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

Duc Ha Vu1,2, Shoudao Huang1 and Thi Diep Tran1,2
1 College of Electrical and Information Engineering, Hunan University, Hunan, China
2 Faculty of Electrical Engineering, Saodo University, Haiduong, Vietnam
Corresponding author’s e-mail address: vuhadhsd@hnu.edu.cn

Abstract. In this paper, a new compound control scheme is proposed for manipulator based on radial basis function neural network (RBFNNs), sliding mode controller (SMC) and proportional–integral (PI) controller. In this control scheme, the filtered tracking error is the input of the RBFNNs update laws, SMC, 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. By using additional PI controllers, the goal of manipulator tuning is to minimize chattering signal, 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 control and can be used as supplementary of traditional neural network (NN) controllers.

1. Introduction
Artificial neural networks, as a kind of tools of computational intelligence, have been credited in various applications as powerful tools capable of providing robust controllers for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties [1]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods, as it shows that they are theoretically capable of uniformly approximating a continuous real function to any degree of accuracy. This has led to recent advances in the area of intelligent control [2]. The properties of NN are learning, nonlinear mapping, generalization abilities and parallelism of computation. Because of these characteristics, NN becomes an 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 manipulator model. These algorithms are designed on the basis of the Lyapunov stability theorem. In [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 [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 [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 [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 RBF network 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 RBF neural network 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
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 a PI controller to improve chattering signal, tracking
performance and overshoot. Therefore different with existing control methods, this method combines
the RBFNNs, SMC and the PI to improve the robust ability, control performance and thus can be
seemed like a robust adaptive 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
Consider the dynamic equation of the n-link rigid manipulator can be written as:

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = \tau - T_d \]  \hspace{1cm} (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. \(F(q) \in \mathbb{R}^{n\times1}\) 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:

\[ m_1\|x\|^2 \leq x^T M(q) x \leq m_2\|x\|^2, \forall x \in \mathbb{R}^n \] \hspace{1cm} (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

\[ x^T [M(q) - 2C(q, \dot{q})] x = 0 \] \hspace{1cm} (3)

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

\[ \|C(q, \dot{q})\| \leq C, \|F(q)\| \leq F_1, \|G(q)\| \leq G \] \hspace{1cm} (4)
with $C_1, G_1, F_1, F_0$ are positive constants.

3. Control strategy

Control strategy based on design of RBFNNs controller, SMC 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.

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

$$M(q) = M_\text{h}(q) + \varepsilon_M = W_M^T \ast h_M(q) + \varepsilon_M$$

$$C(q, \dot{q}) = C_\text{h}(q, \dot{q}) + \varepsilon_C = W_C^T \ast h_C(q, \dot{q}) + \varepsilon_C$$

$$G(q) = G_\text{h}(q) + \varepsilon_G = W_G^T \ast h_G(q) + \varepsilon_G$$

$$F(\dot{q}) = F_\text{h}(\dot{q}) + \varepsilon_F = W_F^T \ast h_F(\dot{q}) + \varepsilon_F$$

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. $\hat{M}_\text{h}(q), \hat{C}_\text{h}(q, \dot{q}), \hat{G}_\text{h}(q)$ and $\hat{F}_\text{h}(\dot{q})$ are the estimated values of the $M_\text{h}(q), C_\text{h}(q, \dot{q}), G_\text{h}(q)$ and $F_\text{h}(\dot{q})$, respectively. They are described by RBF as follows:

$$\hat{M}_\text{h}(q) = \hat{W}_M^T \ast h_M(q)$$

$$\hat{C}_\text{h}(q, \dot{q}) = \hat{W}_C^T \ast h_C(q, \dot{q})$$

$$\hat{G}_\text{h}(q) = \hat{W}_G^T \ast h_G(q)$$

$$\hat{F}_\text{h}(\dot{q}) = \hat{W}_F^T \ast h_F(\dot{q})$$

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 and PI is introduced for the position-tracking control of manipulators. The architecture of the manipulator control system is shown in Fig. 1.

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

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

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

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

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

Since equation (5) - (8), (15) becomes:

$$M \dot{s} = -Cs + f(x) - \tau + T_d + \varepsilon$$

Where $f(x)$ is defined as $f(x) = M_\text{h}(\ddot{q}_d + \lambda \dot{e}) + C_\text{h}(\dot{q}_d + \lambda e) + G_\text{h} + F_\text{h}$ and

$$\varepsilon = e_M(\ddot{q}_d + \lambda \dot{e}) + e_C(\dot{q}_d + \lambda e) + e_G + e_F + T_d$$

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

where:

\[ \tau_{pf} = K_p s + K_i \int_0^\tau s dt, \quad K_p = \text{diag}\{K_{p1}, K_{p2}, \ldots, K_{pv}\}, \quad K_i = \text{diag}\{K_{i1}, K_{i2}, \ldots, K_{iv}\} \]  
(18)

\[ K_p, K_i \] 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 \( \hat{f}(x) \) is the approximation of the adaptive function \( f(x) \) and is defined as:

\[ \hat{f}(x) = \hat{M}_s (\ddot{q}_d + \lambda \dot{e}) + \hat{C}_s (\dot{q}_d + \lambda e) + \hat{G}_s + \hat{F}_g \]  

\[ \tau_s = K_s \text{sgn}(s) \]  
(19)

where \( K_s = \text{diag}\{K_{s1}, K_{s2}, \ldots, K_{sv}\} \), and \( K_s > \|0\| \).

Substituting (17) into (16), we have:

\[ M(q)\dot{s} = -C(q, \dot{q})s + \hat{f}(x) - \tau_{pf} - \tau_s + T_d + \varepsilon \]  
(20)

where

\[ \hat{f}(x) = f(x) - \hat{f}(x) = M_s (\ddot{q}_d + \lambda \dot{e}) + \hat{C}_s (\dot{q}_d + \lambda e) + \hat{G}_s (q) + \hat{F}_g \]  
(21)

By applying the adaptive control law equation (17) to the dynamic equation (1), using the SMC function equation (19), and using the PI controller equation (18), the online RBF neural networks adaptive update laws are designed as:

\[
\begin{align*}
\dot{W}_M &= k_M \Xi_M \hat{s} \cdot \Xi_M h_M s^T \\
\dot{W}_c &= k_c \Xi_c \hat{s} \cdot \Xi_c h_c s^T \\
\dot{W}_G &= k_G \Xi_G \hat{s} \cdot \Xi_G h_G s^T \\
\dot{W}_F &= k_F \Xi_F \hat{s} \cdot \Xi_F h_F s^T
\end{align*}
\]  
(22)

where \( \Xi_M, \Xi_c, \Xi_G \) and \( \Xi_F \) are symmetric positive definite constant matrices, \( k_M, k_c, k_G \) and \( k_F \) are positive adaptation rates.

The stability of the closed-loop system in Fig. 1 which is described by equation (17) is established in the following theorem.

**Theorem:** Consider an n-link manipulator represented by (1). If the RBFNNs adaptive update laws are designed as (22), the SMC is give by (19), and PI controller (18). The adaptive control law designed in (17), then the tracking error and the convergence of all the system parameters can approached to zero asymptotically.

**Proof:** Consider the following candidate Lyapunov function:

**Figure 1.** Proposed control scheme.
where \( \vec{W}_M = W_M - \dot{W}_M, \vec{W}_C = W_C - \dot{W}_C, \vec{W}_G = W_G - \dot{W}_G, \vec{W}_F = W_F - \dot{W}_F \).

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

\[
\dot{V}(t) = s^T Ms + \frac{1}{2} \dot{M} s + M \int_0^t \dot{s}(tdt)^T K_s \left( \int_0^t \dot{s}(tdt) + \text{tr}(\vec{W}_C \dot{\Xi}_C^* \vec{W}_C) \right) + \text{tr}(\vec{W}_C \dot{\Xi}_C^* \vec{W}_C) + \text{tr}(\vec{W}_G \dot{\Xi}_G^* \vec{W}_G) + \text{tr}(\vec{W}_F \dot{\Xi}_F^* \vec{W}_F) \tag{23}
\]

By using property 2, and since \( \dot{\vec{W}} = -\dot{\vec{W}}_s \dot{\vec{W}}_h(x) = \text{tr}(\vec{W} \dot{\vec{h}}(x) s^T) \), and from adaptive law (22), (24) becomes:

\[
\dot{V}(t) = -s^T K_p s + s^T (M - 2C) s + s^T \epsilon - s^T \tau_s + \text{tr}(\dot{\vec{W}}_M \dot{\Xi}_M^* \dot{\vec{W}}_M) + h_M s^T) + \text{tr}(\dot{\vec{W}}_C \dot{\Xi}_C^* \dot{\vec{W}}_C) + \text{tr}(\dot{\vec{W}}_G \dot{\Xi}_G^* \dot{\vec{W}}_G) + \text{tr}(\dot{\vec{W}}_F \dot{\Xi}_F^* \dot{\vec{W}}_F) \tag{24}
\]

By using \( \tau \dot{\vec{W}}(W - \vec{W}) = (\vec{W}, \dot{\vec{W}}) - \|\dot{\vec{W}}\|^2 \leq \|\dot{\vec{W}}\|\|W\| - \|\dot{\vec{W}}\|^2 \) and \( \|\dot{\vec{W}}\| \leq \|W\| \)

We have

\[
\dot{V}(t) \leq -s^T K_p s + s^T \epsilon - s^T K_s \text{sign}(s) + K_M s \left( \|\dot{\vec{W}}_M\| \|W_M\| - \|\dot{\vec{W}}_M\|^2 \right) + K_C s \left( \|\dot{\vec{W}}_C\| \|W_C\| - \|\dot{\vec{W}}_C\|^2 \right) + K_G \left( \|\dot{\vec{W}}_G\| - \frac{W_{G_{\text{max}}}}{2} \right) + K_F \left( \|\dot{\vec{W}}_F\| - \frac{W_{F_{\text{max}}}}{2} \right) \tag{25}
\]

In (26), considering \( K_s > \|\epsilon\| \), and if gain \( K_p \) and \( s \) are selected to satisfy the following inequality:

\[
K_p \geq \frac{1}{s} \left[ \frac{k_M W_{M_{\text{max}}}}{4} + \frac{k_C W_{C_{\text{max}}}}{4} + \frac{k_G W_{G_{\text{max}}}}{4} + \frac{k_F W_{F_{\text{max}}}}{4} \right] \tag{26}
\]

Then

\[
\dot{V}(t) \leq 0 \tag{28}
\]

Since (28), \( \dot{V}(s(t), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F) \leq 0 \) is a negative semidefinite function, \( \dot{V}(s(0), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F) \leq \dot{V}(s(0), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F) \) if all parameters such as \( s(t), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F \) are bounded with \( t > 0 \). By defining \( \Gamma(t) = s^T K_p s - s^T \epsilon + s^T K_s \text{sign}(s) \), so \( \Gamma(t) \leq -\dot{V}(t) \) and integrate the \( \Gamma(t) \) with respect to time as follows:

\[
\int_0^t \Gamma(\zeta) d\zeta \leq V(s(0), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F) - V(s(t), \dot{\vec{W}}_M, \dot{\vec{W}}_C, \dot{\vec{W}}_G, \dot{\vec{W}}_F) \]
Because $V(s(0),\dot{\mathbf{w}}_M,\ddot{\mathbf{w}}_C,\dot{\mathbf{w}}_G,\ddot{\mathbf{w}}_F)$ is a bounded function, and $V(s(0),\dot{\mathbf{w}}_M,\ddot{\mathbf{w}}_C,\dot{\mathbf{w}}_G,\ddot{\mathbf{w}}_F)$ is nonincreasing and bounded, we have:

$$\lim_{t \to \infty} \int_{0}^{t} \Gamma(\mathbf{z})d\mathbf{z} < \infty \quad (29)$$

According to Barbalat’s Lemma [13], when $\Gamma(t)$ is bounded function. It can be shown that $\lim_{t \to \infty} \int_{0}^{t} \Gamma(t)dt = 0$. 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).

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: $q_{gd} = q_{zd} = 0.5 \sin(t)$, and initial positions of joints are $q_{0} = [0.09 \quad -0.09]^T$, and initial velocities of joints are $\dot{q}_{0} = [0.0 \quad 0.0]^T$. The parameter values used in the compound control system are:

$$\Xi_M = \Xi_C = \Xi_G = \Xi_F = \text{diag}[15, 15]; \lambda = \text{diag}[5, 5]; K_p = \text{diag}[50, 50]; K_I = \text{diag}[100, 100];$$

$$K_c = \text{diag}[0.1, 0.1].$$

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 $y_{pd} = K \chi e(t) + \dot{K}_\chi \dot{e}(t)$, and the PD gains were selected to be $K_p = [200, 250]; K_I = [20, 20].$

Fig. 2 shows the position tracking trajectories of the two joints by using the RBFNNs, PD and the proposed controller. Fig. 3 presents the tracking errors of the two joints by using the RBFNNs, PD and the proposed controller. In Fig. 4, the control efforts of the RBFNNs, PD and the proposed controller is presented. To validate ability minimize chattering signal, control input of the RBFNNs, PD and the proposed controller for zoomed-in time frame (5 – 7 s) can see clearly in Fig 5.

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 Fig. 2, 3 and 4 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 Fig. 4, 5 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 PI controllers, the goal of the proposed controller is to minimize chattering signal, error tracking, and overshoot is achieved when compared with the RBFNNs, can see in Fig. 2, 3, 4 and 5. 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. 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. Moreover, from Fig. 4, 5 can observe that the control force of the proposed controller is smoother and has smaller oscillation than the RBFNNs and PD to achieve the requested level of performance when the tracking errors reach the big value.

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

**Figure 3.** Tracking errors of the RBFNNs, PD and the proposed controller: (a) link-1; (b) link-2.

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

**Figure 5.** Control efforts of the RBFNNs, PD and the proposed controller for zoomed-in time frame (5–7 s): (a) link-1; (b) link-2.
5. Conclusions
In this paper, a new compound NN, SMC 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 and PI controller to improve chattering signal, tracking performance and overshoot. The difficulty to find approximate values of the unknown dynamic of 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.

6. Reference
[1] Chaoui H and Gueaieb W 2008 Type-2 fuzzy logic control of a flexible-joint manipulator Journal of Intelligent and Robotic Systems 51 159-86
[2] Sun C, He W and Hong J 2017 Neural network control of a flexible robotic manipulator using the lumped spring-mass model IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 1863-74
[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 unknown 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)