Design of Robust Adaptive Controller for Industrial Robot Based on Sliding Mode Control and Neural Network

Tran Duc Chuyen, Hoa Van Doan*, Pham Van Minh, and Vu Viet Thong

Faculty of Electrical Engineering, University of Economics-Technology for Industries, Ha Noi, Viet Nam
Email: tdchuyen@uneti.edu.vn (T.D.C.), pvminh@uneti.edu.vn (P.V.M.), vtvthong@uneti.edu.vn (V.V.T.)
*Correspondence: rvhoa@uneti.edu.vn (H.V.D.)

***Abstract*—Today, industrial robots play an important role in industrial production lines. One of the most important problems in motion control of industrial robot systems is the tracking of reference motion trajectories. However, in designing the controller, it is difficult to build an accurate mathematical model for the robot. Especially in the real-time working process, the industrial robot is always affected by external noise, variable load, nonlinear friction, and unexpected changes in model parameters. To solve this problem, the paper which is built a robust adaptive controller based on the sliding mode controller and the RBF neural network. In the controller, the RBF neural network is used to approximate the unknown dynamics and the adaptive update law of the parameters of the network is built based on Lyapunov stability theory. The results of the controller are verified on Matlab Simulink software and show good tracking and high robustness.**

***Keywords*—robust adaptive control, sliding mode control, RBF neural network, robot**

I. INTRODUCTION

Today, industrial robots play an important role in industrial production lines. One of the important problems in motion control of industrial robot systems is the tracking of reference motion trajectories. Currently, there are many methods to build a controller for asymptotic tracking robot motion according to the preset sample signal. The basic traditional method to build a controller that satisfies the above requirements is gravity-compensated PD [1]. However, during real-time work, industrial robots are always affected by external noise, load changes, nonlinear friction, unexpected changes in model parameters (due to loss of model), wear of equipment, and deviation of system specifications after a long working time). Therefore, the gravity compensation PD controller does not guarantee the control quality for the robot during real-time work. In order to improve control quality as well as limit some disadvantages of PD controller, adaptive controller, backstepping control, sliding mode control have been introduced in the documents [2-4], and adaptive sliding mode control in works [5-9]. When the dynamics equations have uncertain parameters and are affected by noise, then robust adaptive control is included in the design [10-12]. Currently, intelligent controllers based on fuzzy control and neural networks have been widely applied in industrial robot control [13-20]. Fuzzy controllers are an effective tool for approximating nonlinear systems [13, 14, 16, 19]. In [19], an adaptive controller based on fuzzy logic has been applied to control a nonlinear system with an uncertain structure and the presence of external noise in the control process. Here fuzzy logic is used to approximate the unknown dynamics of the nonlinear system. However, fuzzy controller while building control rules often depends on the experience of the designer. Documents [18] designed a robust adaptive controller based on the sliding mode controller and neural network to control industrial robots, by combining the advantages of the sliding mode controller and the online learning ability of the controller, neuron control. The results of [18] give good quality of tracking control, but the neural network only approximates the friction function and noise, the controller still depends on the dynamic parameter.

From the above analysis, the article builds a robust adaptive controller based on the combination of the sliding mode controller and the RBF neural network, in which the controller is designed without knowing the robot dynamics while having to take into account both noise and friction factors.

II. ROBOT DYNAMICS

Consider the dynamics equation of an n-joint manipulator as Eq. (1) [21]:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - F(\dot{q}) - \tau_d \qquad (1)$$

where: The variables $q, \dot{q}, \ddot{q} \in R^{n\times 1}$ are the position, speed and angular acceleration of the joints, $H(q) \in R^{n\times n}$ is the matrix of moments of inertia, $C(q,\dot{q}) \in R^{n\times n}$ is the matrix of centrifugal and Coriolis forces, $G(q) \in R^{n}$ is the gravity components, $F(\dot{q}) \in R^{n}$ is the friction force, τ_d is the noise, and τ is the torque acting on the joints.

The dynamics Eq. (1) has the following properties:

Manuscript received July 1, 2022; revised September 14, 2022; accepted December 27, 2022.

doi: 10.18178/ijmerr.12.3.145-150

Property 1: The matrix of moments of inertia $H(q)$ is a symmetric, positive definite matrix and there exist two positive numbers m_1, m_2 satisfying

$$m_1 I \leq H(q) \leq m_2 I \tag{2}$$

Property 2: $\dot{H}(q) - 2C(q, \dot{q})$ is a skewed symmetric matrix for any vector x:

$$x^T\left(\dot{H}(q) - 2C(q, \dot{q})\right)x = 0 \tag{3}$$

Property 3: The matrix $C(q, \dot{q})$ is bounded, that is, for known $c_b(q)$ it exists

$$\|C(q, \dot{q})\| \leq c_b(q)\|\dot{q}\| \tag{4}$$

Property 4: Unknown noise τ_d blocked, $\|\tau_d\| \leq \tau_m$ (τ_m is a positive constant)

III. Controller Design

A. *Neural Network Model RBF*

RBF network can be used to folate f(x) , as described in Fig. 1 [22].

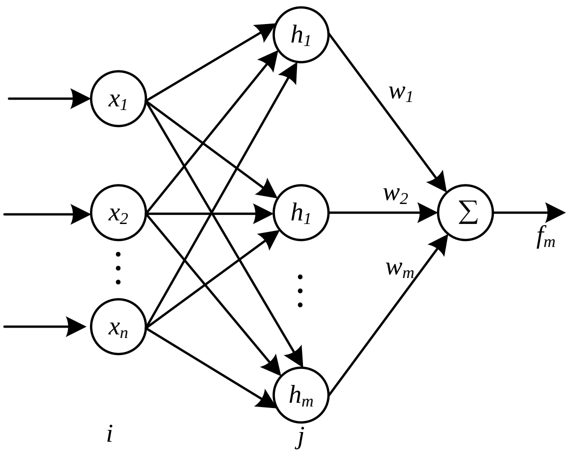

Figure 1. Neural Network Structure RBF

The structure of the RBF neural network consists of 3 layers:

-Layer 1 is the input layer including input variables $x_1, x_2, \cdots, x_n$

-The second layer is the hidden layer: The output of the hidden layer is calculated according to the following formula:

$$h_j = \exp\frac{\|x - c_i\|^2}{b_j^2}, j = 1,2,\ldots,n \tag{5}$$

-Layer 3 is the output layer of the neural network, calculated as follows:

$$f(x) = W^T h + \varepsilon \tag{6}$$

where x is the input state of the network, $h = [h_j]^T$ is the Gaussian function, m is the number of hidden layer neurons, c_i is the center of the function, b_j is the dispersion of the function, ε is the approximate error of the neural network, $W = [w_1, w_2, \ldots, w_m]^T$ is the weight vector number of the RBF network.

Here we use a neural network for approximation. Therefore, it will exist an optimal neuron function as follows:

$$f(x) = W^* h + \varepsilon \tag{7}$$

Here: The optimal weight value is W^*, and the approximate error vector is ε.

Assumption: Approximate deviation is limited:

$$\|\varepsilon\| \leq \varepsilon_0 \tag{8}$$

where ε_0 is a positive real value.

We use the RBF network to approximate f(x) , we have:

$$\hat{f}(x) = \widehat{W}^T h \tag{9}$$

With $\widetilde{W} = W - \widehat{W}, \|W\|_F \leq W_{max}$

From Eq. (6) and Eq. (9), we have:

$$f - \hat{f} = W^T h + \varepsilon - \widehat{W}^T h = \widetilde{W}^T h + \varepsilon \tag{10}$$

From the expression $f(x)$, the input of the RBF network is chosen as:

$$x = [e^T \quad \dot{e}^T \quad q_d^T \quad \dot{q}_d^T \quad \ddot{q}_d^T] \tag{11}$$

B. *Controller Structure*

The multi-link robot is a MIMO nonlinear system with cross-interactions. Therefore, it is difficult to accurately determine the parameters of the industrial robot model, due to the complexity of determining the values of mass, torque as well as geometrical dimensions of the robot. In addition, the parameters can be changed depending on the working mode of the robot, so the kinetic and dynamic parameters of the robot are considered uncertain parameters. The control objective in this paper is to build a preset trajectory tracking controller for the uncertain model to ensure the closed system is stable and global robust, the tracking error is zero, and is not affected by noise.

The controller's task is to control the movement of joints $q(t)$ following the set signal $q_d(t)$. Determine the tracking error as:

$$e(t) = q_d(t) - q(t) \tag{12}$$

Then the sliding surface is selected as follows:

$$s = \dot{e} + \wedge e \tag{13}$$

in there, $\wedge$ is a symmetric positive definite constant matrix and $\wedge = \wedge^T = [\lambda_1 \quad \lambda_2 \quad \ldots \quad \lambda_n]^T > 0$, so we have:

$$\dot{q} = -s + \dot{q}_d + \wedge e \tag{14}$$

We have the tracking error kinematics [22]:

$$\begin{aligned} H\dot{s} &= H(\ddot{q}_d - \ddot{q} + \wedge \dot{e}) = H(\ddot{q}_d + \wedge \dot{e}) - H\ddot{q} \\ &= H(\ddot{q}_d + \wedge \dot{e}) + C\dot{q} + G + F + \tau_d - \tau \\ &= H(\ddot{q}_d + \wedge \dot{e}) - Cs + C(\dot{q}_d + \wedge e) + G + F + \tau_d - \tau \\ &= -Cs - \tau + f + \tau_d \end{aligned} \tag{15}$$

In there:

$$f(x) = H(\ddot{q}_d + \wedge \dot{e}) + C(\dot{q}_d + \wedge e) + G + F \tag{16}$$

where f(x) is unknown, so we need to approximate $f(x)$, in this paper RBF network is used to approximate f(x) .

The controller is designed as:

$$\tau = \hat{f}(x) + K_v s - v \quad (17)$$

where K_v is a symmetric positive definite constant matrix, $\hat{f}(x)$ is the output of the RBF network. $\hat{f}(x)$ is an approximation of f(x) . The sliding mode controller is designed as:

$$v = -(\varepsilon_N + b_d)\,\mathrm{sgn}(s) \quad (18)$$

In there: $\|\varepsilon\| \le \varepsilon_N, \|\tau_d\| \le b_d$

To reduce chattering, the function sgn(s) is replaced by the function tanh(s) (hyperbolic tangent).

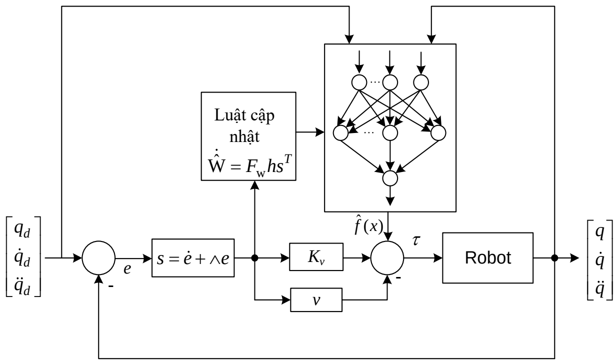

Figure 2. Structure of robust adaptive control system based on sliding mode control and RBF neural network

C. Proof of Stability

From Eqs. (15) and (17), we have:

$$H\dot{s} = -(K_v + C)s + \tilde{W}^T\varphi(x) + (\varepsilon + \tau_d) + v = -(K_v + C)s + \zeta_1 \quad (19)$$

In which $\zeta_1 = \tilde{W}^T\varphi(x) + (\varepsilon + \tau_d) + v$ (20)

Choose the Lyapunov [22] function as:

$$L = \frac{1}{2}s^T H s + \frac{1}{2}tr(\tilde{W}^T F_w^{-1}\tilde{W}) \quad (21)$$

where H and F are positive matrices, we have:

$$\dot{L} = s^T H\dot{s} + \frac{1}{2}s^T\dot{H}s + \frac{1}{2}tr\left(\tilde{W}^T F_w^{-1}\dot{\tilde{W}}\right) \quad (22)$$

From Eq. (19) we have:

$$\dot{L} = s^T K_v\dot{s} + \frac{1}{2}s^T(\dot{H} - 2C)s + tr\tilde{W}^T\left(F_w^{-1}\dot{\tilde{W}} + \varphi s^T\right) + s^T(\varepsilon + \tau_d + v) \quad (23)$$

We know that robot dynamics has the characteristic $s^T(\dot{H} - 2C)s = 0$. Choose $\dot{\tilde{W}} = F_w hs^T$, that is, the weight update adaptive rule of the network is:

$$\dot{\hat{W}} = F_w hs^T \quad (24)$$

So,

$$\dot{L} = -s^T K_v s + s^T(\varepsilon + \tau_d + v) \quad (25)$$

Because:

$$s^T(\varepsilon + \tau_d + v) = s^T(\varepsilon + \tau_d) + s^T v = s^T(\varepsilon + \tau_d) - \|s\|(\varepsilon_N + b_d) \le 0 \quad (26)$$

Hence we have: $\dot{L} \le -s^T K_v s \le 0$

From above analysis, we can see that RBF approximation error can be overcome by the robust term.

From $\dot{L} \le -s^T K_v s \le 0$, we have:

$$\int_0^t \dot{L}dt \le \lambda_{min}(K_v)\int_0^t \|s\|dt \quad (27)$$

$$L(t) - L(0) \le \lambda_{min}(K_v)\int_0^t \|s\|dt \quad (28)$$

where $\lambda_{min}(K_v)$ is the minimum eigenvalue of K_v.

Then, L is limited, s and $\tilde{W}$ are all limited, from $\dot{s}$ expression, $\dot{s}$ is limited, and the $\int_0^\infty \|s\|dt$ is limited. From Barbalat Lemmma [23], when $t \to \infty$, we have $s \to \infty$, then $e \to 0$; $\dot{e} \to 0$, and the convergence precision is related to K_v.

Since $L \ge 0$; $\dot{L} \le 0$, L is limited as $t \to \infty$; thus, $\hat{W}$ is limited. Since when $\dot{L} \equiv 0$, we cannot get $\tilde{W} \equiv 0$; $\hat{W}$ cannot converge to W.

IV. Simulation Results

The simulation results of the controller are illustrated on the three degrees of freedom manipulator model, the manipulator model is designed on Simscape Multibody as shown in Fig. 3.

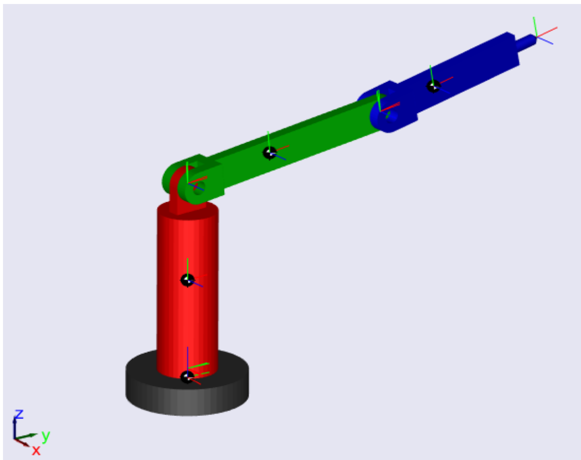

Figure 3. Robot model by Simscape Multibody on Simulink

The robot parameters are given in Table I. Simultaneously, the model of friction force and impact noise is simulated as follows:

$$F(\dot{q}) = 0.2\,sgn(\dot{q})$$
$$\tau_d = [0.2\,sin(t) \quad 0.2\,sin(t) \quad 0.2\,sin(t)]^T$$

The detailed system parameters of the three-link robot model are given as follows [24]:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$$G = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$$

$$H_{11} = l_1^2\left(\frac{m_1}{3} + m_2 + m_3\right) + l_1 l_2 (m_2 + 2m_3)cos(q_2) + l_2^2\left(\frac{m_2}{3} + m_3\right)$$

$$H_{12} = H_{21} = -l_1 l_2\left(\frac{m_2}{2} + m_3\right)cos(q_2) - l_2^2\left(\frac{m_2}{3} + m_3\right)$$

$$H_{13} = H_{23} = H_{31} = H_{32} = 0$$

$$H_{22} = -l_2^2\left(\frac{m_2}{3} + m_3\right)$$

$$H_{33} = m_3$$

$$C_{11} = -\dot{q}_2(m_2 + 2m_3)$$

$$C_{12} = C_{21} = -\dot{q}_2\left(\frac{m_2}{2} + m_3\right)$$

$$C_{13} = C_{22} = C_{23} = C_{31} = C_{32} = C_{33} = 0$$

$$g_1 = g_2 = g_3 = -m_3 g$$

To verify the effectiveness of the control algorithm, we simulated on Matlab - Simulink software with sine reference trajectory. The robust, adaptive controller based on the proposed sliding mode controller and neural network (SRBF) was compared with the gravity compensated PD controller in the literature [1] to demonstrate the advantages of this method. The robot parameters are given in the table as follows:

TABLE I. PARAMETERS OF THE THREE DEGREES OF FREEDOM MANIPULATOR MODEL

Symbol	**Meaning**	**Unit**	**Value**
m_1	Mass of link 1	kg	1.8
m_2	Mass of link 2	kg	0.5
m_3	Mass of link 3	kg	0.4
l_1	Length of link 1	m	0.7
l_2	Length of link 2	m	0.8
l_3	Length of link 3	m	0.6

The controller is designed with: $\Lambda = diag[30,30,30]$, $K_v = diag[300,300,300]$, RBF neural network structure with the number of hidden layer neurons is 7.

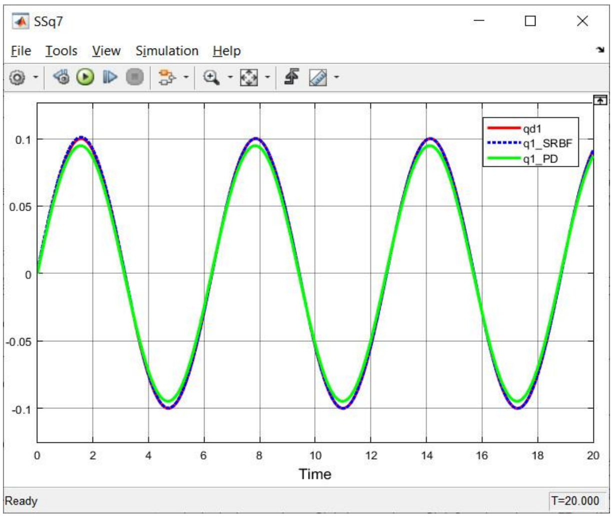

Figure 4. Simulation result of position tracking for joint 1

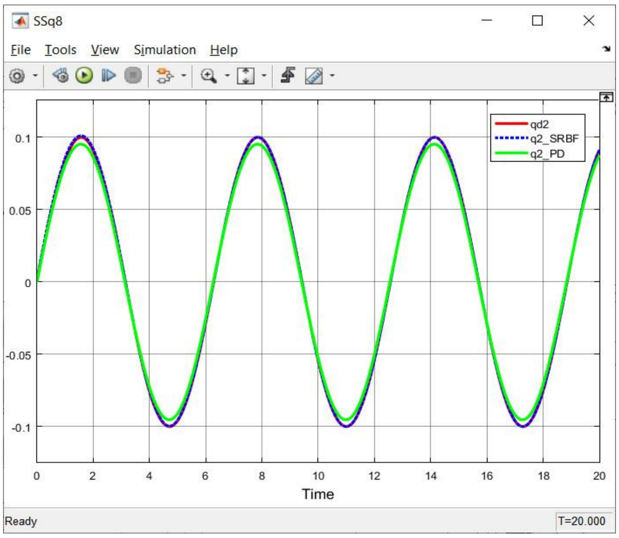

Figure 5. Simulation result of position tracking for joint 2

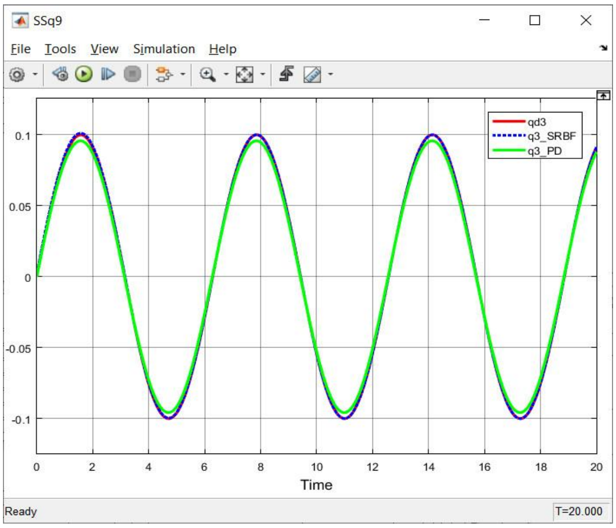

Figure 6. Simulation result of position tracking for joint 3

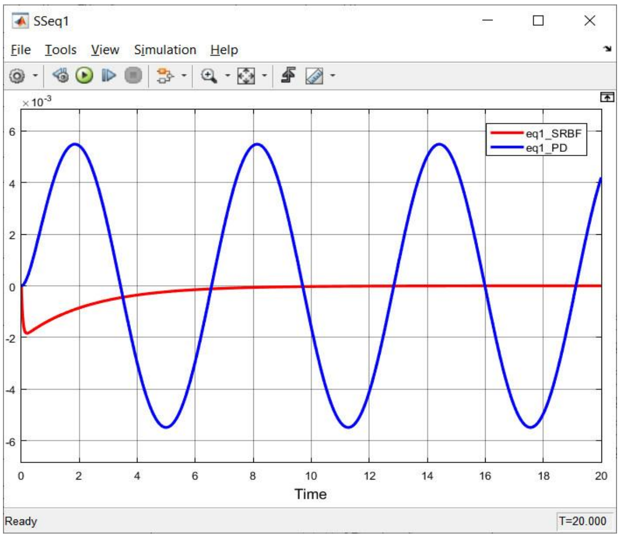

Figure 7. Simulation result of position tracking error for joint 1

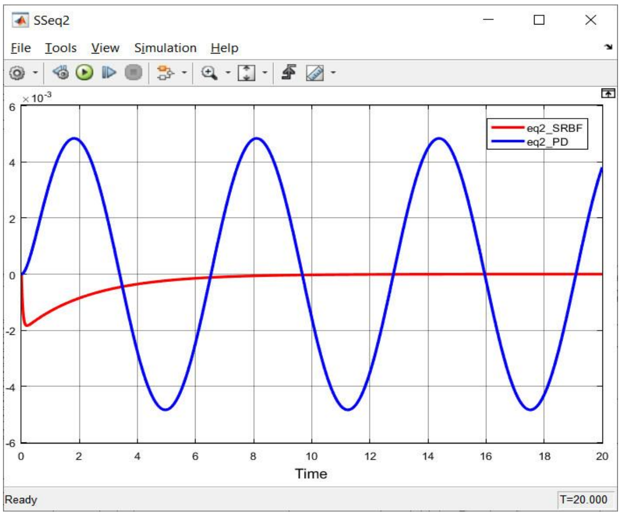

Figure 8. Simulation result of position tracking error for joint 2

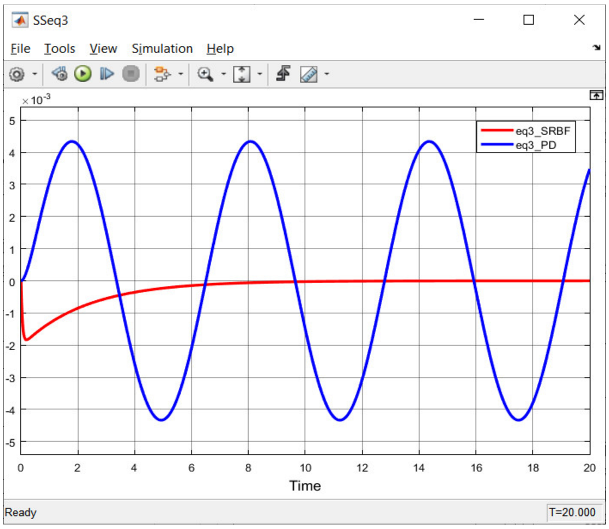

Figure 9. Simulation result of position tracking error for joint 3

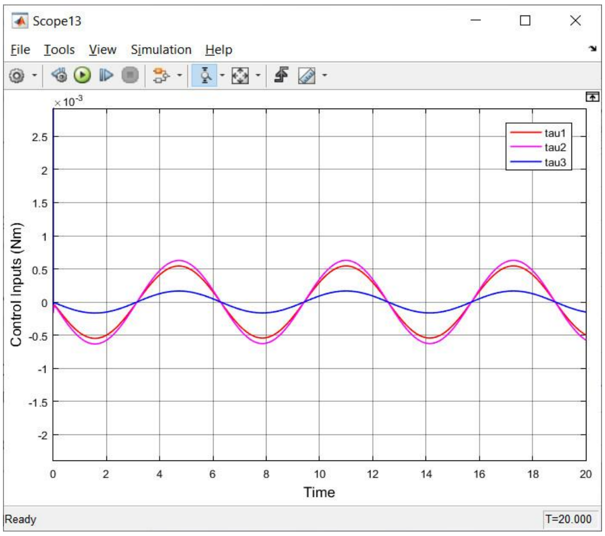

Figure 10. Simulation result of control inputs

TABLE II. SIMULATION PERFORMANCE OF THE CONTROLLER

Tracking error	**Joint 1**	**Joint 2**	**Joint 3**
Average error by location (rad)	-7.407×10^{-5}	-7.753×10^{-5}	-8.033×10^{-5}
Average error by velocity (rad/s)	1.855×10^{-5}	1.124×10^{-5}	2.151×10^{-5}

Above is a drawing of simulation results on the software. Where qd1, qd2, qd3 are the placement positions of joints 1, 2, 3; q1_SRBF, q2_SRBF, q3_SRBF are the positions of joints 1, 2, 3 using the proposed controller; q1_PD, q2_PD, q3_PD are the positions of joints using gravity compensation PD controller; eq1_SRBF, eq2_SRBF, eq3_SRBF are tracking errors using the proposed controller; eq1_PD, eq2_PD, eq3_PD are tracking errors using gravity compensated PD controller. Comparing the tracking errors in the robot manipulator motion compared to the set trajectory when using the SRBF controllers in turn, the PD controller can see that both controllers can guarantee the tracking error of the system relative to the set orbitals. However, there is still a difference in quality between the controllers in the above results, when using the SRBF controller, it shows that the quality of tracking is much better than that of the PD controller; position error of robot joints of SRBF controller converges faster, smaller and more stable than PD controller.

Through the simulation results in Figs. 4–10, we see that, during the working process, the system is affected by noise, the change of friction or the load changes the designed controller still converges, ensuring ensure stability and robust in the working process. Moreover, by using sliding mode control to compensate for estimation error as well as in the working process the weights of the neural network controller are always updated continuously through the learning rule, so the position error The position of robot joints of the faster converging controller is smaller and more stable as shown in Table II. Thereby proving that the robot control quality by using the robust adaptive controller based on sliding mode control and neural network RBF has been improved. Thereby we can continue to research to put it into practice as well as apply it in practice.

V. CONCLUSION

The robust adaptive controller is built based on a sliding mode control and RBF neural network to apply control to industrial robots, which has achieved good quality of tracking control, high accuracy in the environment noisy working environment, and taking into account the friction factor without knowing the dynamics of the robot. The built-in controller is also proven to be stable throughout the working area based on Lyapunov stability theory. The simulation results are verified on the three-degree-of-freedom robot model, compared with the PD controller. The simulation results show that the SRBF controller has good noise resistance, tracking error, and better stability than the PD controller. Thereby we can continue to research to put it into the experiment as well as to be applied in practice.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Tran Duc Chuyen conducted the research; Hoa Van Doan analyzed the data; Hoa Van Doan, Pham Van Minh, and Vu Viet Thong wrote, commented, and revised the paper; all authors had approved the final version.

FUNDING

The authors wish to thank team research about Artificial Intelligence "*CHD*" of Faculty of Electrical Engineering, University of Economics - Technology for Industries. This work was supported in part by a grant from University of Economics-Technology for Industries. Website: http://www.uneti.edu.vn/.

REFERENCES

[1] L. L. Frank, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control Theory and Practice*, CRC Press, 2003.

[2] I. Shafiqul and X. P. Liu, "Robust sliding mode control for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2444-2453, 2010.

[3] P. V. Vicente, et al., "Dynamic sliding PID control for tracking of robot manipulators: Theory and experiments," vol. 19, no. 6, pp. 967-976, 2003.

[4] Š. Asif, "Variable structure systems with sliding modes in motion control—A survey," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 212-223, 2011.

[5] R. D. Yadav, V. N. Sankaranarayanan, and S. Roy, "Adaptive sliding mode control for autonomous vehicle platoon under unknown friction forces," in *Proc. 2021 20th International Conference on Advanced Robotics (ICAR)*, 2021.

[6] V. N. Sankaranarayanan, S. Roy, and S. Baldi, "Aerial transportation of unknown payloads: adaptive path tracking for quadrotors," in *Proc. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[7] S. Roy, S. Baldi, and L. M. Fridman, "On adaptive sliding mode control without a priori bounded uncertainty," *Automatica,* 2019.

[8] S. Roy, S. Baldi, P. Li, and V. N. Sankaranarayanan, "Artificial-delay adaptive control for underactuated euler–lagrange robotics," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, December 2021.

[9] S. Roy, S. Baldi, and P. A. Ioannou, "An adaptive control framework for underactuated switched Euler–lagrange systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, August 2022.

[10] S. Roy, J. Lee, and S. Baldi, "A new adaptive-robust design for time delay control under state-dependent stability condition," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2020.

[11] S. Roy and S. Baldi, "On reduced-complexity robust adaptive control of switched Euler–Lagrange systems," *Nonlinear Analysis: Hybrid Systems*, vol. 34, pp. 226–237, 2019.

[12] S. Roy, E. B. Kosmatopoulos, and S. Baldi, "On vanishing gains in the robust adaptation of switched systems: A new leakage-based result for a class of Euler–Lagrange dynamics," *Systems & Control Letters*, vol. 144, p. 104773, 2020.

[13] B. Chen and X. P. Liu, "Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by backstepping and application to chemical processes," *IEEE Transactions on Fuzzy systems*, vol. 13, no. 6, pp. 832-847, 2005.

[14] K. Vineet and K. P. S. Rana, "Nonlinear adaptive fractional order fuzzy PID control of a 2-link planar rigid manipulator with payload," *Journal of the Franklin Institute*, vol. 354, no. 2, pp. 993-1022, 2017.

[15] T. S. Li, S. C. Tong, and G. Feng, "A novel robust adaptive-fuzzy-tracking control for a class of nonlinearmulti-input/multi-output systems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 150-160, 2009.

[16] P. S. Londhe, et al. "Robust nonlinear PID-like fuzzy logic control of a planar parallel (2PRP-PPR) manipulator," *ISA Transactions*, 63, pp. 218-232, 2016.

[17] T. L. Mai and Y. N. Wang, "Adaptive position tracking control system based on recurrent fuzzy wavelet neural networks for robot manipulators," in *Proc. the Institution of Mechanical Engineers, Part I: Journal of Systems Control Engineering*, vol. 228, no. 7, 2014, pp. 500-520.

[18] J. Z. Peng, et al., "A neural network sliding mode controller with application to robotic manipulator," in *Proc. 2006 6th World Congress on Intelligent Control and Automation, IEEE*, 2006, pp. 2101-2105.

[19] S. C. Tong, et al., "Fuzzy adaptive output feedback control for MIMO nonlinear systems," *Fuzzy Sets and Systems*, vol. 156, no. 2, pp. 285-299, 2005.

[20] V. T. Yen, et al., "Robust adaptive sliding mode control for industrial robot manipulator using fuzzy wavelet neural networks," *International Journal of Control, Automation Systems*, vol. 15, no. 6, pp. 2930-2941, 2017.

[21] J. J. Craig, "Introduction to robotics mechanics and control," *Pearson Education International*, 2005.

[22] J. K. Liu, *Intelligent Control Design and MATLAB Simulation*, Springer, 2018.

[23] P. A. Ioannou and J. Sun, "Robust adaptive control," *PTR Prentice-Hall*, pp. 75–76, 1996.

[24] A. Rezoug, B. Tondu, M. Hamerlain, M. Tadjine, "RBFNN-HOMS nonsingular terminal sliding control of n-DOF robotic manipulator," *Journal of Control Engineering and Applied Informatics (CEAI)*, vol. 18, no. 3 pp. 52-62, 2016.