

Modeling and Control of a 6-DOF Manipulator: A Comparison of PID, Fuzzy-PID and NN-RBF-PID Controllers for Circular Trajectory Tracking under Model Mismatch

Dao Duc Thang, Vu Quang Huy, Duy-Trung Nguyen, Huy-Hoang Hoang,
N. – K. Nguyen*

Faculty of Automation, Artificial Intelligence and Intelligent Systems, Electric Power University, Hanoi,
Vietnam

*Corresponding Author: nkn.research@gmail.com

ABSTRACT: This paper presents a comparison of three feedback controllers for a six-degree-of-freedom (6-DOF) elbow manipulator with a spherical wrist. The comparison is made under identical conditions on a circular tracking task (two revolutions of radius 0.10 m in a horizontal plane). Forward and inverse kinematics were calculated in closed form using the Denavit-Hartenberg convention. The dynamic model was obtained symbolically from the Euler-Lagrange formulation and checked for four structural properties: symmetry and positive definiteness of the inertia matrix, skew-symmetry of $\dot{M} - 2C$ and conservation of mechanical energy. The three controllers were a classical PID, a self-tuning fuzzy PID (Fuzzy-PID) and a PID enhanced by an online-trained radial basis function network (NN-RBF-PID). All three used the same baseline gains and the same model-based feedforward compensation. Thus, any difference in performance can be attributed to the feedback law alone. The MATLAB/Simulink simulations included the 15% mass error in the compensation model and Coulomb friction in the joints. The root mean square (RMS) radial error was 3.137 mm for PID, 1.873 mm for Fuzzy-PID and 0.828 for NN-RBF-PID. Compared with PID, the maximum deviation in the vertical (out-of-plane) direction was reduced by 87.1% and the integral absolute error (IAE) was reduced by 84.6% without the increase of joint torque and control energy. The sensitivity analysis carried out over six scenarios shows that the RMS error increased by a factor of 2.2 for NN-RBF-PID and by a factor of 4.8 for PID when the mass error increased from 5% to 25%. Increasing the proportional gain by a factor of five reduced the PID error to 0.722 mm, while NN-RBF-PID at the same gain reduced the error by an additional 55%.

NOMENCLATURE

Symbol	Description	Unit
a_i, d_i	D-H parameters: link length, link offset	m
b	Gaussian width of the RBF network	
$C(q, \dot{q})$	Coriolis–centrifugal matrix	kg·m ² /s
e	Joint tracking error, $e = q_d - q$	rad
$e_{\max}$	Maximum joint error during drawing	deg
$e_N, \dot{e}_N$	Error and error-rate normalization factors	rad, rad/s
F_c	Coulomb friction torque	N·m
$G(q)$	Gravity torque vector	N·m
h	Hidden-layer output of the RBF network	
IAE	Integral of absolute error	rad·s
K_p, K_i, K_d	Proportional, integral and derivative gains	
$M(q)$	6×6 inertia matrix	kg·m ²
m_i	Mass of link i	kg
p	Pen-tip position	m
$q, \dot{q}, \ddot{q}$	Joint angle, velocity, acceleration	rad, rad/s, rad/s ²
R	Radius of the circle to be drawn	m
$r_{\text{RMS}}, r_{\text{max}}$	RMS and maximum radial error	mm
s	Sliding variable, $s = \dot{e} + \lambda e$	rad/s
W	Weight matrix of the RBF network	
$z_{\max}$	Maximum vertical pen-tip deviation	mm

Greek letter

α_i, θ_i	Link twist angle, joint angle (D-H)	rad
γ	Learning rate of the RBF network	
ζ	Damping ratio	
η	Time-scaling polynomial	
K_{p2}, K_{i2}, K_{d2}	Gains of the fuzzy corrections	

λ	Sliding-surface coefficient	$1/s$
ξ	Normalized time	
σ	Weight-drift suppression coefficient	
τ	Joint torque vector	$N \cdot m$
φ, ω	Sweep angle and sweep angular velocity	$rad, rad/s$
ω_n	Natural frequency	rad/s

Subscripts

c	Circle center; link center of mass
d	Desired (reference) value
ff, fb	Feedforward, feedback
f	Friction
n	Normalized to $[-1, 1]$
nn	Neural network
0	Baseline / initial value

Date of Submission: 09-10-2026

Date of acceptance: 20-10-2026

I. INTRODUCTION

A common industrial robot arm is the six-degree-of-freedom (6-DOF) elbow manipulator with a spherical wrist. The robot has six revolute joints that allow the end-effector to be positioned and oriented anywhere within the reachable workspace, subject to joint limits and kinematic singularities. In this configuration, the first three joints determine the position of the wrist center and the last three joints determine the end-effector orientation. The intersecting wrist-joint axes provide the kinematic decoupling allowing the closed-form inverse kinematic solutions [1], [6], [7]. The end-effector must follow a given trajectory when performing operations like seam welding, laser cutting, adhesive dispensing, and drawing, rather than just moving to a desired pose. ISO 9283 [22] therefore distinguishes between path accuracy and pose accuracy. Trajectory tracking is a challenging problem because manipulator dynamics are nonlinear and have a configuration-dependent inertia matrix, inter-joint coupling, and unknown inertial and friction parameters. In industrial applications, independent joint PID controllers, often with gravity compensation or dynamic feedforward compensation, are still widely used [18], [21]. With fixed PID gains residual errors due to imperfect compensation need to be compensated by feedback whose corrective action depends on the resulting tracking errors.

The literature relevant to this study can be roughly divided into two categories: manipulator modeling and traditional control [1], [6]–[8], [14]–[16], [19], [21] and intelligent and adaptive control [2]–[5], [9]–[13], [17], [20], [24]–[30]. Trinh Quang Vinh et al. [1] have introduced a systematic approach of Denavit – Hartenberg (D – H) modeling, kinematics and manipulator dynamics based on Lagrange and Newton–Euler formulations. Huynh Thai Hoang [2] discussed about fuzzy logic, multilayer perceptron (MLP) and radial basis function (RBF) networks, genetic algorithms and neural control of robot manipulators. In [6]–[8] it is shown that the matrix $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric if the Coriolis and centrifugal matrix $C(q, \dot{q})$ is constructed from the Christoffel symbols. This structural property forms the basis for stability analysis of adaptive and neural control techniques [14], [16], [21].

Intelligent and adaptive tracking control has been studied in several works. An adaptive neural robust controller using sliding mode control and Lyapunov-based adaptation laws was proposed by Vu Thi Yen et al. [3] to improve the tracking performance of a 3-DOF manipulator in comparison with adaptive fuzzy and backstepping controllers. Le Tien Dung and Bui Huu Thanh [4] developed an improved neural sliding-mode controller for a 2-DOF manipulator considering the drive-motor dynamics. Nguyen Dinh Tu et al. [5] used an RBF network in a model reference adaptive control scheme for a ball-and-beam system. Zhu et al. [9] proposed fuzzy adaptive model predictive control for visual servoing of 6-DOF manipulators. In [10], Barhaghtalab et al. developed a hybrid controller based on an adaptive neuro-fuzzy inference system (ANFIS) combined with PID control, with a Lyapunov-based convergence analysis. In [11] the inverse kinematics problem was solved by artificial neural networks.

In more recent times, a variety of neural, fuzzy and fractional-order control strategies have been explored. Song et al. [24] proposed an adaptive robust controller based on RBF network for 6-DOF manipulator and tested by simulations and experiments. Zhang and Zhao [25] proposed a combined dynamic surface control scheme and RBF network for the flexible-joint manipulator. In [26], Oleiwi et al. employed the bat algorithm to optimize hybrid neural-network and fractional-order PID controllers for a three-link manipulator. Abdulameer and Mohamed [27] presented the comparison of four integer-order and fractional-order fuzzy PID structures for a two-link manipulator. Hamzah et al. [28] realized a Mamdani fuzzy controller

on a microcontroller for a 6 DOF arm. Eltayeb et al. [29] employed a genetic algorithm for tuning conventional and fractional-order PID controllers. Pizarro-Lerma et al. [30] combined sectorial fuzzy feedback with adaptive neural feedforward control to a 2-DOF arm. Relevant methodological bases are fuzzy PID gain tuning based on error information [12], universal approximation properties of RBF networks [13] and neural compensation combination with PD or PID feedback [14], [20]. To guaranty boundedness of the adaptive weights, the σ -modification method [15] can be used under the premises of the related stability analysis.

However, the progress made is frequently hindered by variations in robot models, reference trajectories, tuning procedures, compensation schemes, and performance indicators across the literature. Additional benchmark cases of increased PID gains or improved model-based compensation are also useful to discriminate the benefits of intelligent adaptation from the benefits of conventional controller tuning. Hence, this paper compares the PID, the self-tuning fuzzy PID (Fuzzy-PID) and the RBF neural-network-assisted PID (NN-RBF-PID) controllers under matched conditions. In each comparison scenario the controllers use the same robot model, circular drawing trajectory, feedforward compensation, nominal PID gains and prescribed model mismatch. The only difference is the controller-specific feedback or adaptive correction mechanism.

The main contributions are as follows:

- (i) A controlled comparison framework with matched plant, trajectory, compensation, and nominal tuning conditions,
- (ii) six control scenarios designed to examine the effects of PID gain selection, model mismatch, online learning, and feedback, and
- (iii) quantitative verification of the kinematic and dynamic models before the evaluation of the controllers, including the kinematic consistency, the skew-symmetry of $\dot{M} - 2C$, and energy conservation under conservative conditions. The entire framework was implemented in MATLAB/Simulink, using the Symbolic Math Toolbox and the Fuzzy Logic Toolbox [23].

II. MODELING AND CONTROLLER DESIGN

Fig. 1 illustrates the task proposed in this study. From a pen-up position, the pen tip reached the circle edge in 2 s and then drew two consecutive circles of radius $R = 0.10$ m centered at $(0.35, 0, 0.25)$ m in a horizontal plane. The trajectory was mapped to joint space by inverse kinematics, and the Euler–Lagrange model served as the “real robot”.

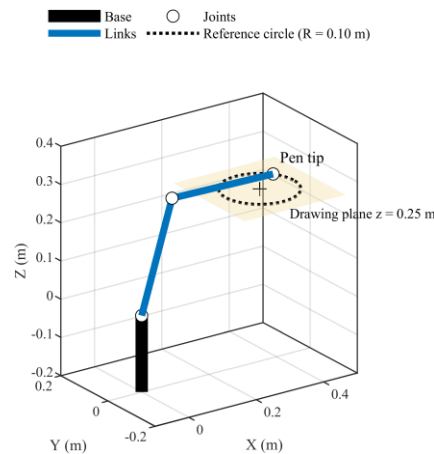


Fig. 1 The 6-DOF robot and the circle to be drawn in the plane $z = 0.25$ m

ROBOT KINEMATICS

The robot has six revolute joints in an elbow arrangement (Fig. 2). Joint 1 rotates about the vertical axis; joints 2 (shoulder) and 3 (elbow) have parallel horizontal axes and form a planar two-link mechanism with lengths $a_2 = 0.30$ m and $d_4 = 0.30$ m; the axes of joints 4, 5 and 6 intersect at the wrist center. The tool length was assumed to be zero ($d_6 = 0$), so the pen tip coincides with the wrist center. Link dimensions and masses are assumed values typical of a small laboratory manipulator (reach about 0.6 m, total moving mass 10.5 kg). Frames were assigned by the D-H convention [1], [6] with the parameters in Table 1; the transformation between adjacent links and that of the last link relative to the base are:

$${}^{i-1}A_i = R_z(\theta_i) T_z(d_i) T_x(a_i) R_x(\alpha_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^0T_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 = \begin{bmatrix} R & p \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2)$$

Table (1). D-H parameters of the robot

Link i	θ_i	d_i (m)	a_i (m)	α_i
1	q_1	0	0	$+90^\circ$
2	q_2	0	$a_2 = 0.30$	0
3	q_3	0	0	$+90^\circ$
4	q_4	$d_4 = 0.30$	0	-90°
5	q_5	0	0	$+90^\circ$
6	q_6	0	0	0

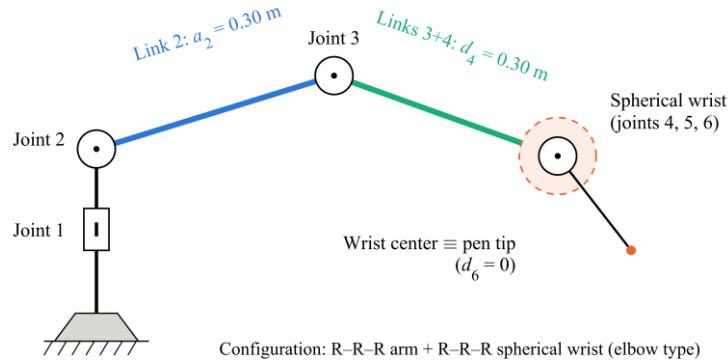


Fig. 2 Kinematic diagram of the 6-DOF manipulator

Substituting the parameters of Table 1 into (1) and multiplying the first three matrices, with the shorthand $c_i = \cos q_i$, $s_i = \sin q_i$, $c_{23} = \cos(q_2 + q_3)$, $s_{23} = \sin(q_2 + q_3)$, gives the position of the pen tip (which coincides with the wrist center):

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_1(a_2 c_2 + d_4 s_{23}) \\ s_1(a_2 c_2 + d_4 s_{23}) \\ a_2 s_2 - d_4 c_{23} \end{bmatrix} \quad (3)$$

It follows from (3) that the distance from the origin to the wrist center depends only on q_3 :

$$r^2 + p_z^2 = a_2^2 + d_4^2 + 2a_2 d_4 \sin q_3, \quad r = \sqrt{p_x^2 + p_y^2} \quad (4)$$

Ignoring joint limits, the wrist center can reach a sphere of radius 0.6 m. The trajectory lies 0.35 m to 0.51 m from the origin, well inside the workspace and about 0.1 m from its outer boundary, so the arm never fully stretches out (boundary singularity) while drawing. Following Pieper's theorem, the inverse kinematics was split into wrist-center position and wrist orientation subproblems [6], [8]. For the "elbow-up" configuration ($\cos q_3 > 0$), the first three joint angles are:

$$q_1 = \text{atan2}(p_y, p_x) \quad (5)$$

$$s_3 = \frac{r^2 + p_z^2 - a_2^2 - d_4^2}{2a_2 d_4}, \quad c_3 = +\sqrt{1 - s_3^2}, \quad q_3 = \text{atan2}(s_3, c_3) \quad (6)$$

$$q_2 = \text{atan2}(k_1 p_z + k_2 r, k_1 r - k_2 p_z), \quad k_1 = a_2 + d_4 s_3, \quad k_2 = d_4 c_3 \quad (7)$$

Along the trajectory considered here, s_3 varies only between -0.31 and 0.47 , so the solvability condition $|s_3| \leq 1$ is always satisfied. Once q_1, q_2, q_3 are known, the orientation that the wrist must produce is ${}^3R_6 = ({}^0R_3)^T R_d = [r_{ij}]$. Matching this with the Z-Y-Z Euler structure of ${}^3R_6$ and choosing $\sin q_5 > 0$ (non-flipped wrist) yields:

$$q_5 = \text{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right), \quad q_4 = \text{atan2}(r_{23}, r_{13}), \quad q_6 = \text{atan2}(r_{32}, -r_{31}) \quad (8)$$

The pen orientation was fixed at $R_d = {}^0R_3(\text{center}) \cdot R_y(0.5 \text{ rad})$, keeping q_5 within 21° – 34° , far from the wrist singularity. The forward (FK) and inverse (IK) kinematics were cross-checked by the closed-loop test

FK(IK(p_d)) at 5501 reference points: the maximum position error was $1.8 \cdot 10^{-16}$ m and the maximum orientation-matrix error $5.7 \cdot 10^{-16}$, i.e., 64-bit machine precision, with no branch switching at any point.

DYNAMIC MODEL

The dynamic model was derived with the Euler–Lagrange method [1], [6], with the Lagrangian equal to the total kinetic energy K minus the total potential energy P :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i, \quad L = K - P, \quad i = 1, \dots, 6 \quad (9)$$

$$K = \sum_{i=1}^6 \left(\frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} I_i \omega_i^2 \right), \quad v_{ci} = J_{ci}(q) \dot{q}, \quad P = \sum_{i=1}^6 m_i g z_{ci}(q) \quad (10)$$

where $J_{ci} = \partial p_{ci} / \partial q$ is the translational Jacobian and z_{ci} is the height of the center of mass of link i . For tractability, each link was modeled as a slender rod with $I_i = m_i l_{ci}^2 / 3$ and cross-rotational terms between links were neglected; since all three controllers act on the same model, this does not bias the comparison. Link parameters are listed in Table 2. Equation (9) was expanded with the Symbolic Math Toolbox [23], separated according to (11) and exported automatically as MATLAB functions:

$$M_{ij} = \frac{\partial \tau_i}{\partial \ddot{q}_j}, \quad G = \tau_i|_{\dot{q}=0, \ddot{q}=0}, \quad C_{ij} = \frac{1}{2} \frac{\partial (\tau - M\ddot{q} - G)_i}{\partial \dot{q}_j} \quad (11)$$

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + \tau_f(\dot{q}) = \tau + \tau_{ext} \quad (12)$$

where τ_{ext} is an external disturbance (zero in all scenarios). Coulomb friction was smoothed by a steep tanh to avoid the discontinuity at $\dot{q} = 0$ (with a slope factor of 50, the friction reaches 96% of the Coulomb value at a joint velocity of only 0.04 rad/s):

$$\tau_{f,i} = F_{c,i} \tanh(50 \dot{q}_i) \quad (13)$$

Table (2). Dynamic parameters of the robot (assumed values)

Link	1	2	3	4	5	6
m_i (kg)	3.0	2.5	2.0	1.5	1.0	0.5
l_{ci} (m)	0.10	0.15	0.10	0.15	0.05	0.05
I_i (kg·m ²)	0.0100	0.0188	0.0067	0.0113	0.0008	0.0004
$F_{c,i}$ (N·m)	0.30	0.30	0.25	0.15	0.15	0.10

The model was checked at 400 random states ($q, \dot{q}$) against three theoretical properties: $M(q)$ was exactly symmetric; along the trajectory its smallest and largest eigenvalues were about $3 \cdot 10^{-4}$ kg·m² and 1.48 kg·m², both positive; and for $N = \dot{M} - 2C$, the largest value of $|N + N^T|$ was $9.2 \cdot 10^{-9}$, merely the numerical differentiation error. The strongest test was energy conservation: in 2 s of free motion ($\tau = 0$, no friction), kinetic and potential energy exchanged within ± 46 J while the total energy drifted by at most $2.9 \cdot 10^{-9}$ J. Gravity dominates the required torque (about 20.7 N·m at joint 2 and 11.5 N·m at joint 3), so a 15% mass mismatch leaves up to about 3 N·m uncompensated at joint 2; this is the main “problem” the feedback part must solve.

REFERENCE TRAJECTORY

The approach segment ($0 \leq t < T_{ic} = 2$ s) moves the pen tip in a straight line from the pen-up point $p_0 = (0.39, 0, 0.28)$ m to the drawing start point $p_s = (0.45, 0, 0.25)$ m, with a fifth-order polynomial time scaling [19]; the circle is parameterized by the sweep angle $\varphi(t)$:

$$p(t) = p_0 + \eta(\xi) (p_s - p_0), \quad \eta(\xi) = 10\xi^3 - 15\xi^4 + 6\xi^5, \quad \xi = t/T_{ic} \quad (14)$$

$$p(t) = \begin{bmatrix} x_c + R \cos \varphi(t) \\ y_c + R \sin \varphi(t) \\ z_c \end{bmatrix}, \quad (x_c, y_c, z_c) = (0.35, 0, 0.25) \text{ m} \quad (15)$$

For continuous velocity and acceleration at the junctions, the sweep angle has three phases: acceleration over $T_r = 0.5$ s, uniform rotation at $\omega = 2\pi/4$ rad/s, and a symmetric deceleration. In the acceleration phase, with $\xi = t'/T_r$ (t' measured from the start of drawing):

$$\varphi(t') = \omega T_r (2.5 \xi^4 - 3 \xi^5 + \xi^6), \quad \dot{\varphi}(t') = \omega (10 \xi^3 - 15 \xi^4 + 6 \xi^5) \quad (16)$$

The total sweep angle is exactly 4π (two laps); drawing lasts 8.5 s, the total simulation time is $T_{stop} = 2 + 8.5 + 0.5 = 11$ s, and the maximum pen-tip speed is $\omega R \approx 0.157$ m/s. At each instant, the inverse kinematics block solved for $t - \Delta t$, t , $t + \Delta t$ with $\Delta t = 1$ ms, and the desired joint velocities and accelerations were obtained by central differences.

CONTROLLER DESIGN

All three controllers apply to the robot the sum of a feedforward torque τ_{ff} and a feedback torque τ_{fb} :

$$\tau = \tau_{ff} + \tau_{fb}, \quad \tau_{ff} = \hat{M}(q_d) \ddot{q}_d + \hat{C}(q_d, \dot{q}_d) \dot{q}_d + \hat{G}(q_d) \quad (17)$$

The hatted matrices use the compensation-model masses $\hat{m}_i$ (85% of the true masses in the baseline scenario) and no friction. Since τ_{ff} depends on the reference trajectory, not on the measured state, it adds no feedback loop and leaves the closed-loop stability unchanged [21]. The error is defined throughout as $e = q_d - q$, $\dot{e} = \dot{q}_d - \dot{q}$.

PID controller. The feedback torque was computed per joint with the parallel structure [18], where K_p , K_i , K_d are 6×6 diagonal matrices. If each joint is approximated by a pure inertia $M_{ij}s^2$, the approximate natural frequency and damping ratio of joint j are given by (19):

$$\tau_{fb} = K_p e + K_i \int e dt + K_d \dot{e} \quad (18)$$

$$\omega_{n,j} \approx \sqrt{K_{p,j}/M_{j,j}}, \quad \zeta_j \approx \frac{K_{d,j}}{2\sqrt{K_{p,j}M_{j,j}}} \quad (19)$$

The gains give the heavily loaded joints (1, 2, 3) a large K_p to counteract the uncompensated gravity, the damping ratio satisfies $\zeta \geq 1$ to avoid overshoot distorting the line, and small gains at the wrist. The final values (Table 3) are also the baseline gains of the other two controllers.

Table (3). Baseline PID gains and approximate characteristics of each joint at the initial configuration

Quantity	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
K_p	120	120	100	40	40	20
K_i	5	5	5	2	2	1
K_d	25	25	20	6	6	3
$M_{ij}(q_0)$ (kg·m ²)	0.636	1.080	0.342	0.0459	0.0050	0.0004
ω_n (rad/s)	13.7	10.5	17.1	29.5	89.4	219
ζ	1.43	1.10	1.71	2.21	6.71	16.4

Fuzzy-PID controller. This controller adapts the PID gains to the current error state [2], [12]. For each joint, corrections ΔK_p , ΔK_i , ΔK_d are inferred from the error and its derivative normalized to $[-1, 1]$, $e_n = \text{sat}(e/e_N)$, $\dot{e}_n = \text{sat}(\dot{e}/\dot{e}_N)$, with $e_N = [0.20, 0.10, 0.10, 0.15, 0.15, 0.30]$ rad and $\dot{e}_N = [2.0, 2.0, 2.0, 1.5, 1.5, 2.5]$ rad/s. The effective gains are the baseline gains plus the corrections, clipped to a safe range:

$$\begin{aligned} K_p &= \text{clip}(K_{p0} + \kappa_p \Delta K_p, 0.1K_{p0}, 5K_{p0}) \\ K_i &= \text{clip}(K_{i0} + \kappa_i \Delta K_i, 0, 10K_{i0}) \\ K_d &= \text{clip}(K_{d0} + \kappa_d \Delta K_d, 0.1K_{d0}, 3K_{d0}) \end{aligned} \quad (20)$$

with $\kappa_p = 1.5K_{p0}$, $\kappa_i = 8K_{i0}$, $\kappa_d = 1.5K_{d0}$. Each input and output has five triangular fuzzy sets NB, NS, ZE, PS, PB peaking at $-1, -0.5, 0, 0.5$ and 1 . Each inference system has $5 \times 5 = 25$ rules and uses Mamdani inference [17] (min for AND and implication, max aggregation, centroid defuzzification). In the ΔK_p table, the gain rises when $|e_n|$ or $|\dot{e}_n|$ is large and is ZE at the center; ΔK_i peaks when both inputs are near zero and falls for large errors to prevent integral windup; ΔK_d rises when $\dot{e}_n$ is near zero to damp small oscillations. For speed, the three fuzzy systems were precomputed on a 21×21 grid over $[-1, 1]^2$ and bilinearly interpolated at run time, with an accepted deviation from direct evaluation of 10^{-3} .

NN-RBF-PID controller. The discrepancy between the real robot and the compensation model (under-compensated gravity and inertia, friction) is an unknown nonlinear function of the state. By the universal approximation theorem [13], a sufficiently large RBF network can approximate it; NN-RBF-PID therefore keeps the baseline PID and adds a network torque whose weights are updated online during operation [14], [20]:

$$\tau_{fb} = K_p e + K_i \int e dt + K_d \dot{e} + u_{nn}, \quad u_{nn,j} = \text{sat}_{\bar{u}_j}(W_j^T h_j) \quad (21)$$

Each joint has its own 2–9–1 network. The input is $x = [e_n, \dot{e}_n]^T$ (normalized as in the fuzzy controller); the hidden layer has nine Gaussian neurons with centers c_k on a uniform 3×3 grid with coordinates $\{-1, 0, 1\}$ and width $b = 0.8$; the output is a single linear neuron, giving 54 weights in total. The output is limited by $\bar{u} = [20, 20, 15, 5, 5, 3]$ N·m. The network learns online from $W(0) = 0$ using the sliding variable $s_j = \dot{e}_j + \lambda_j e_j$, $\lambda = [8, 8, 8, 10, 10, 12]$, with a gradient law plus a σ -modification term [15]:

$$h_k(x) = \exp\left(-\frac{\|x - c_k\|^2}{2b^2}\right), \quad k = 1, \dots, 9 \quad (22)$$

$$\dot{W}_j = \gamma h_j s_j - \sigma W_j, \quad \gamma = 20, \quad \sigma = 0.1, \quad W_j \in [-60, 60] \quad (23)$$

When $s_j > 0$ (the joint lags the reference), the weights of the current state region grow and increase u_{nn} ; the term $-\sigma W$ pulls the weights gently toward zero to prevent drift. A law of the form (23) follows from the Lyapunov function $V = \frac{1}{2}s^T M(q)s + (1/2\gamma)\Sigma \tilde{W}_i^T \tilde{W}_i$ and the skew-symmetry of $\dot{M} - 2C$; the standard result in [14] guarantees that s and $\tilde{W}$ are uniformly ultimately bounded. As the controller also includes an integral term and

output saturation, the stability of this closed loop was verified by simulation.

SIMULATION SETUP

The Simulink diagram (Fig. 3) was scripted with `add_block`, `add_line` and `set_param`; it comprises a trajectory generator, an inverse kinematics block and three parallel branches, PID, FUZZY and NN, each with its own robot model but the same feedforward compensation. It was solved with `ode45` (maximum step 1 ms, $T_{\text{stop}} = 11$ s). The baseline scenario used: compensation-model masses $\hat{m} = 0.85m$ (15% mismatch); Coulomb friction $F_c = [0.30, 0.30, 0.25, 0.15, 0.15, 0.10]$ N·m; $\tau_{\text{ext}} = 0$; $q(0) = q_d(0)$, $\dot{q}(0) = 0$, $\int e(0) = 0$, $W(0) = 0$.

Performance was assessed with five metrics, where $\mathcal{T}$ is the set of time instants in the drawing interval $[2, 10.5]$ s and N is the number of samples in $\mathcal{T}$; only IAE is computed over the whole simulation time:

$$\Delta r(t) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - R, \quad r_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{t \in \mathcal{T}} \Delta r(t)^2}, \quad r_{\text{max}} = \max_{t \in \mathcal{T}} |\Delta r(t)| \quad (24)$$

$$z_{\text{max}} = \max_{t \in \mathcal{T}} |z(t) - z_c|, \quad e_{\text{max}} = \max_{t \in \mathcal{T}, j} |e_j(t)|, \quad \text{IAE} = \int_0^{T_{\text{stop}}} \sum_{j=1}^6 |e_j(t)| dt \quad (25)$$

Besides the baseline (scenario 1), six control scenarios were designed, each changing one factor: (2) K_p (and κ_p) multiplied by 5, to check whether the neural network beats a high-gain PID; (3) and (4) $\hat{m} = 0.95m$ and $0.75m$, to assess robustness; (5) $\gamma = 0$, to test the role of learning; (6) $K_p = K_i = K_d = 0$, feedforward only; and (7) $\hat{m} = m$ without friction, i.e., perfect compensation.

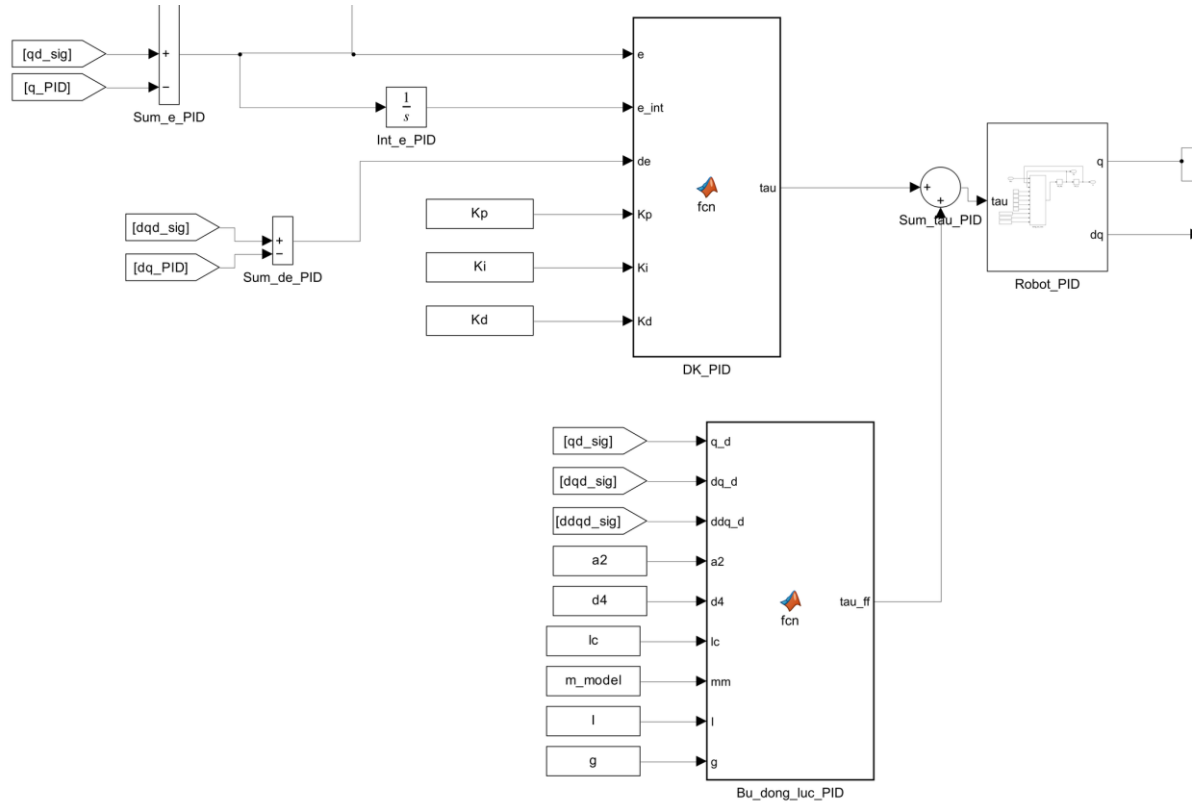


Fig. 3 PID branch of the Simulink model *Mo_Hinh_Ve_Tron.slx*: controller block, dynamic feedforward compensation and robot model (the Fuzzy-PID and NN-RBF-PID branches have the same structure and differ only in the controller block)

III. RESULTS AND DISCUSSION

All data and plots in this section come from MATLAB/Simulink (R2024b) simulations of *Mo_Hinh_Ve_Tron.slx*. Unlogged quantities (joint torques, control energy, fuzzy gains, network torques) were recomputed from the stored signals q , e , W , uniformly resampled at 2 ms.

TRACKING PERFORMANCE IN THE BASELINE SCENARIO

In the XY plane (Fig. 4), the three paths form distinct layers around the reference, with PID outermost; where the deviation is largest (right side, near the drawing start point), the PID pen tip strayed outward by about

6 mm. The radial errors of PID and Fuzzy-PID (Fig. 5) oscillated with exactly the 4 s drawing period, a systematic, configuration-dependent error typical of under-compensated gravity. The PID radial error ranged from -2.3 mm to 6.2 mm; for Fuzzy-PID the highest peak fell to 4.65 mm in the first lap and 3.67 mm in the second; for NN-RBF-PID the error stayed between -0.1 mm and 1.5 mm, without large oscillations at the drawing period.

The vertical deviation separated the controllers most clearly. With PID, the pen tip had dropped about 17 mm below the drawing plane by $t = 2$ s and stayed between -5 mm and -17 mm throughout drawing: the arm “sagged” because the feedforward supplied only 85% of the gravity torque, and PID with baseline gains was too weak to lift it fully. Fuzzy-PID cut the maximum deviation to 12.6 mm at the start of drawing, then gradually raised the pen tip, oscillating around the plane in the second lap. NN-RBF-PID kept the pen tip within -2.2 mm to 0.9 mm of the plane throughout.

Table 4 and Fig. 6 summarize the quantitative results, with improvements relative to PID. The ranking PID $\rightarrow$ Fuzzy-PID $\rightarrow$ NN-RBF-PID held for all five metrics. Fuzzy-PID improved r_{RMS} and IAE by about 40% but the peak metrics by only about 25% , i.e., it reduced the average error more than the error peaks. NN-RBF-PID improved z_{max} the most (87.1%), consistent with the network having learned the under-compensated gravity, the main cause of the sag.

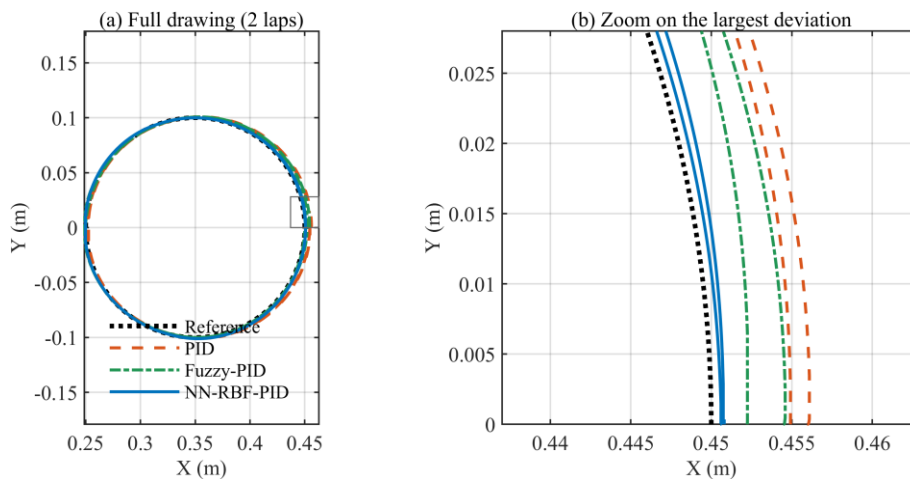


Fig. 4 Drawn paths of the three controllers in the XY plane

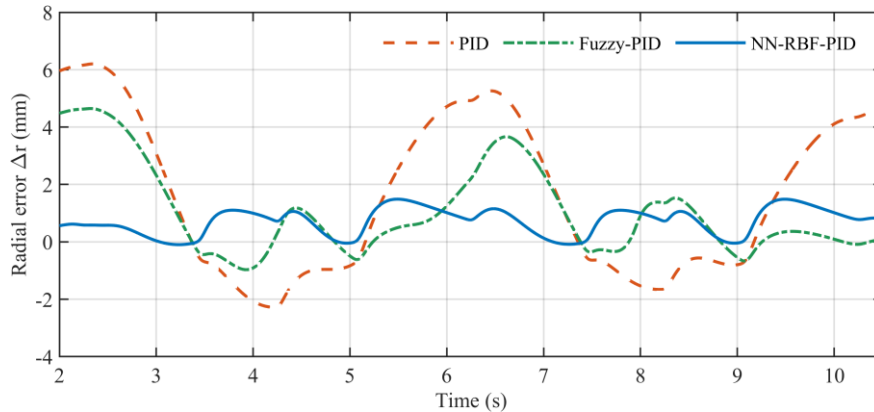


Fig. 5 Radial errors of the three controllers during drawing

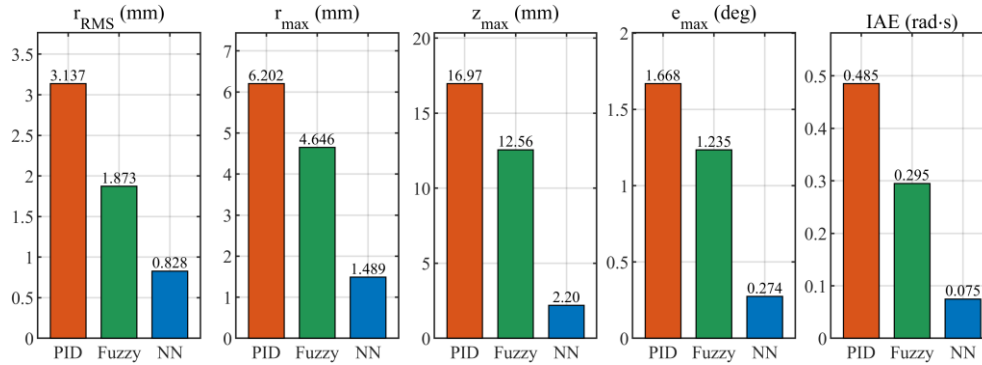


Fig. 6 Bar chart of the five performance metrics for the three controllers

Table (4). Performance metrics of the three controllers in the baseline scenario (MATLAB/Simulink data)

Metric	PID	Fuzzy-PID	NN-RBF-PID
r_{RMS} (mm)	3.137	1.873 (−40.3%)	0.828 (−73.6%)
r_{max} (mm)	6.202	4.646 (−25.1%)	1.489 (−76.0%)
z_{max} (mm)	16.965	12.562 (−26.0%)	2.196 (−87.1%)
e_{max} (deg)	1.668	1.235 (−26.0%)	0.274 (−83.6%)
IAE (rad·s)	0.485	0.295 (−39.2%)	0.075 (−84.6%)

CONTROL MECHANISM ANALYSIS

In Fuzzy-PID, K_p and K_d rose only slightly, at most about 1.33 times (joint 2) and 1.22 times (joint 3) their baseline values during large errors, whereas the median K_i/K_{i0} at joints 1 and 3–6 was 6.3–7.2 because their normalized errors stayed near ZE. The stronger integral action compensated the missing gravity faster, reducing the RMS error of joint 2 from 1.03° to 0.65° and of joint 3 from 0.73° to 0.33°. Yet, by only retuning gains, it must wait for an error to appear and cannot remove a systematic, repetitive discrepancy such as the gravity deficit.

With NN-RBF-PID, the joint 2 and 3 weights grew quickly from zero during the approach, then settled into a steady 4 s periodic pattern repeated in both laps; the largest absolute weight was only about 1.57, far below the ± 60 limit. The network torque (1.1–3.2 N·m at joint 2, 1.4–1.9 N·m at joint 3) matched the 15% gravity deficit (1.3–3.1 N·m and about 1.7 N·m, respectively). The network thus learned the missing torque, leaving the PID part of the NN branch near zero. During drawing, the RMS joint errors were 1.7 (joint 1) to 10.7 (joint 3) times smaller than with PID. No extra torque was needed: peak torques of the three controllers at joints 1–5 differed by less than 2%, and the control energy $U = \int \tau^T \tau dt$ of NN-RBF-PID (4613 N²·m²·s) was about 1.7% lower than that of PID (4693 N²·m²·s).

CONTROL SCENARIOS

Table 5 summarizes the seven scenarios (MATLAB/Simulink data). **Scenario 2** asks whether the network merely mimics a high-gain PID. With K_p increased by a factor of 5, the PID r_{RMS} fell to 0.722 mm, below even NN-RBF-PID with baseline gains (0.828 mm). At the same gain, however, NN-RBF-PID cut the error by a further 55% (0.327 mm versus 0.722 mm) and z_{max} from 3.46 mm to 1.40 mm, so the network adds a real benefit on top of higher gains. On real hardware, a 5-fold K_p would amplify measurement noise and excite vibrations.

 Table (5). RMS radial error r_{RMS} (mm) and maximum vertical deviation z_{max} (mm) in the seven scenarios (MATLAB/Simulink)

Scenario	r_{RMS} PID	r_{RMS} Fuzzy	r_{RMS} NN	z_{max} PID	z_{max} Fuzzy	z_{max} NN
1. Baseline (15%)	3.137	1.873	0.828	16.97	12.56	2.20
2. $K_p \times 5$	0.722	0.566	0.327	3.46	2.93	1.40
3. Mismatch 5%	1.058	0.605	0.528	5.80	3.76	0.90
4. Mismatch 25%	5.126	3.217	1.144	28.44	20.60	3.63
5. Learning off ($\gamma = 0$)	3.147	1.878	3.147	16.97	12.56	16.97
6. Feedforward only	361.5	361.5	361.5	848.5	848.5	848.5
7. Perfect compensation	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0

Scenarios 3 and 4 examine robustness to the model mismatch level (Fig. 7). As the mass mismatch

rose from 5% to 25%, r_{RMS} grew 4.8 times for PID (1.058 $\rightarrow$ 5.126 mm) and 5.3 times for Fuzzy-PID (0.605 $\rightarrow$ 3.217 mm), but only 2.2 times for NN-RBF-PID (0.528 $\rightarrow$ 1.144 mm). Nearly equal at 5%, Fuzzy-PID and NN-RBF-PID diverged as the model degraded: the larger the unknown part, the clearer the network's advantage.

In **scenario 5** ($\gamma = 0$), the weights stayed at zero, $u_{nn} = 0$, and the NN branch matched the PID branch of the same run digit for digit (3.147 mm vs. 3.137 mm due to the variable-step solver), so all of the gain stems from online learning. In **scenario 6** (feedforward only), the joint error exceeded 10° after only 0.39 s and 90° at 0.76 s; the robot could not hold its posture (RMS radial error 361.5 mm), so feedback is indispensable. In **scenario 7** (perfect compensation), all three gave identical, practically zero errors (0.000 mm rounded to 10^{-3} mm), which is why the baseline must include a model mismatch.

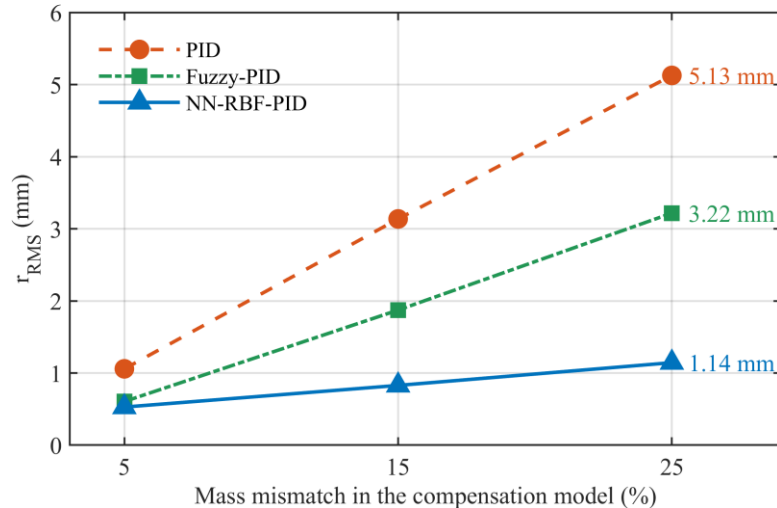


Fig. 7 Robustness of the three controllers to the mass mismatch in the compensation model

IV. CONCLUSION

Using verified models of a 6-DOF arm with a spherical wrist, PID, Fuzzy-PID and NN-RBF-PID were compared under identical conditions on a circle-drawing task. The main conclusions are as follows:

1. With a 15% mass mismatch and Coulomb friction, Fuzzy-PID reduces the RMS radial error by 40.3% and IAE by 39.2% relative to PID, mainly through stronger integral action at small errors.
2. NN-RBF-PID gives the best results: $r_{RMS} = 0.828$ mm (73.6% lower), $z_{max} = 2.196$ mm (87.1% lower) and an 84.6% reduction in IAE, without extra peak torque or control energy, as the network learns the gravity deficit.
3. From 5% to 25% model mismatch, the NN-RBF-PID error grows only 2.2 times, versus 4.8 times for PID and 5.3 times for Fuzzy-PID. A 5-fold K_p brings PID to 0.722 mm, yet at that gain NN-RBF-PID still lowers the error by a further 55%.

The study has some limitations and the results are based on simulations assuming the parameters. The inertia model was simplified (slender rods, no cross-rotational terms), motors, torque limits, measurement noise, sampling delay and link flexibility were neglected. The network makes use only of the error and its derivative, on one repetitive trajectory and speed. Gains were obtained by trial tuning and approximate analysis, without systematic optimization, and no comparison was made with model based adaptive or sliding-mode control [16]. Future work will focus on hardware tests, GA/PSO tuning [2] and comparison with these methods.

FUNDING STATEMENT

This research is funded by Electric Power University under research project code ĐTNH.32/2026.

REFERENCES

- [1]. Trinh, Q. V. [2013] "Industrial robots – Structure, kinematics and dynamics" Science and Technics Publishing House, Hanoi, URL: <https://thuvienso.bvu.edu.vn:8443/handle/TVDHBRVT/3787> (in Vietnamese).
- [2]. Huynh, T. H. [2014] "Intelligent control systems" Vietnam National University Ho Chi Minh City Press, ISBN: 978-604-73-2014-1 (in Vietnamese).
- [3]. Vu, T. Y., Bui, V. H. and Phung, T. V. [2020] "Design of adaptive robust controller based on neural networks control for industrial robot manipulator" The University of Danang – Journal of Science and Technology, Vol. 18, Issue 11: pp.21–26, URL: <https://jst-ud.vn/jst-ud/article/view/3330> (in Vietnamese).
- [4]. Bui, H. T. and Le, T. D. [2017] "Designing adaptive controllers for robot manipulators considering motor model" The University of Danang – Journal of Science and Technology, Issue 1(110): pp.80–85, URL: <https://jst-ud.vn/jst-ud/article/view/2352> (in Vietnamese).

- [5]. Nguyen, D. T., Le, H. D., Tran, C. C. and Nguyen, C. N. [2017] "RBF-based model reference adaptive control system" Can Tho University Journal of Science, Issue 50: pp.37–42, doi: 10.22144/ctu.jvn.2017.064 (in Vietnamese).
- [6]. Spong, M. W., Hutchinson, S. and Vidyasagar, M. [2020] "Robot Modeling and Control" 2nd ed., John Wiley & Sons, ISBN: 978-1-119-52399-4.
- [7]. Craig, J. J. [2018] "Introduction to Robotics: Mechanics and Control" 4th ed., Pearson, ISBN: 978-0-13-348979-8.
- [8]. Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. [2009] "Robotics: Modelling, Planning and Control" Springer, London, doi: 10.1007/978-1-84628-642-1.
- [9]. Zhu, T., Mao, J., Han, L. and Zhang, C. [2024] "Fuzzy adaptive model predictive control for image-based visual servoing of robot manipulators with kinematic constraints" International Journal of Control, Automation and Systems, Vol. 22, Issue 1: pp.311–322, doi: 10.1007/s12555-022-0205-6.
- [10]. Hadi Barhaghtalab, M., Meigoli, V. and Ghaffari, V. [2019] "ANFIS+PID hybrid controller design for controlling of a 6-DOF robot manipulator and its error convergence analysis" Journal of Control, Vol. 13, Issue 3: pp.51–70, URL: <https://joc.kntu.ac.ir/article-1-524-en.html>.
- [11]. Çetinkaya, M. B., Yildirim, K. and Yildirim, Ş. [2024] "Trajectory analysis of 6-DOF industrial robot manipulators by using artificial neural networks" Sensors, Vol. 24, Issue 13: Article 4416, doi: 10.3390/s24134416.
- [12]. Zhao, Z.-Y., Tomizuka, M. and Isaka, S. [1993] "Fuzzy gain scheduling of PID controllers" IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, Issue 5: pp.1392–1398, doi: 10.1109/21.260670.
- [13]. Park, J. and Sandberg, I. W. [1991] "Universal approximation using radial-basis-function networks" Neural Computation, Vol. 3, Issue 2: pp.246–257, doi: 10.1162/neco.1991.3.2.246.
- [14]. Lewis, F. L., Jagannathan, S. and Yesildirek, A. [1999] "Neural Network Control of Robot Manipulators and Nonlinear Systems" Taylor & Francis, London, URL: https://lewisgroup.uta.edu/FL%20books/Lewis_Jagannathan_Yesildirek%20-%20neural%20network%20control%201999.pdf.
- [15]. Ioannou, P. A. and Sun, J. [1996] "Robust Adaptive Control" Prentice Hall, ISBN: 0-13-439100-4.
- [16]. Slotine, J.-J. E. and Li, W. [1991] "Applied Nonlinear Control" Prentice Hall, ISBN: 0-13-040890-5.
- [17]. Mamdani, E. H. and Assilian, S. [1975] "An experiment in linguistic synthesis with a fuzzy logic controller" International Journal of Man-Machine Studies, Vol. 7, Issue 1: pp.1–13, doi: 10.1016/S0020-7373(75)80002-2.
- [18]. Åström, K. J. and Hägglund, T. [2006] "Advanced PID Control" ISA – The Instrumentation, Systems, and Automation Society, ISBN: 978-1-55617-942-6.
- [19]. Biagiotti, L. and Melchiorri, C. [2008] "Trajectory Planning for Automatic Machines and Robots" Springer, Berlin, doi: 10.1007/978-3-540-85629-0.
- [20]. Liu, J. [2013] "Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation" Springer, Berlin, doi: 10.1007/978-3-642-34816-7.
- [21]. Kelly, R., Santibáñez, V. and Loria, A. [2005] "Control of Robot Manipulators in Joint Space" Springer, New York, doi: 10.1007/b135572.
- [22]. ISO [1998] "ISO 9283:1998 Manipulating industrial robots – Performance criteria and related test methods" International Organization for Standardization, URL: <https://www.iso.org/standard/22244.html>.
- [23]. The MathWorks, Inc. [2026] "MATLAB, Simulink, Symbolic Math Toolbox and Fuzzy Logic Toolbox – Documentation" <https://www.mathworks.com/help/>, accessed 2026.
- [24]. Song, Q., Li, S., Bai, Q., Yang, J., Zhang, A., Zhang, X. and Zhe, L. [2021] "Trajectory planning of robot manipulator based on RBF neural network" Entropy, Vol. 23, Issue 9: Article 1207, doi: 10.3390/e23091207.
- [25]. Zhang, Y. and Zhao, L. [2025] "Adaptive control and state error prediction of flexible manipulators using radial basis function neural network and dynamic surface control method" PLoS ONE, Vol. 20, Issue 2: Article e0318601, doi: 10.1371/journal.pone.0318601.
- [26]. Olewi, B. K., Jasim, M., Azar, A. T., Ahmed, S. and Mahlous, A. R. [2025] "Bat optimization of hybrid neural network-FOPID controllers for robust robot manipulator control" Frontiers in Robotics and AI, Vol. 12: Article 1487844, doi: 10.3389/frobt.2025.1487844.
- [27]. Abdulameer, H. I. and Mohamed, M. J. [2022] "Fractional order fuzzy PID controller design for 2-link rigid robot manipulator" International Journal of Intelligent Engineering and Systems, Vol. 15, Issue 3: pp.103–117, doi: 10.22266/ijies2022.0630.10.
- [28]. Hamzah, M. H. M., Thamrin, N. M. and Tajjudin, M. [2022] "Robotic arm position control using Mamdani fuzzy logic on Arduino microcontroller" Journal of Mechanical Engineering, Vol. 19, Issue 3: pp.235–255, URL: <https://jmeche.uitm.edu.my/wp-content/uploads/2022/09/14.%20JMECHE-2022-0020.pdf>.
- [29]. Eltayeb, A., Ahmed, G., Imran, I. H., Alyazidi, N. M. and Abubaker, A. [2024] "Comparative analysis: fractional PID vs. PID controllers for robotic arm using genetic algorithm optimization" Automation, Vol. 5, Issue 3: pp.230–245, doi: 10.3390/automation5030014.
- [30]. Pizarro-Lerma, A., Santibáñez, V., Garcia-Hernandez, R., Villalobos-Chin, J. and Moreno-Valenzuela, J. [2025] "A new motion tracking controller with feedforward compensation for robot manipulators based on sectorial fuzzy control and adaptive neural networks" Mathematics, Vol. 13, Issue 6: Article 977, doi: 10.3390/math13060977.