot{W}_C \|)) + k_g s (\| \ddot{W}_G \| (W_{\text{max}} - \| \ddot{W}_G \|)) + k_f s (\| \ddot{W}_F \| (W_{\text{max}} - \| \ddot{W}_F \|))
\]

(27)

In (28), considering \( K_i > \| \dot{e} \| \), and if gain \( K_p \) and \( s \) are selected to satisfy the following inequality:
$$K_p \geq \frac{1}{s} \left[ \frac{k_m W_{M_{\text{max}}}^2}{4} + \frac{k_c W_{C_{\text{max}}}^2}{4} + \frac{k_a W_{A_{\text{max}}}^2}{4} + \frac{k_s W_{S_{\text{max}}}^2}{4} \right]$$

(29)

Then

$$\dot{V}(t) \leq 0$$

(30)

Since (30), $$\dot{V}(s(t), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F) \leq 0$$ is a negative semidefinite function,

$$\dot{V}(s(t), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F) \leq \dot{V}(s(0), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F)$$ if all parameters such as $$s(t), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F$$ are bounded with $$t > 0$$. By defining $$\Gamma(t) = s^T K_p s - s^T s + s^T K_i \text{sgn}(s)$$, so $$\Gamma(t) \leq -\dot{V}(t)$$ and integrate the $$\Gamma(t)$$ with respect to time as follows:

$$\int_0^t \Gamma(\xi) d\xi \leq V(s(0), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F) - V(s(t), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F)$$

Because $$V(s(0), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F)$$ is a bounded function, and $$V(s(t), \bar{W}_M, \bar{W}_C, \bar{W}_G, \bar{W}_F)$$ is nonincreasing and bounded, we have

$$\lim_{t \to \infty} \int \Gamma(\xi) d\xi < \infty$$

(31)

According to Barbalat’s Lemma [13], when $$\Gamma(t)$$ is bounded function. It can be shown that

$$\lim_{t \to \infty} \int \Gamma(t) dt = 0.$$ From this result, we see that $$s(t)$$ will converge to zero when $$t \to \infty$$ and the global stability of the control system for the manipulator is guaranteed by the compound control law (17).

Theorem 2: Consider an n-link robot manipulator represented by (1). If the adaptive control law is defined as (22) and the fixed parameter $$K_s$$ in equation (22) is substitute by a replacement cost based on the magnitude of $$s$$ filtered tracking error through the fuzzy logic controller, the chattering signal in the system will be completely eliminated.

Proof: From equation (22), it is clear that the main component causing the chattering phenomenon in the system is the function $$K_s \text{sgn}(s)$$. To overcome this phenomenon, we add a fuzzy processing element in the controller to eliminate the $$\text{sgn}$$.

By the focal defuzzification method parameter $$K_s$$ is defined:

$$K_s = \sum_{i=1}^{7} \beta_i K_i'$$

(32)

In there $$\beta_i$$ is the correctness of the $$i$$th rule:

$$\beta_1 = \mu_a(s); \beta_2 = \mu_b(s); \beta_3 = \mu_c(s);$$

$$\beta_4 = \mu_d(s); \beta_5 = \mu_e(s); \beta_6 = \mu_f(s);$$

$$\beta_7 = \mu_g(s)$$

(33)

From (32) and (33) we obtain:

$$\lim_{s \to 0} K_s = \lim_{s \to 0} \sum_{i=1}^{7} \beta_i K_i' = 0$$

(34)

From (34) deduce

$$\lim_{s \to 0} K_s \text{sgn}(s) = 0$$

(35)
According to theorem 1 we have:

\[
\lim_{t \to \infty} s = 0
\]  

(36)

From (35) and (36) we deduce:

\[
\lim_{t \to \infty} K_s \text{sgn}(s) = 0
\]  

(37)

According to (37) when time \( t \) tends to \( \infty \), function \( K_s \text{sgn}(s) \) is completely eliminated in adaptive control law (22). Thus, chattering signal at the tracking position has been completely eliminated in the proposed controller.

4. Simulation results and discussions

In order to show the effectiveness of the proposed control system, simulation experiments are carried out. In this section, we consider a two-link robot manipulator. The dynamics of the manipulator can be obtained as in [12]. In the simulation experiments, the desired position trajectories are chosen by:

\[
q_{id} = q_{zd} = 0.5 \sin(t), \text{ and initial positions of joints are } q_0 = \begin{bmatrix} 0.09 & -0.09 \end{bmatrix}^T, \text{ and initial velocities of joints are } \dot{q}_0 = \begin{bmatrix} 0.0 & 0.0 \end{bmatrix}^T.
\]

The parameter values used in the compound control system are:

\[
\begin{align*}
\Xi_M &= \Xi_c = \Xi_q = \Xi_\beta = \text{diag}[15, 15]; \\
\lambda &= \text{diag}[5, 5]; \\
K_p &= \text{diag}[50, 50]; \\
K_r &= \text{diag}[100, 100]; \\
K_s &= \text{diag}[0.1, 0.1].
\end{align*}
\]

In the following passage, our proposed neuron network control scheme is applied to comparison with the RBFNNs [12] and the proportional differential (PD) controller, where the output of the PD controller can be expressed to be \( y_{pd} = K_s \chi(t) + K_s \beta(t) \), and the PD gains were selected to be \( K_s = \text{diag}[200, 250]; K_\beta = \text{diag}[20, 20] \).

Figure 4 shows the position tracking trajectories of the two joints by using the RBFNNs, PD and the proposed controller. Figure 5 presents the tracking errors of the two joints by using the RBFNNs, PD and the proposed controller. In figure 6, the control efforts of the RBFNNs, PD and the proposed controller is presented. To validate ability completely removes chattering signal, control input of the RBFNNs, PD and the proposed controller for zoomed-in time frame \((5 - 7s)\) can see clearly in figure 7.

The above simulation results for the two-link manipulator demonstrate that the PD control has low trajectory tracking precision, slow response time and large position-tracking errors. This is because the PD controller only controls for objects with nominal models. The PD controllers are unable to determine the appropriate PD gains in the case of nonlinear and uncertain controlled plants and can’t compensate static errors.

As shown in figure 4, 5, 6 the RBFNNs and the proposed controller can realize higher trajectory tracking precision and better adaptive control capability due to the powerful capabilities of learning adaptability. This is also because in the RBFNNs controller and the proposed controller contains a robust component. It is the SMC that compensates approximate errors in the neuron network controller. However, see in Figure 6, 7 the SMC causes in the RBFNNs is chattering phenomenon for the control force. To solve the above problem, in our controller with simple and more effective parameter update law. Our update laws only have four parameters to update while the update laws in RBFNNs have six parameters to update. This leads to our controller being able to calculate and simulate faster. In addition, by using additional the FLC and PI controllers, the goal of the proposed controller is to eliminating chattering signal, minimize error tracking and overshoot is achieved when compared with the RBFNNs, can see in figure 4, 5, 6, 7. This is because proportional gain \( K_p \) works in conjunction with integral gain \( K_i \) to reduce overshoot, tracking error and provide damping to the system, while keeping response time to acceptable levels. Moreover, Fuzzy logic control rules is used to replace for sign function of sliding control law. From figure 6, 7 can observe that the control force of the proposed controller is smoother than the RBFNNs and PD to achieve the requested level of performance when the tracking errors reach the big value. The advantages of the proposed control scheme have position...
tracking improvements better than that of the RBFNNs, PD and all updated parameters in the dynamic structure RBFNNs are simpleness adjusted. Theory and simulation results have show that the proposed controller not only has better tracking performance but also completely suppresses the chattering phenomenon when compared to the RBFNNs and PD controllers.

Figure 4. Position responses of the RBFNNs, PD and the proposed controller: (a) link-1; (b) link-2.

Figure 5. Tracking errors of the RBFNNs, PD and the proposed controller: (a) link-1; (b) link-2.

Figure 6. Control efforts of the RBFNNs, PD and the proposed controller: (a) link-1; (b) link-2.
5. Conclusions

In this paper, a new compound NN, SMC, FLC and PI control scheme has been proposed. It has been also successfully implemented to control the joints of a two-link robot manipulator for achieving high precision position tracking by combining the advantages of radial basis function neural network, sliding mode robust term function, the FLC and PI controller to completely removes chattering signal, improve tracking performance and overshoot. The difficulty to find approximate values of the unknown dynamic of robot manipulator has been to solve by radial basis function neural network control. All the adaptive online training for the weights of the radial basis function neural network are obtained by Lyapunov theorem, and trained online by an adaptive learning algorithm. From the simulation results of a two-links robot manipulator, we can see that the performance of the proposed control is improved so much. The future research work, we can continue to research to put into experimental as well as be applied in practice.

Reference

[1] Wu Y, Huang R, Li X and Liu S 2019 Adaptive neural network control of uncertain robotic manipulators with external disturbance and time-varying output constraints Neurocomputing 323 108-16

[2] Luan F, Na J, Huang Y and Gao G 2019 Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence Neurocomputing 337 153-64

[3] Choi Y K, Lee M J, Kim S and Kay Y C 2001 Design and implementation of an adaptive neural-network compensator for control system IEEE Transactions on Industrial Electronics 48 416-23

[4] Zhao Z, He W and Ge S S 2014 Adaptive neural network control for a fully actuated marine surface vessel with multiple output constraints IEEE Trans. Control Systems Technology 22 1536-43

[5] Xu B, Yang C G and Shi Z K 2014 Reinforcement learning output feedback NN control using deterministic learning techniques IEEE Trans. Neural Network and Learning Systems 25 635-41

[6] He W, Chen Y and Zhao Y 2016 Adaptive neural network control of an uncertain robot with full-state constraints IEEE Transactions on Cybernetics 46 620-29

[7] He W, David O A, Yang C and Dawei G 2017 Adaptive neural network control of a robotic manipulator with un-known backlash-like hysteresis IET Control Theory Appl 11 567-75

[8] Rossomando F G and Soria C M 2016 Adaptive neural sliding mode control in discrete time for a SCARA robot arm IEEE Latin America transactions 14 2556-64

[9] Seshagiri S and Khail H K 2000 Output feedback control of nonlinear systems using RBF neural networks IEEE Trans. Neural Networks 11 69–79
[10] Zhihong M, Wu H R and Palaniswami M 1998 An adaptive tracking controller using neural networks for a class of nonlinear systems IEEE Trans. Neural Networks 9 947–55

[11] Yen V T, Nan W Y and Cuong P V 2019 Robust Adaptive Sliding Mode Neural Networks Control for Industrial Robot Manipulators International Journal of Control, Automation and Systems 17 783-92

[12] Cuong P V and Nan W Y 2015 Adaptive trajectory tracking neural network control with robust compensator for robot manipulators Neural Computing and Applications 27 525-36

[13] Slotine J J E and Li W 1991 Applied Nonlinear Control (Hoboken: Prentice-Hall)