EAI Endorsed Transactions
on Sustainable Manufacturing and Renewable Energy Research Article **EAI.EU**

Advanced Delta Robot Control Using Adaptive Neural PID with Recurrent Fuzzy Neural Network Modeling

H.Q. Nhan[1*], V.B. Dung[1] and P.B.D. Loc[1]

[1]Department of Mechatronics Engineering, Faculty of Mechanical Engineering, Ho Chi Minh city University of Technology (HCMUT), 268 Ly Thuong Kiet, Dien Hong Ward, Ho Chi Minh City, Viet Nam

Abstract

This study presents an adaptive control method for a 3-degree-of-freedom parallel Delta robot using a neuron PID controller combined with a recurrent fuzzy neural network (RFNN) identifier. Due to the nonlinear kinematics and dynamics, coupling, and load changes of the Delta robot, traditional PID controllers often do not ensure optimal control quality, thereby requiring adaptive intelligent control techniques [1], [2]. In the proposed model, the PID is represented as a linear neuron capable of self-updating the parameters K_p, K_i, Kd_d based on the Jacobian information estimated online by the RFNN identifier, inheriting the backpropagation learning principle in the recurrent neuro-fuzzy system [3]. MATLAB simulation results show that the response time is improved, the steady-state error is eliminated, and the system maintains its stability when the load changes, which is consistent with previous studies on neural network-based adaptive PID for nonlinear systems [4]–[6]. This method contributes to affirming the effectiveness of combining PID neuron and RFNN for precise control of parallel robots.

Keywords: Delta robot, Single Neuron PID, Recurrent Fuzzy Neural Network (RFNN), Robot model, Trajectory tracking

Received on 26 November 2025, accepted on 27 January 2026, published on 04 February 2026

doi: 10.4108/eetsmre.11113

*Corresponding author. Email: nhan.haquyen@hcmut.edu.vn

1. Introduction

Parallel robots are one of the topics that have been developing rapidly in recent years. With flexible structures and advantages in speed, force and accuracy, delta robots have become popular and widely used in industry [11]. One of the studies is to use PD and LQR controllers to compare their effectiveness in tracking the trajectory [5][6][9]. In addition, in controlling delta robots, many research groups use non-linear PID controllers to help robots eliminate noise and have better performance [4][5][7][8][9]. Not only that, the technique of using self-tuning controllers in PID helps robots to self-correct errors during operation, but in return, this is one of the most difficult techniques in control [1-10]. PID controllers have been successfully developed for robots from 2-6 DOF, however, PID controllers are often effective with fixed input parameters. Continuously changing parameters such as load, external conditions, etc., the PID controller does not achieve the desired performance. Therefore, neural networks and fuzzy logic are applied to improve the control of Delta robots [1-4] [7-8][10]. RFNN controllers and identifiers have been developed and applied to non-linear systems. During the control process, the RFNN identifier can estimate the sensitivity of the object, which is called Jacobian information. This information is used to train the RFNN controller online. Therefore, by using RFNN, the control technique can flexibly adjust the online parameters to suit the control conditions.

Unlike conventional single-neuron PID controllers whose parameter adaptation relies solely on tracking error signals, the proposed approach integrates a recurrent fuzzy neural network (RFNN) identifier to estimate the system Jacobian online. This Jacobian information reflects the real-time sensitivity of the Delta robot dynamics and is incorporated into the learning rule of the single-neuron PID controller.

As a result, the controller can adapt not only to tracking errors but also to variations in nonlinear dynamics, coupling effects, and payload changes, which are typical challenges in high-speed Delta robot applications. Furthermore, the proposed controller is implemented as a coordinated MIMO structure, where each robot arm is regulated by its own neuron PID trained by a corresponding RFNN identifier, making it suitable for strongly nonlinear parallel manipulators.

2. The Proposed Approach

2.1. The model of Delta robot

In this study, the Delta robot model is established based on commonly adopted simplifying assumptions in the literature [7]–[9]. The links BiDi are represented by two lumped mass points located at Bi and Di, connected by rigid and weightless bars.

These assumptions significantly reduce model complexity while maintaining sufficient accuracy for control design and performance evaluation. Although the inertia of the links and vibration effects may introduce minor dynamic discrepancies during high-speed motion, such modeling errors are compensated by the adaptive learning capability of the proposed neuron PID controller combined with the RFNN identifier. Therefore, the simplified model remains suitable for validating the effectiveness of the adaptive control strategy.

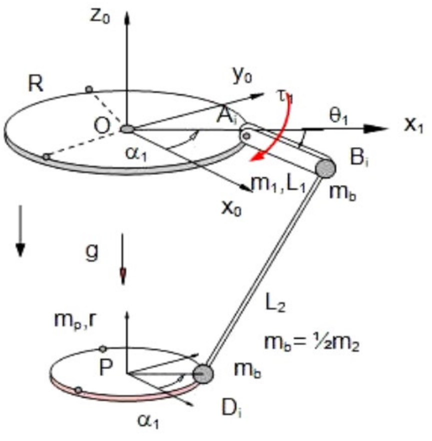

Figure 1. Robot delta Urus [9]

The set of robot includes:

$$q = [\theta_1 \theta_2 \theta_3 x_p y_p z_p] \tag{1}$$

2.2. Establishing the linking equation

According to Figure 1, we determine the linking equation for point B₁ and D₁:

$$l^2 - (r_{D_1} - r_{B_1})^T (r_{D_1} - r_{B_1}) = 0 \tag{2}$$

Where:

r_{B_1}: Position of vector OB₁ in the Oxyz coordinate.

r_{D_1}: Position of vector OD₁ in the Oxyz coordinate.

The vector equation r_{B_1} in Ox₁y₁z₁ coordinate:

$$r_{B_1} = r_{A_1} + U_{A_1B_1} = [R + l_1 \cos\theta_1 \quad 0 \quad -l_1 \sin\theta_1]^T \tag{3}$$

Where:

$r_{A_1} = [R \quad 0 \quad 0]^T$

$U_{A_1B_1} = [l_1 \cos\theta_1 \quad 0 \quad -l_1 \sin\theta_1]^T$

We determine the z - rotation matrix:

$$A_{z,\alpha_1} = \begin{bmatrix} \cos\alpha_1 & -\sin\alpha_1 & 0 \\ \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Thus, we determine the r_{B_1} in Oxyz coordinate:

$$\begin{aligned} r_{B_1} &= \begin{bmatrix} \cos\alpha_1 & -\sin\alpha_1 & 0 \\ \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R + l_1\cos\theta_1 \\ 0 \\ -l_1\sin\theta_1 \end{bmatrix} \\ &= \begin{bmatrix} R\cos\alpha_1 + l_1\cos\alpha_1\cos\theta_1 \\ R\sin\alpha_1 + l_1\sin\alpha_1\cos\theta_1 \\ -l_1\sin\theta_1 \end{bmatrix} \end{aligned} \tag{5}$$

The r_{D_1} in Oxyz coordinate:

$$\begin{aligned} r_{D_1} = r_P + U_{PU_1} &= \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \begin{bmatrix} \cos\alpha_1 & -\sin\alpha_1 & 0 \\ \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} x_p + r\cos\alpha_1 \\ y_p + r\sin\alpha_1 \\ z_p \end{bmatrix} \end{aligned} \tag{6}$$

Combine the (6) and (5), we have:

$$r_{D_1} - r_{B_1} = \begin{bmatrix} (R-r)\cos\alpha_1 + l_1\cos\alpha_1\cos\theta_1 - x_p \\ (R-r)\sin\alpha_1 + l_1\sin\alpha_1\cos\theta_1 - y_p \\ -l_1\sin\theta_1 - z_p \end{bmatrix} \tag{7}$$

Substituting the (7) into (2), we get the equation for first link. Similar to second and third links, we get the link equation as follows:

$$\begin{aligned} f_1 = l_2^2 - \left[(R-r)\cos\alpha_1 + l_1\cos\alpha_1\cos\theta_1 - x_p\right]^2 \\ - \left[(R-r)\sin\alpha_1 + l_1\sin\alpha_1\cos\theta_1 \right. \\ \left. - y_p\right]^2 - \left[l_1\sin\theta_1 + z_p\right]^2 = 0 \\ f_2 = l_2^2 - \left[(R-r)\cos\alpha_2 + l_1\cos\alpha_2\cos\theta_2 - x_p\right]^2 \\ - \left[(R-r)\sin\alpha_2 + l_1\sin\alpha_2\cos\theta_2 \right. \\ \left. - y_p\right]^2 - \left[l_1\sin\theta_2 + z_p\right]^2 = 0 \\ f_3 = l_2^2 - \left[(R-r)\cos\alpha_3 + l_1\cos\alpha_3\cos\theta_3 - x_p\right]^2 \\ - \left[(R-r)\sin\alpha_3 + l_1\sin\alpha_3\cos\theta_3 \right. \\ \left. - y_p\right]^2 - \left[l_1\sin\theta_3 + z_p\right]^2 = 0 \end{aligned}$$

2.3. The kinetic and potential energy of Delta robot

The kinetic energy of robot include the kinetic energy of AiBi stages, kinetic energy of mass mb is set as B1 and kinetic energy of moving table and mb masses:

$$T = \frac{1}{2}\left(I_{1y} + m_b l_1^2\right)\left(\dot{\theta_1}^2 + \dot{\theta_2}^2 + \dot{\theta_3}^2\right) + \frac{1}{2}\left(m_p + 3m_b\right)\left(\dot{x_P}^2 + \dot{y_P}^2 + \dot{z_P}^2\right) \quad (9)$$

The potential energy of robot:

$$\pi = -gl_1\left(\frac{1}{2}m_1 + m_b\right)(\sin\theta_1 + \sin\theta_2 + \sin\theta_3) - g\left(m_p + 3m_b\right)z_P \quad (10)$$

2.4. Establish the motion differential equations for the Delta robot.

We use Lagrange factor to establish the motion differential equation with the following form:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_k}\right) - \frac{\partial T}{\partial q_k} = Q_k - \sum_{i=1}^{r} \lambda_i \frac{\partial f_i}{\partial q_k}, (k = 1, 2, \dots, m) \quad (11)$$

Where

q_k: is the extrapolation coordinates of the robot

f_i: Linking equation

Q_k: Extrapolation force

Substituting the kinetic and potential energy into (11), we have the motion equations of delta robot:

$$(I_{ly} + m_b l_1^2)\ddot{\theta_1} = gl_1(\frac{1}{2}m_1 + m_b)\cos\theta_1 + \tau_1 - 2\lambda_1 l_1\left(\begin{array}{c} \sin\theta_1(R - r) - \cos\alpha_1 \sin\theta_1 x_p \\ - \sin\alpha_1 \sin\theta_1 y_p - \cos\theta_1 z_p \end{array}\right) \quad (12)$$

$$(I_{ly} + m_b l_1^2)\ddot{\theta_2} = gl_1(\frac{1}{2}m_1 + m_b)\cos\theta_2 + \tau_2 - 2\lambda_1 l_1\left(\begin{array}{c} \sin\theta_2(R - r) - \cos\alpha_2 \sin\theta_2 x_p \\ - \sin\alpha_2 \sin\theta_2 y_p - \cos\theta_2 z_p \end{array}\right) \quad (13)$$

$$(I_{ly} + m_b l_1^2)\ddot{\theta_3} = gl_1(\frac{1}{2}m_1 + m_b)\cos\theta_3 + \tau_3 - 2\lambda_3 l_1\left(\begin{array}{c} \sin\theta_3(R - r) - \cos\alpha_3 \sin\theta_3 x_p \\ - \sin\alpha_3 \sin\theta_3 y_p - \cos\theta_3 z_p \end{array}\right) \quad (14)$$

$$(m_p + 3m_b)\ddot{x_p} = -2\lambda_1(\cos\alpha_1(R - r) + l_1\cos\alpha_1\cos\theta_1 - x_p) - 2\lambda_2(\cos\alpha_2(R - r) + l_1\cos\alpha_2\cos\theta_2 - x_p) - 2\lambda_2\left(\cos\alpha_3(R - r) + l_1\cos\alpha_3\cos\theta_3 - x_p\right) \quad (15)$$

$$(m_p + 3m_b)\ddot{y_p} = -2\lambda_1(\sin\alpha_1(R - r) + l_1\sin\alpha_1\cos\theta_1 - y_p) - 2\lambda_2(\sin\alpha_2(R - r) + l_1\sin\alpha_2\cos\theta_2 - y_p) - 2\lambda_2\left(\sin\alpha_3(R - r) + l_1\sin\alpha_3\cos\theta_3 - y_p\right) \quad (16)$$

$$(m_p + 3m_b)\ddot{z_p} = -(m_p + 3m_b)g + 2\lambda_1(z_p + l_1\sin\theta_1) + 2\lambda_2(z_p + l_1\sin\theta_2) + 2\lambda_3\left(z_p + l_1\sin\theta_3\right) \quad (17)$$

2.5. Designing the Single Neuron PID

Single Neuron PID is an adaptive PID controller, in which the parameters Kp, Ki, Kd are considered as the weights of a neuron and are automatically adjusted according to the learning rule to optimize the error.

Controller structure

With 3 inputs: Kp, Ki, Kd, the PID controller set as:

$$u(k) = u(k - 1) + K_p\Delta e1 + K_i\Delta e2 + K_d\Delta e3 \quad (18)$$

Where:

$$\Delta e1 = e(k)$$

$$\Delta e2 = \int_0^\infty e(k)dk$$

$$\Delta e3 = \frac{de(k)}{dk}$$

Controller training

The goal of training the single neuron PID controller is to adjust the network's weight set w1i (i=1,2,3) to minimize the cost function

$$E(k) = \frac{1}{2}e^2(k) = \frac{1}{2}[y_{ref}(k) - y(k)]^2 \quad (19)$$

Where:

$y_{ref}(k)$: The reference signal

$y(k)$: System response

To adjust the weight set w_{1i} (i=1,2,3), the gradient descent method was applied

$$K_p = w_{11}(k + 1) = w_{11}(k) + \Delta w_{11}(k) \quad (20)$$

$$K_i = w_{12}(k + 1) = w_{12}(k) + \Delta w_{12}(k) \quad (21)$$

$$K_d = w_{13}(k + 1) = w_{13}(k) + \Delta w_{13}(k) \quad (22)$$

Where $\Delta w_{11}(k), \Delta w_{12}(k)\ and\ \Delta w_{13}(k)$ are considered by:

$$\Delta w_{11}(k) = \eta^{K_p}\left(-\frac{\partial E(k)}{\partial w_{11}(k)}\right) = -\eta^{K_p}e(k)\frac{\partial y(k)}{\partial u(k)}\Delta e1 \quad (23)$$

$$\Delta w_{12}(k) = \eta^{K_i}\left(-\frac{\partial E(k)}{\partial w_{12}(k)}\right) = -\eta^{K_i}e(k)\frac{\partial y(k)}{\partial u(k)}\Delta e2 \quad (24)$$

$$\Delta w_{13}(k) = \eta^{K_d}\left(-\frac{\partial E(k)}{\partial w_{13}(k)}\right) = -\eta^{K_d}e(k)\frac{\partial y(k)}{\partial u(k)}\Delta e3 \quad (24)$$

Where:

$\eta^{K_p}, \eta^{K_i}, \eta^{K_d}$ are learning rate constant.

$\frac{\partial y(k)}{\partial u(k)}$: is the respone sensitivity, which is the Jacobian information.

2.6. Designing the RFNN Identifier

RFNN Identifier structure

The RFNN identifier has 4 layers, with an input layer of 2 nodes, a fuzzy layer of 10 nodes, a fuzzy rule class of 25 nodes, and an output layer with 1 node.
Layer 1: Input consists of 2 nodes that convey the input values to the next layer.

$$O_i^1(k) = x_i^1(k) + \theta_i^1(k) O_i^k(k-1), (i = 1,2) \quad (25)$$

Where:
θ_i^1: connection weight at current k.
$x_1^1(k) = u(k)$: The current control signal.
$x_2^1(k) = y(k-1)$: The past output of response

Layer 2: Fuzzy layer: This layer consists of (2x5) nodes, each node representing a related function of the Gaussian form with mean value m_{ij} and standard deviation σ_{ij}

$$O_{ij}^2(k) = \exp\left\{-\frac{\left(O_i^1(k) - m_{ij}\right)^2}{\sigma_{ij}}\right\}, (i = 1,2; j = 1,2,\ldots,5) (26)$$

Layer 3: Law layer: this layer consists of (5x5) nodes. The output of the q node in this class is determined as follows:

$$O_q^3(k) = \prod_i O_{iq_i}^2(k), (i = 1,2,\ldots,5; q_i = 1,2,\ldots,5) \quad (27)$$

Layer 4: Output layer: This layer includes 1 linear neuron with the defined output as follows:

$$O_i^4(k) = \sum_j w_{ij}^4 O_j^3(k), \ (i = 1, j = 1,2,\ldots,25) \quad (28)$$

RFNN Identifier training

The goal of an online training algorithm for RFNN identifier is to adjust the weighting sets of the network and the parameters of the fuzzy class dependent functions to achieve the minimum value of cost function

$$E(k) = \frac{1}{2}[y(k) - y_m(k)]^2$$
$$= \frac{1}{2}[y(k) - O_1^4(k)]^2 \quad (29)$$

The weight of each RFNN network layer is updated as follows:

$$w_{ij}^4(k+1) = m_{ij}(k) +$$
$$\eta^m \sum_k e(k) w_{ik}^4 \, O_k^3 \frac{2\left[O_{ij}^1(k) - m_{ij}\right]}{\left(\sigma_{ij}\right)^2} \quad (30)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) +$$
$$\eta^\sigma \sum_k e(k) w_{ik}^4 \, O_k^3 \frac{2\left[O_{ij}^1(k) - m_{ij}\right]^2}{\left(\sigma_{ij}\right)^3} \quad (31)$$

$$\theta_i^1(k+1) = \theta_i^1(k) +$$
$$\eta^\theta \sum_k e(k) w_{ik}^4 \, O_k^3 \frac{(-2)\left[O_{ij}^1(k) - m_{ij}\right] O_{ij}^1(k-1)}{\left(\sigma_{ij}\right)^2} \quad (32)$$

The RFNN identifier must also estimate Jacobian information for online training of the single neuron PID controller. Jacobian information is determined as follows:

$$\frac{\partial y(k)}{\partial u(k)} = \frac{\partial O_1^4}{\partial u}$$
$$= \sum_q w_{ij}^4 \cdot \left\{\sum_s \frac{\partial O_1^3}{\partial O_{qs}^2} \cdot \frac{(-2)\left[O_{ij}^1(k) - m_{ij}\right]}{\left(\sigma_{ij}\right)^2}\right\} \quad (33)$$

3. Simulation and Results Simulation Parameters

Table 1. Delta robot specifications

Symbol	Value	Unit
L_1	0.3	m
L_2	0.8	m
R	0.26	m
r	0.04	m
α_1	0	rad/s
α_2	$\frac{2\pi}{3}$	rad/s
α_3	$\frac{4\pi}{3}$	rad/s
m_l	0.42	kg
m_b	0.2	kg
m_p	0.75	kg

Simulation Results

The desired trajectory of Delta robot as follow:

$$\begin{aligned} x_d &= 0.17\sin(2\pi t) + 0.3 \\ y_d &= 0.17\sin(2\pi t) + 0.2 \\ z_d &= -0.7 \end{aligned} \qquad (34)$$

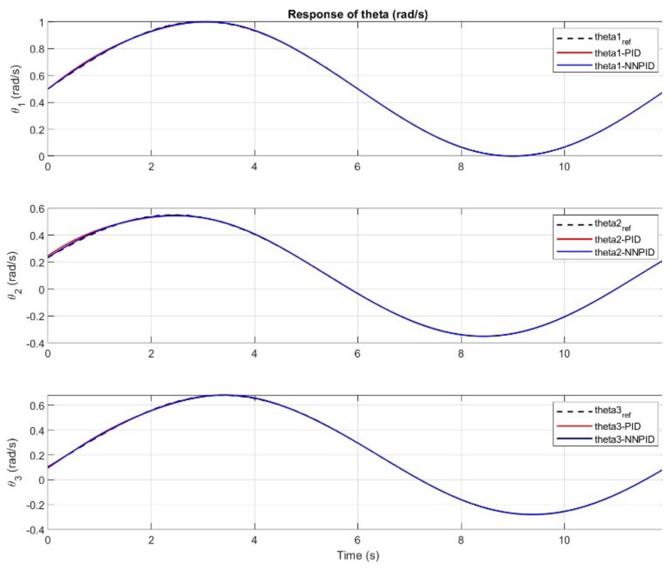

Figure 2. Compare the response of thetha between PID and Single neuron PID

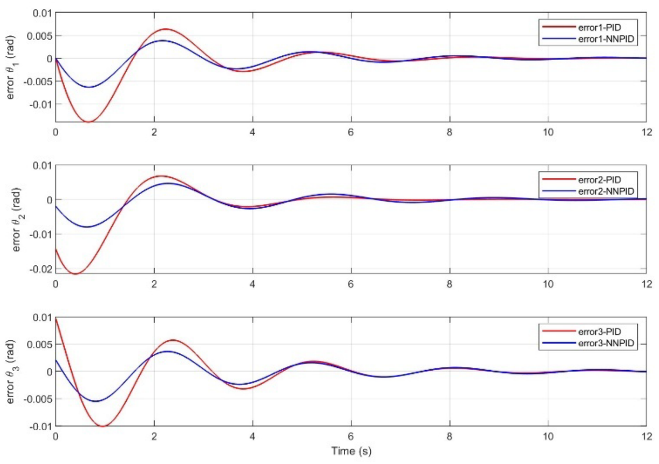

Figure 3. Compare the error of thetha between PID and Single neuron PID

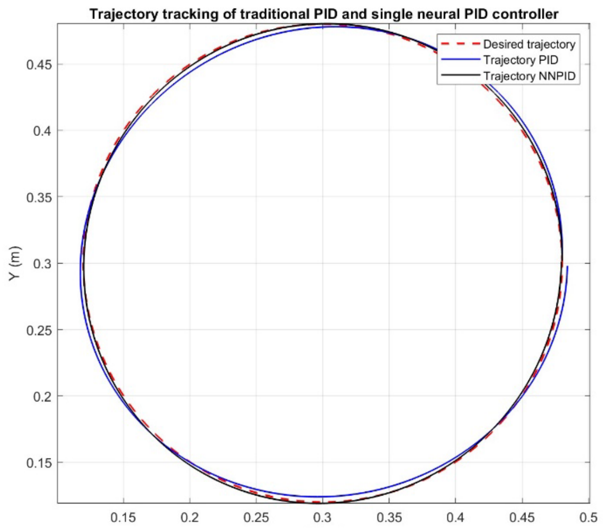

Figure 4. Compare the trajectory tracking between PID and Single neuron PID

Table 2. System comparison

Indicators	PID	Neuron PID
Settling time	0.4 ± 0.001 (s)	0.3 ± 0.001 (s)
Overshoot	1.61%	0.7%
Rising time	1.965 (s)	1.933 (s)

4. Conclusion

This study employed single-neuron PID controllers combined with RFNN-based identifiers to control a 3-DOF delta robot, which is a nonlinear MIMO system. Each arm of the robot was regulated by its own single-neuron PID controller, trained online using Jacobian information provided by the corresponding RFNN identifier. Simulation results confirm that both the controllers and identifiers can be updated in real time during operation, and the system achieves better performance than conventional PID control (Table IV). The proposed method demonstrates stable behavior and fast response in simulation. Future work will involve implementing and testing the controllers on an actual delta robot.

References

[1] Le Minh Thanh, L. H. T., Loc, P. T., & Nguyen, C. N. (2020). Delta robot control using single neuron PID algorithms based on recurrent fuzzy neural network identifiers. International Journal of Mechanical Engineering and Robotics Research, 9(10), 1411-1418.

[2] Le Minh Thanh, L. H. T., Tung, P. T., Pham, C. T., & Nguyen, C. N. (2021). Evaluating the quality of intelligent controllers for 3-DOF delta robot control. International Journal of Mechanical Engineering and Robotics Research, 10(10), 542-552.

[3] Minh Nguyet, N. T., & Ba, D. X. (2023). A neural flexible PID controller for task-space control of robotic manipulators. Frontiers in Robotics and AI, 9, 975850.

[4] Yang, Q., Zhang, F., & Wang, C. (2024). Deterministic learning-based neural PID control for nonlinear robotic systems. IEEE/CAA Journal of Automatic Sinica, 11(5), 1227-1238.

[5] Zhao, A., Toudeshki, A., Ehsani, R., Viers, J. H., & Sun, J. Q. (2024). Evaluation of neural network effectiveness on sliding mode control of Delta robot for trajectory tracking. Algorithms, 17(3), 113.

[6] Urrea, C., Domínguez, C., & Kern, J. (2024). Modelling, design and control of a 4-arm delta parallel manipulator employing type-1 and interval type-2 fuzzy logic-based techniques for precision applications. Robotics and Autonomous Systems, 175, 104661.

[7] Wu, M., & Corves, B. (2025). Adaptive optimal trajectory tracking control of delta robots with online parameter estimation. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 09596518251322230.

[8] Thi, Y. V., Yao, N. W., Huu, H. N., Van, C. P., & Manh, T. N. (2025). Neural network-based adaptive robust fractional PID control for robotic systems. Journal of Robotics and Control (JRC), 6(4), 1790-1801.

[9] Saleem, D., Shamseldin, M. A., & Eleashy, H. (2025). Design and Control of Delta Robot (Mini Review). Future Engineering Journal, 5(1).

[10] Tamizi, M. G., Kashani, A. A. A., Azad, F. A., Kalhor, A., & Masouleh, M. T. (2022). Experimental study on a novel simultaneous control and identification of a 3-dof delta robot using model reference adaptive control. *European journal of control*, *67*, 100715.

[11] BENCHORA, A., & METCHAT, A. A. (2020). Conception et réalisation d'un robot à structure parallèle (Doctoral dissertation, M MALTI Abed).