

Adaptive synchronous sliding control for a robot manipulator based on neural networks and fuzzy logic

Dien Nguyen Duc, Thong Vu Viet

Faculty of Electrical Engineering, University of Economics-Technology for Industry, Ha Noi, Viet Nam

Article Info

Article history:

Received Oct 16, 2023

Revised Jan 9, 2024

Accepted Jan 12, 2024

Keywords:

Adaptive control

Fuzzy logic

Neural network

Robot manipulator

Sliding mode control

Synchronization tracking error

ABSTRACT

Robot manipulators have become important equipment in production lines, medical fields, and transportation. Improving the quality of trajectory tracking for robot hands is always an attractive topic in the research community. This is a challenging problem because robot manipulators are complex nonlinear systems and are often subject to fluctuations in loads and external disturbances. This article proposes an adaptive synchronous sliding control scheme to improve trajectory tracking performance for a robot manipulator. The proposed controller ensures that the positions of the joints track the desired trajectory, synchronize the errors, and significantly reduces chattering. First, the synchronous tracking errors and synchronous sliding surfaces are presented. Second, the synchronous tracking error dynamics are determined. Third, a robust adaptive control law is designed, the unknown components of the model are estimated online by the neural network, and the parameters of the switching elements are selected by fuzzy logic. The built algorithm ensures that the tracking and approximation errors are ultimately uniformly bounded (UUB). Finally, the effectiveness of the constructed algorithm is demonstrated through simulation and experimental results. Simulation and experimental results show that the proposed controller is effective with small synchronous tracking errors, and the chattering phenomenon is significantly reduced.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Dien Nguyen Duc

Faculty of Electrical Engineering, University of Economics-Technology for Industry

Ha Noi, Viet Nam

Email: nddien@uneti.edu.vn

1. INTRODUCTION

Robot manipulators have become necessary equipment in production lines, medical fields, and transportation [1]. Therefore, the issue of robot manipulator control research is always an attractive topic for the research community [2], [3]. The robot manipulator control problem can be divided into two types: The first is the construction of a moving trajectory, and the second is the trajectory tracking control. The robot manipulator's motion trajectory must be accurate, flexible, and intelligent. A typical trajectory construction algorithm is the rapidly-exploring random tree (RRT) algorithm [4]–[6]. In [6], the RRT algorithm was improved to help the robot manipulator operate in complex environments, especially the ability to avoid collisions during work. The second problem is the control of tracking the established trajectory. The second problem is trajectory tracking control, which is a challenging problem because the robot manipulator is a complex nonlinear system to load fluctuations and external disturbances. Typical control methods are gravity-compensated proportional derivative (PD), direct torque controller, sliding controller, and backstepping controller [7]. Sliding controllers

have been developed in many different versions, such as terminal sliding controllers, nonsingular sliding mode controllers, and fast terminal sliding mode controllers [8]. The disadvantage of the traditional sliding control method is the requirement for precise dynamics. The problem of determining the exact dynamic model of the robot manipulator is complex because the model parameters change during the working process. In [9], an adaptive integral sliding controller was proposed to improve the tracking error and reduce the influence of noise. In [10], an adaptive sliding controller was developed based on the backstepping technique to reduce the dependence on the robot manipulator's mathematical model. In [11], an adaptive sliding controller with linearized feedback was proposed for robot manipulators. In another approach, sliding controllers combined with fuzzy controllers or neural networks (NNs) to deal with unknown dynamics [12]. In [13], a robust adaptive controller was introduced based on the combination of integral sliding control, adaptive fuzzy, and disturbance observer. A decentralized adaptive fuzzy sliding controller was proposed for robot manipulators [14]. On the other hand, in [15], [16], an NN-based adaptive sliding controller was proposed for robot manipulators with model parameter uncertainty and disturbances. In addition, a robust adaptive controller based on a fuzzy-neural model was introduced in [17]–[19]. However, the above-proposed controllers only consider the local error of each joint. For robot manipulators, the error of the motion trajectory is affected by the position error of the joints. Therefore, the position error of the joints needs to be controlled synchronously to increase the accuracy of the robot manipulator's motion trajectory [20], [21]. On the other hand, because of the switching component of the sliding controller, the controller causes chattering around the sliding surface. This problem has a significant impact on the actuator. In [22], an adaptive sliding mode controller using a time delay estimation technique was proposed, where the adaptive law considers an arbitrarily small neighborhood of the sliding surface, which gives the ability to adapt quickly and reduce chattering. In [23], an adaptive synchronous sliding controller for parallel robots was proposed, in which the uncertainty components and switching components of the controller are approximated by fuzzy logic. The adaptive sliding controller combines low-pass filtering and super-convolutional algorithms used in [24] to eliminate chattering, but the algorithm only applies to a class of nonlinear systems.

From the above analysis, we propose an adaptive synchronous sliding controller (ASSC) for a robot manipulator based on NN and fuzzy logic, in which NN is used to approximate the unknown nonlinear function, and fuzzy logic is used to eliminate the chattering phenomenon. The main contributions of the article are summarized as follows: i) different from [13]–[18], [21], the proposed controller considers synchronous error to increase the accuracy of the robot's motion trajectory and make the robot operate more smoothly. The proposed controller ensures that the positions of the joints track the desired trajectory and synchronize the errors and ii) instead of using NNs as in [15], [16], [21] to approximate the function, we only use one NN, in which NN is the radial basic function (RBF). This reduces the complexity of NN, making the calculation process more efficient. Unlike [23], the proposed controller uses an NN network to approximate the completely unknown nonlinear component instead of only approximating the uncertain components. Furthermore, the proposed controller uses fuzzy logic to reduce chattering significantly compared to the controllers in [15], [16], [21], [23].

2. METHOD

2.1. Dynamic model of robot manipulator

Consider a model of an n-link robot manipulator with joint variables $\phi \in \mathbb{R}^{n \times 1}$ and the dynamic equations are described by [3], [7], [25], [26]:

$$\mathcal{M}(\phi)\ddot{\phi} + \mathcal{C}(\phi, \dot{\phi})\dot{\phi} + \mathcal{G}(\phi) + \mathcal{F}(\dot{\phi}) = \tau - \tau_d, \quad (1)$$

where $\mathcal{M}(\phi) \in \mathbb{R}^{n \times n}$, $\mathcal{C}(\phi, \dot{\phi}) \in \mathbb{R}^{n \times n}$, $\mathcal{G}(\phi) \in \mathbb{R}^{n \times 1}$, $\mathcal{F}(\dot{\phi}) \in \mathbb{R}^{n \times 1}$, $\tau_d \in \mathbb{R}^{n \times 1}$, and $\tau = [\tau_1, \tau_2, \dots, \tau_n]^T \in \mathbb{R}^{n \times 1}$ are the symmetric and positive definite inertia matrix, the Coriolis-centripetal matrix, gravity vector, the static friction vector, the disturbances vector, and the control inputs vector, respectively.

Property 1 The matrices $\mathcal{M}(\phi)$, $\mathcal{C}(\phi, \dot{\phi})$, $\mathcal{G}(\phi)$ are limited by $\bar{m}_{\min} \leq \|\mathcal{M}(\phi)\| \leq \bar{m}_{\max}$, $\|\mathcal{C}(\phi, \dot{\phi})\| \leq b_C$, $\|\mathcal{G}(\phi)\| \leq b_G$, where \bar{m}_{\min} , \bar{m}_{\max} , b_C , b_G are positive constants.

Property 2 $\dot{\mathcal{M}}(\phi) - 2\mathcal{C}(\phi, \dot{\phi})$ is skew symmetric, i.e., $x^T (\dot{\mathcal{M}}(\phi) - 2\mathcal{C}(\phi, \dot{\phi})) x = 0$ for all vectors x .

Property 3 τ_d are limited by $\|\tau_d\| \leq b_d$, where b_d is positive constant.

2.2. Synchronous tracking error dynamics

The controller design aims to control joint positions $\phi(t)$ to track desired trajectories $\phi_d(t)$, i.e., $\lim_{t \rightarrow \infty} (\phi_d(t) - \phi(t)) \rightarrow 0$, and simultaneously synchronize joint errors. First, define the position tracking errors of the joints as $e(t) = \phi_d(t) - \phi(t)$. To build a synchronous sliding controller, we define the synchronous error as $\bar{e} = [e_1 - e_2, e_2 - e_3, \dots, e_n - e_1]^T$. By determining the synchronization error, the control objective is not only to ensure $\lim_{t \rightarrow \infty} e(t) \rightarrow 0$ but also to ensure $\lim_{t \rightarrow \infty} \bar{e}(t) \rightarrow 0$. To ensure position tracking error and synchronization tracking error, we determine the cross synchronization error as (2):

$$z = e + \Lambda_1 \bar{e}, \quad (2)$$

where Λ_1 is a positive definite matrix to balance position and synchronization tracking errors. The sliding surface is given as (3):

$$s = \dot{e} + \Lambda_2 z, \quad (3)$$

where $\Lambda_2 = \Lambda_2^T > 0$. Taking the derivative of both sides of (3) and performing some transformations, we get (4):

$$\begin{aligned} \mathcal{M}\dot{s} &= \mathcal{M}(\ddot{e} + \Lambda_2 \dot{z}) = \mathcal{M}(\ddot{\phi}_d - \ddot{\phi} + \Lambda_2 \dot{z}) \\ &= \mathcal{M}(\ddot{\phi}_d + \Lambda_2 \dot{z}) - \mathcal{C}s + \mathcal{C}(\dot{\phi}_d + \Lambda_2 z) + \mathcal{G} + \mathcal{F} + \tau_d - \tau \\ &= -\mathcal{C}s - \tau + \psi + \tau_d, \end{aligned} \quad (4)$$

where $\psi = \mathcal{M}(\ddot{\phi}_d + \Lambda_2 \dot{z}) + \mathcal{C}(\dot{\phi}_d + \Lambda_2 z) + \mathcal{G} + \mathcal{F}$. Since ψ is unknown, we use an RBFNN with one hidden layer to approximate the function. Radial basis function neural network (RBFNN) is generally described as (5):

$$\psi(x) = W^T \varphi(x) + \varepsilon, \quad (5)$$

where $x = [z^T, \dot{z}^T, \phi_d^T, \dot{\phi}_d^T, \ddot{\phi}_d^T]^T$ is input vector, $\varphi(x)$ is the activation function chosen as (6),

$$\varphi_j = \exp\left(-\frac{\|x - c_i\|^2}{b_j^2}\right), \quad (6)$$

where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, m is the number of hidden layer neurons, c_i is the coordinate value of center point, b_j is the width value, ε is the approximation error, $W = [W_1, W_2, \dots, W_m]^T$ is the ideal weight vector, satisfying $\|W\|_F \leq W_m$, $\|\cdot\|_F$ is the Frobenius norm. Since the weights are unknown, the function approximation is given as (7):

$$\hat{\psi}(x) = \hat{W}^T \varphi(x), \quad (7)$$

where \hat{W} is the approximate weight. From (5) and (7), we have

$$\psi(x) - \hat{\psi}(x) = \tilde{W}^T \varphi(x) + \varepsilon, \quad (8)$$

where $\tilde{W} = W - \hat{W}$.

2.3. Design of an adaptive synchronous sliding controller

For the control objective of system (2.1.), the proposed control law is as (9):

$$\tau = \hat{\psi} + Ks + \beta \operatorname{sgn}(s), \quad (9)$$

where K is a positive definite symmetric matrix, $\beta = \varepsilon_m + b_d$, $\|\varepsilon\| \leq \varepsilon_m$, $\|\tau_d\| \leq b_d$. Substituting (9) into (4), the tracking error dynamics becomes (10),

$$\mathcal{M}\dot{s} = (-K + \mathcal{C})s + \tilde{W}^T \varphi + (\varepsilon + \tau_d - \beta \operatorname{sgn}(s)). \quad (10)$$

The parameter adjustment law of RBFNN is designed as (11),

$$\dot{\hat{W}} = F\varphi s^T - kF\|s\|\hat{W}, \tag{11}$$

where $F = F^T > 0, k > 0$. In the control law (11), the parameter β significantly affects the chattering phenomenon. If this parameter is small, the response time is slow, the chattering phenomenon is reduced, and vice versa. Therefore, this paper uses a fuzzy controller to create a change law for parameter β according to $\|s\|$. The fuzzy controller has a one-input and one-output structure, as shown in Figure 1, in which the coefficients K_1 and K_2 are preprocessing and postprocessing. The membership function of input and the membership function of output are illustrated in Figures 2 and 3. Fuzzy rules are described in Table 1; the inference method is max-min, and the defuzzification method is the central.

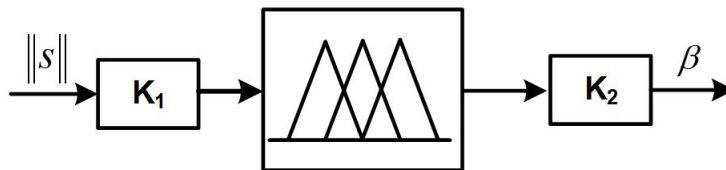


Figure 1. Fuzzy controller structure

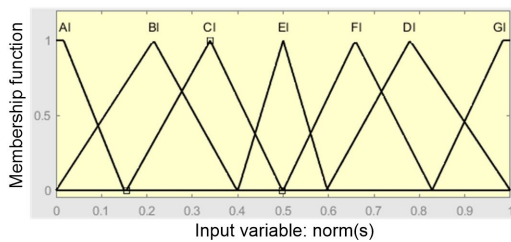


Figure 2. The membership function of input

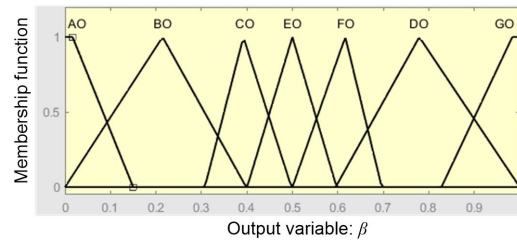


Figure 3. The membership function of output

Table 1. Fuzzy control law

Fuzzy rule	$\ s\ $	β
1	AI	AO
2	BI	BO
3	CI	CO
4	DI	DO
5	EI	EO
6	FI	FO
7	GI	GO

2.4. Analyze stability and convergence

To prove the stability of the system, we choose the Lyapunov function as (12):

$$\mathcal{L} = \frac{1}{2}s^T Ms + \frac{1}{2}tr(\tilde{W}^T F^{-1}\tilde{W}). \tag{12}$$

Differentiating \mathcal{L} , we get the result as (13),

$$\dot{\mathcal{L}} = s^T M\dot{s} + \frac{1}{2}s^T \dot{M}s + tr(\tilde{W}^T F^{-1}\dot{\tilde{W}}). \tag{13}$$

Substituting (10) and (11) into (13), we get (14),

$$\dot{\mathcal{L}} = -s^T Ks + k\|s\|tr(\tilde{W}^T(W_M - \tilde{W}_M)) + (\varepsilon + \tau_d - \beta \text{sgn}(s)). \tag{14}$$

Since $tr(\tilde{W}^T(W - \tilde{W})) = (\tilde{W}, W)_F - \|\tilde{W}\|_F^2 \leq \|\tilde{W}\|_F \|W\|_F - \|\tilde{W}\|_F^2$, we have results in (15),

$$\begin{aligned}\dot{\mathcal{L}} &\leq -\lambda_{\min}(K)\|s\|^2 + k\|s\|\left\|\tilde{W}\right\|_F\left(W_m - \left\|\tilde{W}\right\|_F\right) \\ &= -\|s\|\left(\lambda_{\min}(K)\|s\| + k\left(\left\|\tilde{W}\right\|_F - W_m/2\right)^2 - kW_m^2/4\right),\end{aligned}\quad (15)$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of the matrix. Therefore, $\dot{\mathcal{L}} < 0$ if and only if,

$$\|s\| > \frac{k_M W_m^2}{4\lambda_{\min}(K)} = \eta_1, \quad (16)$$

$$\left\|\tilde{W}\right\|_F > W_m = \eta_2. \quad (17)$$

From (16) and (17), we can see that if $\|s\|$ or $\left\|\tilde{W}\right\|_F$ exceeds the stable region, presented as the compact set η_1 or η_2 , then $\dot{\mathcal{L}} < 0$, the synchronous tracking errors or the approximation errors are pulled into the stable region. Thus, the synchronization errors and approximation errors are ultimately uniformly bounded (UUB).

3. RESULTS AND DISCUSSION

This section verifies the performance of ASSC through comparative simulation and experiment. We compare with the adaptive sliding controller (ASC) in [12]. Consider a Scorbot-ER robot manipulator [27] shown in Figures 4 and 5, where $a_1 = 0.35m$, $a_e = 0.025m$, $a_2 = 0.222m$, $a_3 = 0.222m$.

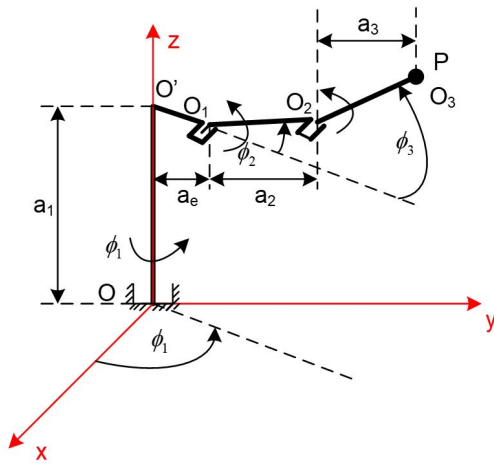


Figure 4. The scheme of Scorbot-ER

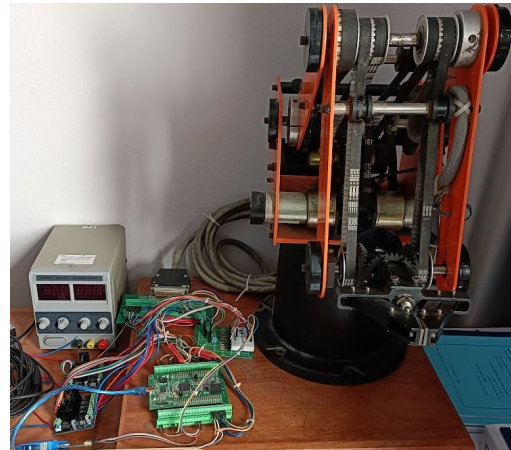


Figure 5. Scorbot-ER

The matrices of the dynamic equations are,

$$\begin{aligned}\mathcal{M}(\phi) &= \begin{bmatrix} \mathcal{M}_{11} & 0 & 0 \\ 0 & \kappa_6 & l_2\kappa_2 \cos(\phi_3 - \phi_2) \\ 0 & l_2\kappa_2 \cos(\phi_3 - \phi_2) & \kappa_7 \end{bmatrix} \\ \mathcal{C}(\phi, \dot{\phi}) &= \begin{bmatrix} \chi_1\dot{\phi}_2 + \chi_2\dot{\phi}_3 & \chi_1\dot{\phi}_1 & \chi_2\dot{\phi}_1 \\ -\chi_1\dot{\phi}_1 & 0 & -\chi_3\dot{\phi}_3 \\ -\chi_2\dot{\phi}_1 & \chi_3\dot{\phi}_2 & 0 \end{bmatrix}, \mathcal{F}(\dot{\phi}) = \begin{bmatrix} \kappa_8\dot{\phi}_1 + \kappa_{11} \operatorname{sgn}(\dot{\phi}_1) \\ \kappa_9\dot{\phi}_2 + \kappa_{12} \operatorname{sgn}(\dot{\phi}_2) \\ \kappa_{10}\dot{\phi}_3 + \kappa_{13} \operatorname{sgn}(\dot{\phi}_3) \end{bmatrix} \\ \mathcal{G}(\phi) &= [0 \quad \kappa_1 g \cos \phi_2 \quad \kappa_2 g \cos \phi_3]^T,\end{aligned}\quad (18)$$

where $\mathcal{M}_{11} = \mathcal{M}_{111} + \mathcal{M}_{112} + \mathcal{M}_{113} + \mathcal{M}_{114}$, $\mathcal{M}_{111} = 2\kappa_1 l_e \cos \phi_2$, $\mathcal{M}_{112} = 2\kappa_2(l_e + l_2 \cos \phi_2) \cos \phi_3$, $\mathcal{M}_{113} = 0.5\kappa_3 \cos(2\phi_2)$, $\mathcal{M}_{114} = 0.5\kappa_4 \cos(2\phi_3) + \kappa_5$, $\chi_1 = -(\kappa_1 l_e \sin \phi_2 + \kappa_2 l_2 \sin \phi_2 \cos \phi_3 + 0.5\kappa_3 \sin(2\phi_2))$, $\chi_2 = -(\kappa_2(l_e + l_2 \cos \phi_2) \sin \phi_3 + 0.5\kappa_4 \sin(2\phi_3))$, $\chi_3 = l_2\kappa_2 \sin(\phi_3 - \phi_2)$, $\kappa_1 = \kappa_4 = \kappa_7 = 0.006$, $\kappa_2 = 0.002$, $\kappa_3 = \kappa_5 = \kappa_6 = 0.011$, $\kappa_8 = \kappa_9 = \kappa_{10} = 0.52$, $\kappa_{11} = 0.019$, $\kappa_{12} = \kappa_{13} = 0.018$,

$\phi(0) = [0.01, 0.01, 0.01]^T$, $\dot{\phi}(0) = [0, 0, 0]^T$. The desired trajectory is given as (19),

$$P_d = \begin{bmatrix} x_{P_d} \\ y_{P_d} \\ z_{P_d} \end{bmatrix} = \begin{bmatrix} 0.36 + 0.05 \sin(0.04\pi) \\ 0.05 \cos(0.04\pi) \\ 0.4 \end{bmatrix}. \tag{19}$$

The position trajectories of the joints are obtained by the inverse kinematic equation.

3.1. Simulation

The parameters of ASSC are chosen as $\Lambda_1 = \text{diag}[1, 1, 1]$, $\Lambda_2 = \text{diag}[5, 5, 5]$, $K = \text{diag}[25, 25, 25]$, $k = 0.01$, $K_1 = 1$, $K_2 = 0.5$, $c_i = [-1.5, -1.0, -0.5, 0, 0.5, 1.0, 1.5]$, $b_i = 10$, the number of hidden layer neurons is 7, corresponding to the weight for three joints is 21, the initial weights are zeros, external disturbances $\tau_d = [-0.5, 0.5]$ Nm. The parameters of ASC are chosen similarly to ASSC. Perform simulation for 50s, with sampling time 0.01s. At the 30th second, change the load to double the initial load. The weight update results of ASSC are illustrated in Figure 6, and the function approximation results are shown in Figure 7, indicating that the weights converge quickly and the approximation error reaches 10^{-5} .

Figures 8 and 9 show the tracking errors of the controllers, showing that the tracking performance of ASSC is better than ASC. The trajectory tracking results in the workspace of the entire simulation process are illustrated as Figure 10 and after the algorithm converges as Figure 11. The control torque of the controllers is presented in Figures 12 and 13. At the time of load change, the weight values change as shown in Figure 6 so that the control torque changes large enough as shown in Figure 12 to ensure ASSC tracking error performance. On the other hand, ASSC converges to the overall system error faster than ASC.

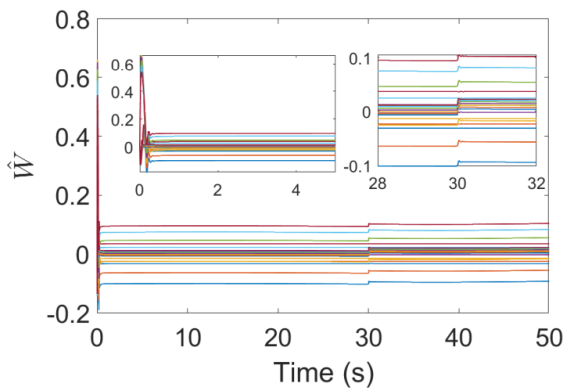


Figure 6. Convergence of weights

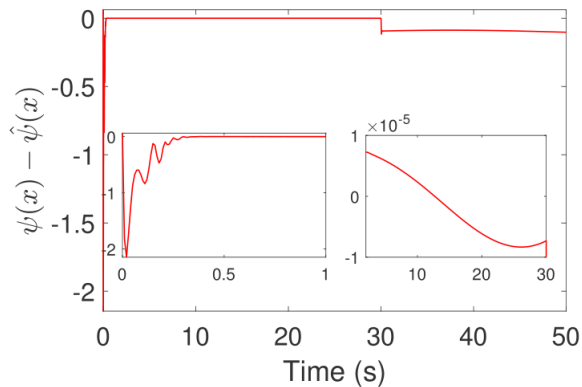


Figure 7. Function approximation error

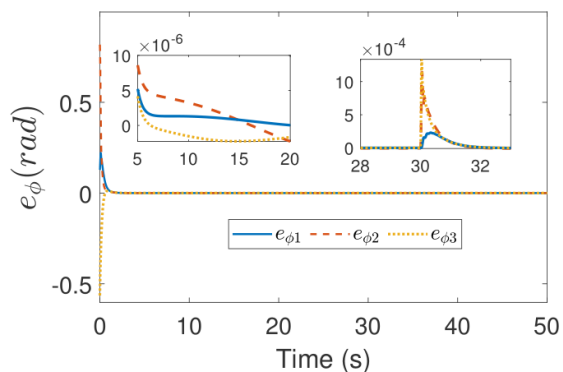


Figure 8. Tracking errors of ASSC

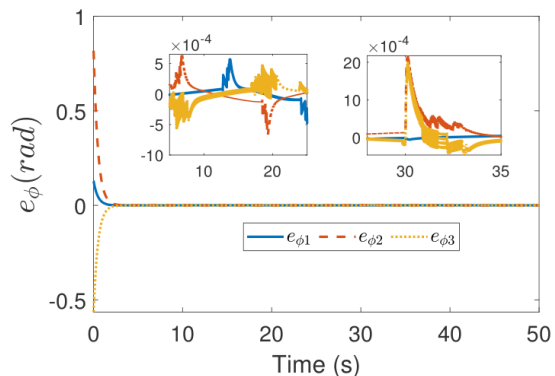


Figure 9. Tracking errors of ASC

In addition, the results in Figures 12 and 13 show that ASSC has significantly eliminated the chattering phenomenon. For ASC, the tracking error and torque fluctuate enormously. Through simulation results, the effectiveness of ASSC has been verified.

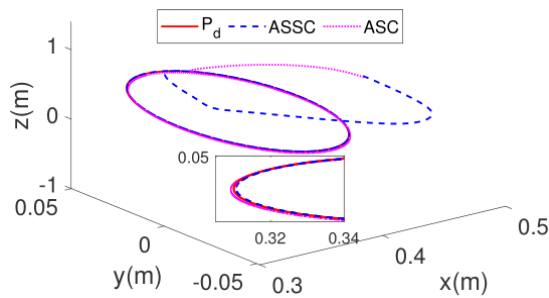


Figure 10. Trajectory tracking in the workspace (the entire simulation process)

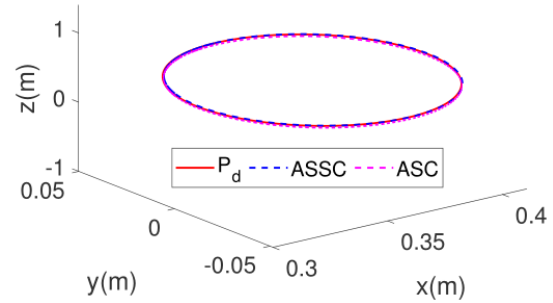


Figure 11. Trajectory tracking in the workspace after the algorithms converge

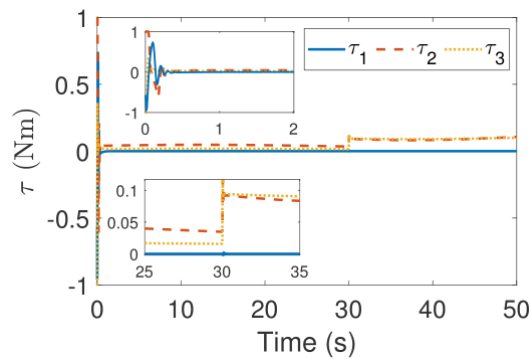


Figure 12. Control inputs of ASSC

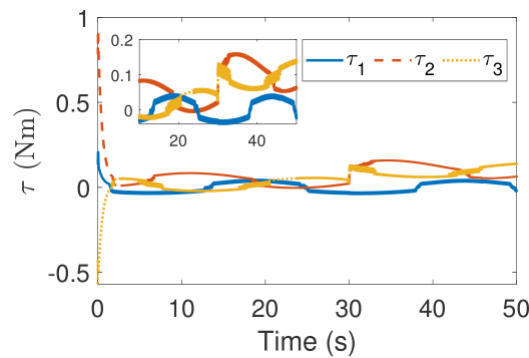


Figure 13. Control inputs of ASC

3.2. Experiment

We build an experiment on Scorbot-ER, illustrated in Figure 5, where the joints use a direct current (DC) motor with a gearbox attached with a gear ratio of 127.7:1 and an encoder. We use the STM32F407 microcontroller to implement the controllers and drive L298 to communicate between the microcontroller and motor. Data from the microcontroller is sent to the computer via universal asynchronous receiver/transmitter (UART) protocol.

To make the network training process fast, we use the convergence weights in the simulation for the experiment. Experimental results of the tracking error of the controllers are shown in Figure 14 and

Figure 15, indicating that ASSC has a smaller tracking error than ASC. Specifically, ASSC's tracking error does not exceed 5.10^{-3} rad for the first joint, 5.10^{-3} rad for the second joint, and 0.02 rad for the third joint. Figure 16 is the result of tracking the trajectory of the controllers in the workspace after the algorithms converge. Experimental results have shown the effectiveness of ASSC. Thus, simulation and experimental results show that the proposed controller is effective with small synchronous tracking errors, and the chattering phenomenon is significantly reduced.

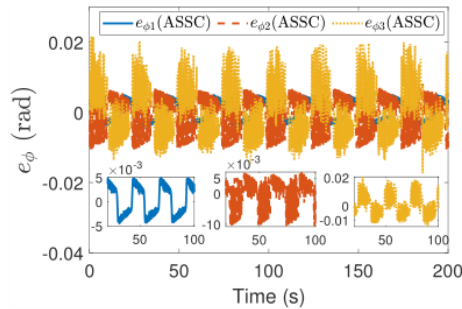


Figure 14. The tracking errors of ASSC in the experiment

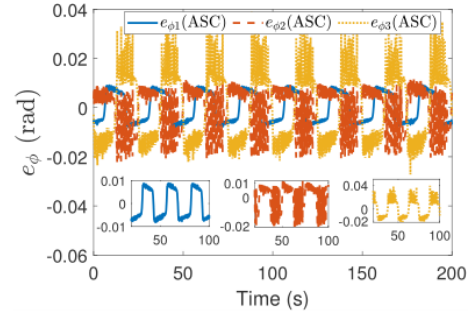


Figure 15. The tracking errors of ASC in the experiment

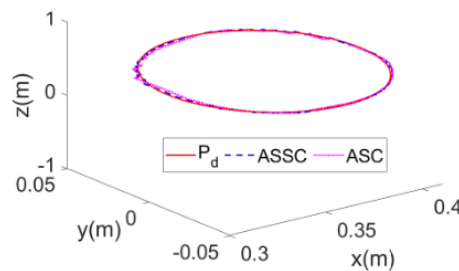


Figure 16. Experimental results: trajectory tracking in the workspace after the algorithms converge

4. CONCLUSION

The article proposed an adaptive synchronous sliding controller for a robot manipulator. The proposed algorithm ensures tracking errors for the robot manipulator, eliminating chattering. The unknown dynamics are approximated online by NN, and the chattering phenomenon is eliminated by fuzzy logic. According to the extended Lyapunov technique, the synchronization tracking and approximation errors of NN are UUB stable. The effectiveness of the proposed algorithm has been shown through comparative simulation and experimental results. Simulation and experimental results show that the proposed controller is effective with small synchronous tracking errors, and the chattering phenomenon is significantly reduced. Our subsequent work is to develop the algorithm in the workspace.





REFERENCES

- [1] S. Liu, L. Wang, and R. X. Gao, "Cognitive neuroscience and robotics: advancements and future research directions," *Robotics and Computer-Integrated Manufacturing*, vol. 85, 2024, doi: 10.1016/j.rcim.2023.102610.
- [2] W. He, Z. Li, and C. L. P. Chen, "A survey of human-centered intelligent robots: Issues and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017, doi: 10.1109/JAS.2017.7510604.
- [3] S. Zeghloul, M. A. Laribi, and J.-P. Gazeau, "Robotics and mechatronics," in *2010 3rd International Symposium on Resilient Control Systems*, Aug. 2010, pp. 76–78, doi: 10.1109/ISRCS.2010.5603174.
- [4] S. Wu, G. Liu, Y. Zhang, and A. Xue, "A fast and accurate compound collision detector for RRT motion planning," *Robotics and Autonomous Systems*, vol. 167, 2023, doi: 10.1016/j.robot.2023.104484.
- [5] J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu, and L. Wang, "An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling," *Journal of Computational Science*, vol. 67, Mar. 2023, doi: 10.1016/j.jocs.2022.101937.
- [6] Q. Gao, Q. Yuan, Y. Sun, and L. Xu, "Path planning algorithm of robot arm based on improved RRT* and BP neu-





- ral network algorithm,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, 2023, doi: 10.1016/j.jksuci.2023.101650.
- [7] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice 2/e*, vol. 31, no. 5. CRC Press, 2004.
- [8] J. Liu and X. Wang, “Adaptive sliding mode control for mechanical systems,” *Advanced Sliding Mode Control for Mechanical Systems*, pp. 117–135, 2011, doi: 10.1007/978-3-642-20907-9_6.
- [9] J. Lee, P. H. Chang, and M. Jin, “Adaptive integral sliding mode control with time-delay estimation for robot manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6796–6804, 2017, doi: 10.1109/TIE.2017.2698416.
- [10] Y. Wang, Z. Zhang, C. Li, and M. Buss, “Adaptive incremental sliding mode control for a robot manipulator,” *Mechatronics*, vol. 82, Apr. 2022, doi: 10.1016/j.mechatronics.2021.102717.
- [11] A. Elmogy and W. Elawady, “An adaptive continuous sliding mode feedback linearization task space control for robot manipulators,” *Ain Shams Engineering Journal*, 2023, doi: 10.1016/j.asej.2023.102284.
- [12] J. Liu, *Intelligent control design and MATLAB simulation*. Springer, 2017.
- [13] M. Van and S. S. Ge, “Adaptive fuzzy integral sliding-mode control for robust fault-tolerant control of robot manipulators with disturbance observer,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 1284–1296, 2021, doi: 10.1109/TFUZZ.2020.2973955.
- [14] X. Yin, L. Pan, and S. Cai, “Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators,” *Robotics and Computer-Integrated Manufacturing*, vol. 72, Dec. 2021, doi: 10.1016/j.rcim.2019.101884.
- [15] T. Sun, H. Pei, Y. Pan, H. Zhou, and C. Zhang, “Neural network-based sliding mode adaptive control for robot manipulators,” *Neurocomputing*, vol. 74, no. 14–15, pp. 2377–2384, 2011, doi: 10.1016/j.neucom.2011.03.015.
- [16] V. T. Yen, W. Y. Nan, and P. Van Cuong, “Robust adaptive sliding mode neural networks control for industrial robot manipulators,” *International Journal of Control, Automation and Systems*, vol. 17, no. 3, pp. 783–792, 2019, doi: 10.1007/s12555-018-0210-y.
- [17] V. T. Yen, W. Y. Nan, P. Van Cuong, N. X. Quynh, and V. H. Thich, “Robust adaptive sliding mode control for industrial robot manipulator using fuzzy wavelet neural networks,” *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2930–2941, Dec. 2017, doi: 10.1007/s12555-016-0371-5.
- [18] N. X. Quynh, W. Y. Nan, and V. T. Yen, “Design of a robust adaptive sliding mode control using recurrent fuzzy wavelet functional link neural networks for industrial robot manipulator with dead zone,” *Intelligent Service Robotics*, vol. 13, no. 2, pp. 219–233, 2020, doi: 10.1007/s11370-019-00300-y.
- [19] O. Bachir and A.-F. Zoubir, “Adaptive neuro-fuzzy inference system based control of puma 600 robot manipulator,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 2, no. 1, 2011, doi: 10.11591/ijece.v2i1.116.
- [20] Y. Li, Q. Zheng, and L. Yang, “Design of robust sliding mode control with disturbance observer for multi-axis coordinated traveling system,” *Computers and Mathematics with Applications*, vol. 64, no. 5, pp. 759–765, 2012, doi: 10.1016/j.camwa.2011.11.053.
- [21] Q. V. Doan, T. D. Le, Q. D. Le, and H. J. Kang, “A neural network-based synchronized computed torque controller for three degree-of-freedom planar parallel manipulators with uncertainties compensation,” *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, 2018, doi: 10.1177/1729881418767307.
- [22] J. Baek, M. Jin, and S. Han, “A new adaptive sliding-mode control scheme for application to robot manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628–3637, Jun. 2016, doi: 10.1109/TIE.2016.2522386.
- [23] T. D. Le and Q. V. Doan, “Fuzzy adaptive synchronized sliding mode control of parallel manipulators,” in *ACM International Conference Proceeding Series*, 2018, pp. 102–107, doi: 10.1145/3191477.3199060.
- [24] K. Shao, J. Zheng, C. Yang, F. Xu, X. Wang, and X. Li, “Chattering-free adaptive sliding-mode control of nonlinear systems with unknown disturbances,” *Computers and Electrical Engineering*, vol. 96, 2021, doi: 10.1016/j.compeleceng.2021.107538.
- [25] J. Baillieul, *Introduction to ROBOTICS mechanics and control*, vol. 32, no. 5. United States of America: Pearson Educacion, 2004.
- [26] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. Oxford: CRC Press, 2004.
- [27] M. Szuster and P. Gierlak, “Approximate dynamic programming in tracking control of a robotic manipulator,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, 2016, doi: 10.5772/62129.

BIOGRAPHIES OF AUTHORS



Dien Nguyen Duc     graduated from electrical engineering technology, majoring in Automation at the University of Economics–Technology for Industry, Viet Nam. He received a master’s in control and automation engineering from the University of Transport Viet Nam in 2014. From 2012 to the present, he has been a lecturer in the Department of Control and Automation, Faculty of Electrical Engineering, University of Economics–Technology for Industry, Ha Noi, Viet Nam. His main research directions are intelligent control, robotics, adaptive optimal control, sustainable optimal control, and ADP. He can be contacted at email: nddien@uneti.edu.vn.



Thong Vu Viet     was born in 1990. He graduated as an electrical engineering and technology engineer, majoring in Automation at the University of Economic and Industrial Technology. Received a master’s degree in control engineering and automation from the University of Transport and Communications in 2015. He works at the Department of Electrical Engineering, University of Economics–Technology for Industries. The primary research: robust control, mobile robots, neural networks, IoT, and artificial intelligence. He can be contacted at email: vtvtuong@uneti.edu.vn.