PD CONTROL WITH AUTO-TUNED GAINS USING RBF NETWORKS FOR ENHANCED TRAJECTORY TRACKING IN MANIPULATOR ROBOTS

Carlos MUÑIZ-MONTERO*, Luis A. SÁNCHEZ-GASPARIANO, Javier LEMUS-LÓPEZ***

*Academic Program of Electronics and Telecommunications, Polytechnic University of Puebla,
Tercer Carril del Ejido "Serrano" S/N, San Mateo Cuanalá, Juan C. Bonilla, Puebla, Pue. 72640, Mexico
**Faculty of Electronic Sciences, Benemérita Universidad Autónoma de Puebla, Pue. 72590 Mexico

carlos.muniz@uppuebla.edu.mx, luis.sanchezgas@correo.buap.mx, javier.lemus@uppuebla.edu.mx

received 21 September 2024, revised 18 September 2025, accepted 05 October 2025

Abstract: The proposed PD-type position control law for manipulator robots adjusts the proportional gains based on the desired position. This capability improves the PD controller's efficiency in trajectory tracking in terms of accumulated position error and energy efficiency, even when the manipulator's dynamic model parameters are unknown. To accomplish this goal, a Radial Basis Function interpolation network, trained offline to avoid higher computational demands, replaces each proportional gain. The Lyapunov method ensures the system's stability, and its effectiveness in position control is further assessed under parametric uncertainties and external perturbations through Monte Carlo analysis and Kruskal–Wallis statistical tests. Matlab simulations on a two-degree-of-freedom manipulator arm following an owl-shaped trajectory demonstrate that, in trajectory tracking, the proposed controller achieves improved $\mathcal{L}_2$ norm and energy efficiency compared with the PD controllers of Takegaki–Arimoto and Tanh(•) with bounded actions.

Key words: PD control, Radial Basis Function Networks, trajectory tracking, Lyapunov stability

1. INTRODUCTION

In recent decades, industries have used robotic systems to automate repetitive or dangerous tasks in industrial applications. In automated processes where the robot moves within its workspace without interacting with the environment, a control system based on position or motion is a viable option. However, as the demands of scientific and technological production processes increase, the tasks assigned to robotic arms are becoming increasingly complex [1]. Motion control has the drawback of requiring a precise model of all the robot's parameters, which is challenging because complete model information is not always available and often needs to be obtained through regression models [2]. The motion control problem also involves modeling through non-autonomous differential equations, requiring asymptotic stability proofs using strict Lyapunov functions—a non-trivial task [3]. To address this challenge, position control, also known as the regulation problem, simplifies trajectory tracking by using point-to-point control [4]. This approach enables the robot's end-effector to move to a fixed and constant desired position over time, regardless of its initial joint position. Since the initial position does not affect the system's stability, the manipulator can trace a series of consecutive points in space (where the previous point serves as the initial condition) that approximate the desired trajectory.

In the regulation problem, the Proportional-Derivative (PD) control law with gravity compensation proposed in [5] ensures global asymptotic stability of the closed-loop system with an appropriate selection of proportional and derivative gains. However, since these gains are constant, abrupt changes in the desired trajectory can increase tracking error, leading some authors to propose variable gains to improve performance. For instance, in [4], researchers propose a solution to the regulation problem by introducing a set of saturated controllers with variable gains. These controllers generate torques within the prescribed limits of the servomotors. The functions for the variable gains allow for smooth self-tuning as the joint position error and velocity approach zero, but it is still necessary to establish parameters for the proportional and derivative gains. In [6], the authors propose a modified neural network algorithm as an adaptive tuning method to optimize the controller gains. The proposal is complex, but robust against uncertainties in system parameters and various trajectories. Unfortunately, it lacks a stability demonstration and requires careful selection of the controller gain limits. The authors of [7] introduce an optimization technique based on an improved Artificial Bee Colony. This technique uses Lyapunov stability functions to determine the optimal gains of a Proportional-Integral-Derivative controller in a 3 degrees-of-freedom manipulator system. The optimized system shows robustness against various perturbation conditions and uncertainty in the payload mass, but the algorithm lacks a stability demonstration, and understanding its operating principle requires considerable effort. Similar observations can be applied to gain tuning using fuzzy algorithms [8]. The authors in [9] propose a regulator with constant proportional gains and variable derivative gains to enhance the robot's transient response through damping, utilizing position error and velocity. This allows the system to reach a steady state smoothly while meeting the servomotor constraints. Although the results are better than the hyperbolic tangent controller [10], which is known for its effectiveness, tuning still requires designer expertise. In Section 5, we review additional relevant contributions on PD-like controllers with variable gain adaptations [15–23]. The discussion emphasizes the type of controller, the specific structure of the variable gains (e.g. state-dependent, adaptive, fuzzy, or neural-network-based), the stability analysis methods applied (such as Lyapunov theory, singular perturbation theory, or global

convergence arguments) and the validation strategies adopted, ranging from numerical simulations to experimental implementations.

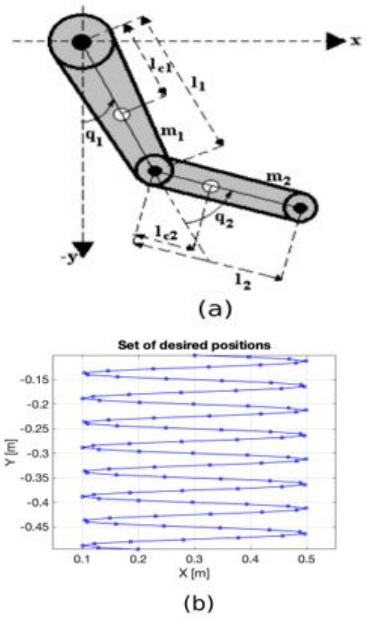

Fig. 1. (a) Two-DOF manipulator diagram; (b) Desired positions in the workspace for RBF training

In this work, we propose a PD-type position control law for robotic manipulators, where the proportional gains are adjusted as functions of the desired position. These gains are obtained through a Radial Basis Function (RBF) interpolation network trained offline, which reduces online computational demands. The objective is to improve trajectory-tracking performance in terms of accumulated error and energy efficiency when compared to the classical PD regulator [5] and the Tanh regulator [10]. The system's stability is formally ensured using Lyapunov's second method. In the remainder of this paper, we denote the proposed approach as the PDN controller. From a theoretical standpoint, PDN does not differ from the classical PD controller of Takegaki–Arimoto [5], since the gains depend solely on the desired position and not on the system states. Consequently, its stability proof is identical to that of the conventional PD law. However, PDN introduces practical advantages: (i) there is no need for re-tuning gains at different desired positions, and (ii) it exhibits enhanced performance in point-to-point trajectory tracking, particularly in terms of robustness, accumulated error, and energy efficiency.

The main contributions of this manuscript can be summarized as follows:

- A methodology for determining the proportional gains of a robotic manipulator controller directly from the desired position, avoiding conventional procedures that depend on designer expertise.
- The use of RBF networks for automatic gain tuning, providing a simpler and more practical implementation compared to alternative methods reported in the literature [6–9].
- Demonstration of improved performance in terms of L2-norm error and energy efficiency in trajectory-tracking tasks with respect to existing approaches.
- Integration of three distinctive features—desired-position dependence, offline RBF training, and Lyapunov-based stability analysis—which together enhance both theoretical guarantees and practical applicability. This combination particularly strengthens point-to-point trajectory tracking and robust regulation without the need for gain re-scheduling, aspects not simultaneously addressed in previous studies.

This work is structured as follows: Section 2 presents the dynamic model of the manipulator robot and the PD regulator. Section 3 discusses the proposed regulator, its stability proof, and the method for training the interpolation networks. Section 4 presents the results of position control and tracking of an owl-shaped trajectory, comparing them with those got from the PD and hyperbolic tangent regulators. A qualitative comparison is presented in Section 5 with state-of-the-art works that employ variable-gain controllers. Section 6 provides the conclusions.

2. MANIPULATOR DYNAMICS AND PD CONTROL

The dynamic model of a manipulator robot with *n* degrees of freedom composed of rigid links (see Fig. 1a) can be written as [11]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + f(\dot{q}) = \tau\,, \tag{1}$$

where $\mathrm{q}, \dot{\mathrm{q}}, \ddot{\mathrm{q}} \in \mathbb{R}^n$ are the joint position, velocity, and acceleration vectors, $\tau \in \mathbb{R}^n$ is the vector of applied torques or control law, $\mathrm{f}(\dot{\mathrm{q}}) \in \mathbb{R}^n$ is the friction phenomena vector (in this work, only viscous friction is considered, $\mathrm{f}(\dot{\mathrm{q}}) = \mathrm{B}\dot{\mathrm{q}}$), $\mathrm{M}(\mathrm{q}) \in \mathbb{R}^{n\mathrm{x}n}$ is the manipulator's inertia matrix, symmetric and positive definite, $\mathrm{C}(\mathrm{q},\dot{\mathrm{q}})\dot{\mathrm{q}} \in \mathbb{R}^n$ is the vector of centrifugal and Coriolis forces, and $\mathrm{g}(\mathrm{q}) \in \mathbb{R}^n$ is the vector of gravitational torques, calculated as the gradient of the manipulator's potential energy $\mathcal{U}(q)$:

$$g(q) = \frac{\partial \mathcal{U}(q)}{\partial q}. \tag{2}$$

For position control or regulation, the dynamic model can be expressed in closed-loop form as:

$$\frac{d}{dt}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}-\dot{q}\\ M^{-1}(q)\{\tau - C(q,\dot{q})\dot{q} - g(q) - B\dot{q}\}\end{bmatrix}, \tag{3}$$

where $q_d = [q_{d1}, q_{d2}, \ldots, q_{dn}]^T \in \mathbb{R}^n$ is the vector of desired positions, and $\tilde{q} = q_d - \mathrm{q} \in \mathbb{R}^n$ is the vector of position errors. The goal of position control is to satisfy [11]:

$$\lim_{t\to\infty}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}, \tag{4}$$

such that the manipulator's end-effector reaches, as time progresses, a fixed and constant desired position with zero velocity $\dot{\mathrm{q}}$=0. To satisfy (4), Takegaki and Arimoto propose the proportional-derivative (PD) control law, given by [5]:

$$\tau = K_p\tilde{q} - K_v\dot{q} + g(q), \tag{5}$$

where K_p, $K_v \in \mathbb{R}^{n\mathrm{x}n}$ are diagonal positive definite matrices of proportional and derivative gains. The values of K_p and K_v can be theoretically approximated according to the tuning rule [11]:

$$k_{pi} < \frac{\tau_i^{max} - k_{g_i(q)}}{\widetilde{q_i}(0)}\,, \tag{6}$$

$$K_{vi} = \rho_i K_{pi}\,, \tag{7}$$

where, for $i = 1, 2, \ldots, n$, k_{pi}>0 is the proportional gain of the i-th link, $k_{g_i(q)}$ represents the upper bound of the gravitational torque, $\widetilde{q_i}(0)$ is the initial position error of the i-th link and ρ_i is a unidimensional positive number between 0 and 1. In practice, K_p and K_v can be chosen through trial and error to meet specifications such

as overshoot or settling time. In either of the two ways, if the desired position is changed, it is necessary to retune K_p and K_v, which limits the precision of the PD regulator for point-to-point trajectory tracking. The expressions (6) and (7) are not optimized values; they are derived from upper bounds to prevent actuator saturation.

3. PROPOSED PDN CONTROL LAW

The following is a proposed variant of the PD position control that automatically adjusts K_p and K_v based on the desired position, transforming from Cartesian space to joint space through inverse kinematics. The aim is to enhance the controller's performance in trajectory tracking, even when the parameters of the dynamic model (1) are not available. This proposal involves replacing the constant gains K_p and K_v in (5) with diagonal matrices $K_p(q_d) = diag\{k_{pi}(q_d)\}$, $K_v(q_d) = diag\{\rho_i k_{pi}(q_d)\}$, for $i = 1, 2, \dots, n$, and $\rho_i \in (0,1)$. The variable gains $k_{pi}(q_d)$ and $k_{vi}(q_d)$ are scalar functions of q_d, the desired position. Each $k_{pi}(q_d)$ corresponds to the output of a Radial Basis Function Interpolation Network [12]. Neural networks are used to learn, from data, the nonlinear map from the desired position to effective PD gains, avoiding manual tuning and gain scheduling. The network is trained offline; at runtime only a fast, deterministic mapping is evaluated, reducing commissioning effort and sustaining consistent performance across desired positions. These networks have an input layer, a hidden layer with Gaussian activation functions $\phi_{i,j}(q_d)$ for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$, where m is the number of neurons in the hidden layer, and an output layer that sums the activation functions weighted by factors w_{ij} [12]. The proposed PD control law is:

$$\tau = diag\{k_{pi}(q_d)\}\tilde{q} - diag\{\rho_i k_{pi}(q_d)\}\dot{q} + g(q), \quad o < \rho_i < 1 \tag{8}$$

$$k_{pi}(q_d) = \sum_{j=1}^{m} w_{ij}\, exp\left[-\left(\frac{\|q_d - C_j\|}{\sigma_i}\right)^2\right] > 0, i = 1 \dots n \tag{9}$$

where w_{ij} are the weights of the hidden layer of the interpolation network $k_{pi}(q_d)$, $C_j \in \mathbb{R}^n$ are the centers of the Gaussian functions, σ_i are constants that controls the width of the Gaussian functions, and ||·|| denotes the Euclidean distance.

3.1. Training Procedure

Next, the following steps detail the procedure for building and training these interpolation networks:

- Design *n* interpolation networks corresponding to the gains $k_{pi}(q_d)$ for each link or degree of freedom.
- Select *m* desired positions distributed within the manipulator's workspace (see Fig. 1b), which will represent the centers of the Gaussian functions in Cartesian coordinates. Convert them to joint coordinates using the manipulator's Inverse Kinematics. The Inverse Kinematics problem involves calculating the angular displacement vector q based on the orientation and position of the end effector, expressed in reference Cartesian coordinates. The expression for the inverse kinematics of the manipulator shown in Fig. 1a, in the "elbow up" configuration and with reference coordinates (x, y), is [13]:

$$q = \begin{bmatrix} \frac{\pi}{2} - \cos^{-1}\left(\frac{l_1^2 - l_2^2 + x^2 + y^2}{2l_1\sqrt{x^2+y^2}}\right) - \cos^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right) \\ \cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1 l_2}\right) \end{bmatrix}. \tag{10}$$

These samples form the training dataset $C=\{C_1, C_2, \dots, C_m\}$, where $C_j \in \mathbb{R}^n$.

- Perform calibration of the gains k_{pi} and select parameters ρ_i for the controller in (5) for each position in the set *C*. The got data will correspond to the training values assigned to the output layer of the *n* interpolation networks, i.e. $\{k_{i1}, \dots, k_{im}\}$, where k_{ij} represents the proportional gain of the i-th joint for the j-th training data.
- Set the value of σ_i such that the activation of the Gaussian functions in the hidden layer is less than 50%:

$$\phi_{i,kj} = exp\left[-\left(\frac{\|Q_k - Q_j\|}{\sigma_i}\right)^2\right] < 0.5\,, \tag{11}$$

with $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, and $k = 1, 2, \dots, m$. This value of σ_i allows the neurons in the hidden layer to specialize in defined regions within the manipulator's workspace. The maximum activation level of these neurons will be reached when the input value is close to their center.

- Calculate the weights w_{ij} of each network by means of:

$$\begin{bmatrix} k_{i1} \\ k_{i2} \\ \vdots \\ k_{im} \end{bmatrix} = \begin{bmatrix} \phi_{i,11} & \phi_{i,12} & \cdots & \phi_{i,1m} \\ \phi_{i,21} & \phi_{i,22} & \cdots & \phi_{i,2m} \\ \vdots & \vdots & \cdots & \vdots \\ \phi_{i,m1} & \phi_{i,m2} & \cdots & \phi_{i,mm} \end{bmatrix} \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{im} \end{bmatrix}, \; i=1,\dots n \tag{12}$$

- Substitute the values of , C_j, and w_{ij} in (8) and (9) to get $k_{pi}(q_d)$ and $k_{vi}(q_d) = \rho_i k_{pi}(q_d)$ as functions of the desired position q_d.
  With this method, *n* interpolation networks are trained offline.

Tab. 1. Parameters of the simulated anthropomorphic arm manipulator

Parameter	Value
l_1, l_2 (link lengths)	0.45m
τ_1^{max} (shoulder), τ_2^{max} (elbow)	150 Nm, 15 Nm
$k_{g_1(q)}$, $k_{g_2(q)}$	40.28 Nm, 1.81 Nm
Inertia Matrix	$M(q) = \begin{bmatrix} 2.351 + 0.167\cos(q_2) & 0.102 + 0.083\cos(q_2) \\ 0.102 + 0.083\cos(q_2) & 0.102 \end{bmatrix}$
Coriolis Matrix	$C(q,\dot{q}) = \begin{bmatrix} -0.1676\sin(q_2)\,\dot{q}_2 & -0.083\sin(q_2)\,\dot{q}_2 \\ 0.084\sin(q_2)\,\dot{q}_1 & 0 \end{bmatrix}$
Gravitational torque vector	$g(q) = 9.81\begin{bmatrix} 3.92\sin(q_1) + 0.186\sin(q_1+q_2) \\ 0.186\sin(q_1+q_2) \end{bmatrix}$
Friction coefficient matrix	$B = \begin{bmatrix} 2.288 & 0 \\ 0 & 0.175 \end{bmatrix}$

3.2. Stability proof

The stability proof of (8) using the direct Lyapunov method is similar to that presented in [5] for conventional PD control. Be $K_p(q_d) = diag\{k_{pi}(q_d)\}$ and $K_v(q_d) = diag\{\rho_i k_{pi}(q_d)\}$, with $k_{pi}(q_d) > 0$ for all $i = 1,2,\dots,n$. These are not functions of time or system states. Let the Lyapunov candidate function be:

$$V(\dot{q},\tilde{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T diag\{k_{pi}(q_d)\}\tilde{q} > 0 \quad (13)$$

Differentiating with respect to time, we have:

$$\dot{V}(\dot{q},\tilde{q}) = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} - \tilde{q}^T diag\{k_{pi}(q_d)\}\dot{q} \quad (14)$$

Substituting $\ddot{q}$ from (3) into (14), considering τ as in (8), and applying the properties $x^T y \equiv y^T x$ and of anti-symmetry $\frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \dot{q}^T C(q,\dot{q})\dot{q} = 0$, we get:

$$\dot{V}(\dot{q},\tilde{q}) = \dot{q}^T diag\{k_{pi}(q_d)\}\tilde{q} - \dot{q}^T diag\{\rho_i k_{pi}(q_d)\}\dot{q} +$$
$$+\dot{q}^T g(q) - \dot{q}^T C(q,\dot{q})\dot{q} - \dot{q}^T g(q) - \dot{q}^T B\dot{q} + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} -$$
$$-\dot{q}^T diag\{k_{pi}(q_d)\}\tilde{q} \quad (15)$$

Simplifying:

$$\dot{V}(\dot{q},\tilde{q}) = \dot{q}^T diag\{k_{pi}(q_d)\}\tilde{q} - \dot{q}^T diag\{\rho_i k_{pi}(q_d)\}\dot{q} +$$
$$-\dot{q}^T B\dot{q} + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \dot{q}^T diag\{k_{pi}(q_d)\}\tilde{q}, \quad (16)$$
$$\dot{V}(\dot{q},\tilde{q}) = -\dot{q}^T diag\{\rho_i k_{pi}(q_d)\}\dot{q} - \dot{q}^T B\dot{q} \le 0 \quad (17)$$

which is true if $\rho_i k_{pi}(q_d) > 0$ for all $i = 1,2,\dots,n$ and because $B > 0$. Thus, global stability of the equilibrium point $[\tilde{q} \quad \dot{q}]^T = [0^T \quad 0^T]^T$ is demonstrated. Since (3) with (8) is an autonomous differential equation, it is possible to prove asymptotic stability using the Barbashin-Krasovskii-LaSalle theorem [14].

4. RESULTS

To validate the PDN control law, simulations were conducted in MATLAB for regulation and trajectory tracking. The trajectory had the shape of an owl within the workspace of the anthropomorphic arm manipulator of Fig. 1a. The parameters of this direct-drive actuator robotic arm are reported in [2, 14], with some of the most important ones shown in Tab. 1.

4.1. Interpolation of $k_{p1}(q_d)$ and $k_{p2}(q_d)$

The gains $k_{p1}(q_d)$ and $k_{p2}(q_d)$ of the PDN regulator were adjusted using two RBF interpolation networks (9) and selecting 50 of the 100 desired positions shown in Fig. 1b. The tuning process aimed to optimize the 100 gains to meet the following criteria: less than 1% overshoot, a response time under 2.5 second, $\tau_1 < 150\ Nm$, $\tau_2 < 15\ Nm$, $\dot{q}_1 < 135$ degrees/s, and $\dot{q}_2 < 270$ degrees/s. These torque and joint velocity limits were set to prevent actuator saturation. The condition in (11) is satisfied with $\sigma_1 = 7.8$ and $\sigma_2 = 13.8$. To determine the weights w_{1j} and w_{2j}, two equations (12) were solved. Fig. 2 presents the obtained functions $k_{p1}(q_d) > 0$ and $k_{p2}(q_d) > 0$. The resulting regulator with $\rho_1 = 0.3$ and $\rho_2 = 0.45$ in (8) will be compared with a PD regulator (5), with parameters k_{p1} = 157, k_{p2} = 6.3, k_{v1} = 0.3k_{p1}, and k_{v2} = 0.45k_{p2} (as reported in [11]), and with a regulator with its bounded actions given by [10]:

$$\tau = diag\{\tanh(k_{pi})\}\tilde{q} - diag\{\tanh(k_{vi})\}\dot{q} + g(q), \quad (18)$$

with k_{p1} = 100, k_{p2} = 6.3, k_{v1} = 0.65k_{p1}, and k_{v2} = 0.65k_{p2}.

4.2. Regulation

Fig. 3 shows the error responses and $\mathcal{L}_2(\tilde{q})$ norms (root-mean-square RMS error) with the PD, Tanh and PDN regulators for $(q_{d1}, q_{d2}) = (-40°, 120°)$, where [11]:

$$\mathcal{L}_2(\tilde{q}) = \sqrt{\frac{1}{T}\int_0^T \|\tilde{q}(t)\|^2 dt}. \quad (19)$$

The PD and PDN controllers exhibit very similar position errors (see Fig. 3a), as expected since the PDN regulator was designed based on the PD regulator. In both, the position errors decrease, reaching a steady state in around 2.5 seconds. The Tanh controller shows a less pronounced correction in q_1 and a more pronounced correction in q_2, but reaching a steady state also around 2.5 seconds. Fig. 3b shows the comparison of the $\mathcal{L}_2$ norms of the three controllers.

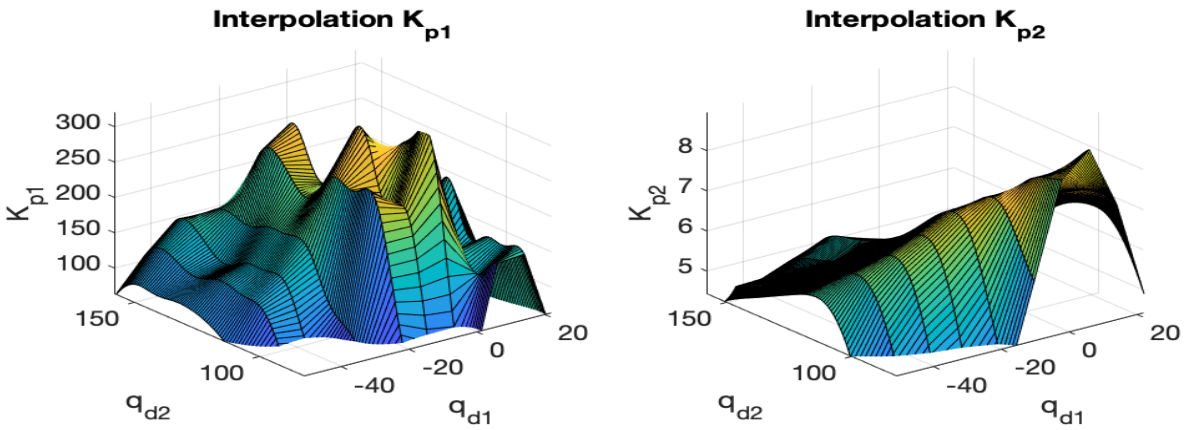

Fig. 2. Interpolation of $k_{p1}(q_{d1}, q_{d2})$ and $k_{p2}(q_{d1}, q_{d2})$

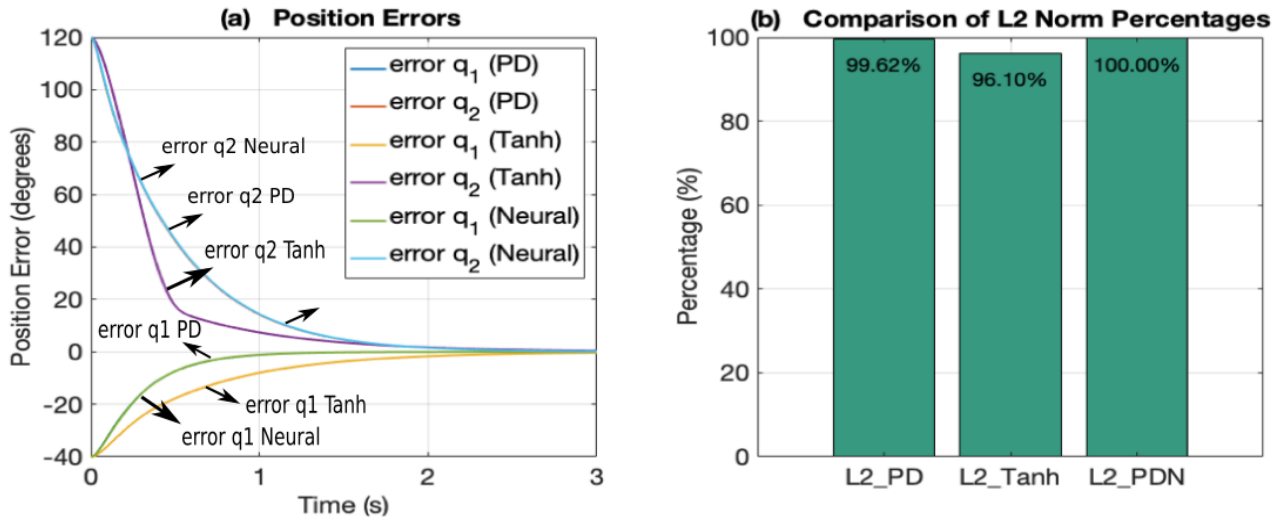

Fig. 3. Responses for $q_{d1} = -40^{o}$ and $q_{d2} = 120^{o}$. (a) Position errors. (b) Comparison of $\mathcal{L}_2$ Norms

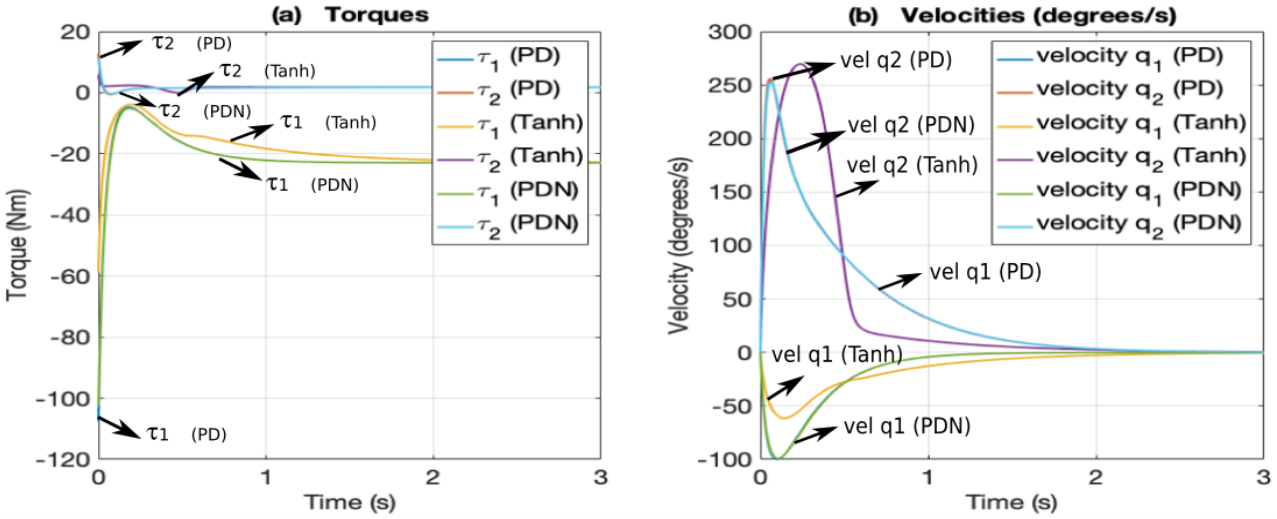

Fig. 4. Responses for $q_{d1} = -40^{o}$ and $q_{d2} = 120^{o}$. (a) Torque responses. (b) Angular velocities

The PDN controller serves as the 100% reference due to its slightly lower performance. The PD and Tanh controllers have lower $\mathcal{L}_2$ norms (less cumulative error), with values of 99.62% and 96.1%. This shows that although all three controllers perform well and similar in terms of position error, the Tanh controller offers a slight advantage by minimizing the $\mathcal{L}_2$ norm, suggesting better overall performance in terms of total error energy compared to the other two controllers in the regulation problem. This comparable performance of the three regulators will allow for a meaningful evaluation of their performance in the point-to-point trajectory-tracking problem.

Fig. 4 shows the corresponding torques and angular velocities. As expected, the responses of PD and PDN are practically identical (see Fig. 4a). Both control laws apply similar forces to the manipulator, resulting in equivalent performance. In both cases, torque τ_1 shows a sharp negative peak at the start (-107.46 Nm and -102.48 Nm, respectively) before stabilizing near to $\tau_1 = -22.9$ Nm, indicating a significant initial effort to correct the position, but still within actuator limit of 150 Nm. Torque τ_2 exhibits a smoother behavior, with an initial peak (12.69 Nm in the PD and 11.51 Nm in the PDN, still within actuator limit of 15 Nm) that stabilizes to $\tau_2 = 1.79$ Nm. The Tanh regulator generates lower torques than the PD and PDN controllers (τ_1 initial of -58.74 Nm and τ_2 initial of 5.75 Nm). Additionally, the torque τ_1 in Tanh stabilizes more slowly to $\tau_1 = -22.9$ Nm, suggesting a behavior with bounded actions. As with the other controllers, torque τ_2 is smoother than τ_1 and stabilizes to $\tau_2 = 1.79$ Nm. The angular velocities in Fig. 4b show that the PD and PDN controllers exhibit almost identical behavior, as expected. In both cases, the velocities of q_1 and q_2 reach their maximum values and then decreases. The Tanh controller results in a more damped response, especially for q_1, showing that its bounded actions result in smoother transitions. In all scenarios, the angular velocities remain within the maximum limits.

DOI 10.2478/ama-2025-0070

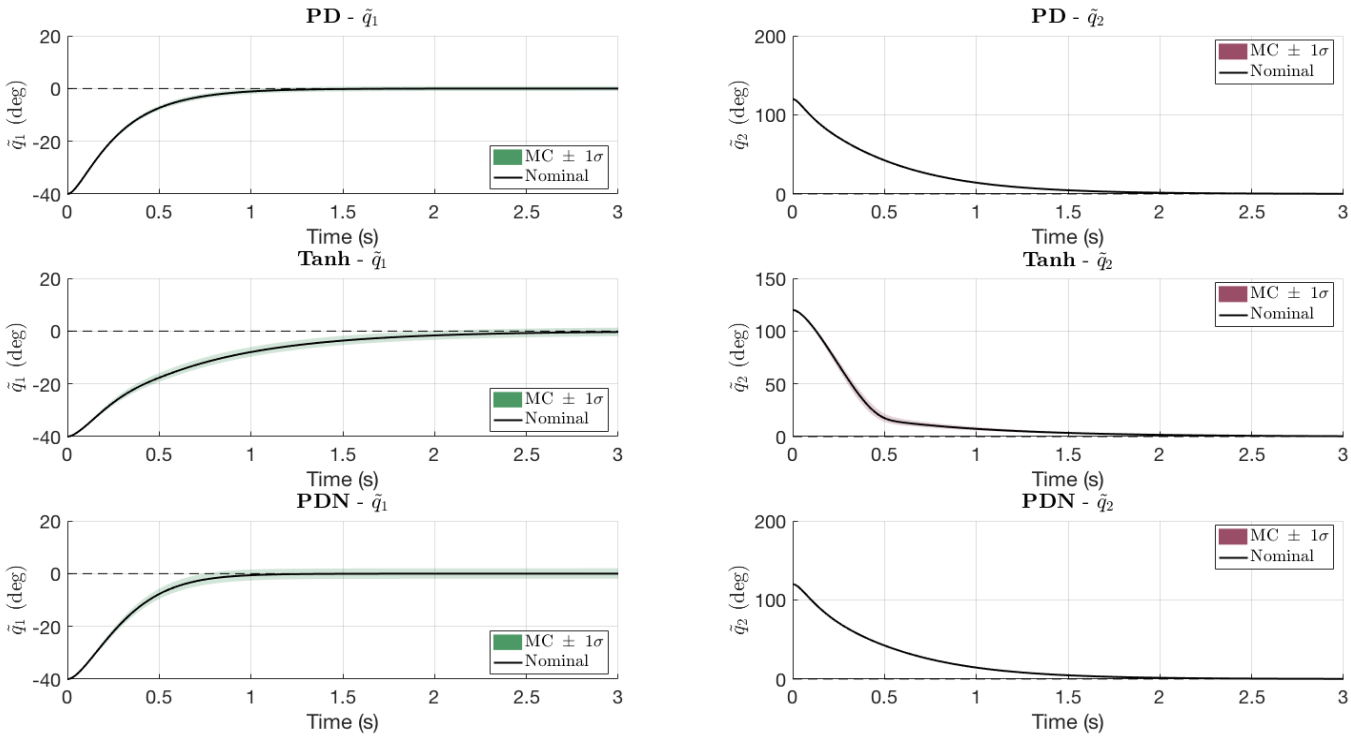

Fig. 5. Monte Carlo robustness of PD, Tanh, and PDN controllers under parametric uncertainty and (q_{d1}, q_{d2}) = (−40°, 120°)

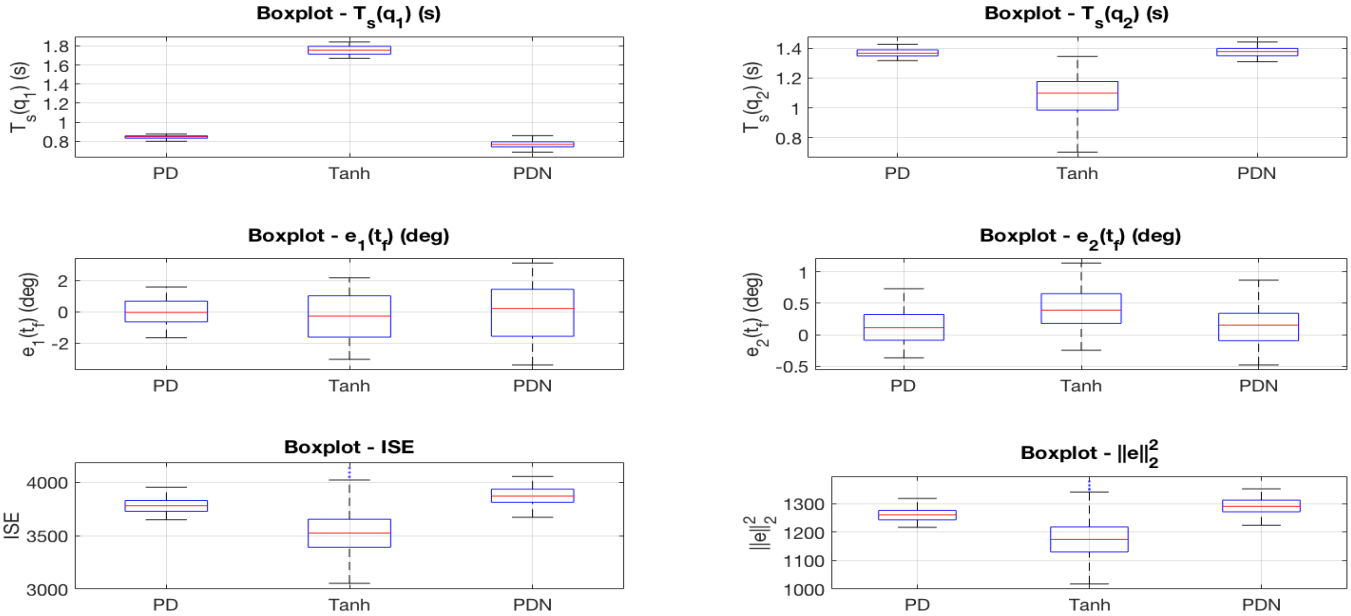

Fig. 6. Boxplots of robustness metrics (T_s, $e(t_s)$, ISE, and $\mathcal{L}_2 - \text{i.e.}, \|e\|^2 -$) for PD, Tanh, and PDN controllers with (q_{d1}, q_{d2}) = (−40°, 120°), ±15% parametric uncertainty, and 200 Monte Carlo trials

To evaluate controller robustness under parametric uncertainty—for example, payload-driven changes in link-2 mass and center of mass—we ran a 200-trial Monte Carlo with truncated Gaussian perturbations (±15%) applied to all the Tab. 1 coefficients with the desired position $(q_{d1}, q_{d2}) = (-40°, 120°)$.

The Fig. 5 overlays the nominal error response (black) with shaded ±1 standard deviation bands, and Fig. 6 shows boxplots of robustness metrics obtained by Kruskal-Wallis and bootstrap analyses. Qualitatively, PD exhibits the tightest dispersion and quickest convergence in the q_1 error; PDN shows similar transients but a wider spread, consistent with PD gains being hand-tuned for this set-point while PDN gains are produced by a neural estimator not tailored to the operating point. For q_2, all controllers decay rapidly, but Tanh is visibly poorer and with larger spread.

Because the Monte Carlo metrics were non-Gaussian (normality rejected by a Lilliefors/Kolmogorov–Smirnov test), we used Kruskal–Wallis and complemented it with bootstrap 95% confidence intervals for the medians. The analysis confirms strong between-controller differences for settling time $T_s(q_1)$ (significance p<0.05): PDN attains the smallest median $T_s(q_1)$ = 0.773 s with confidence interval [0.768, 0.779], a 56.1% improvement over the worst case (Tanh).

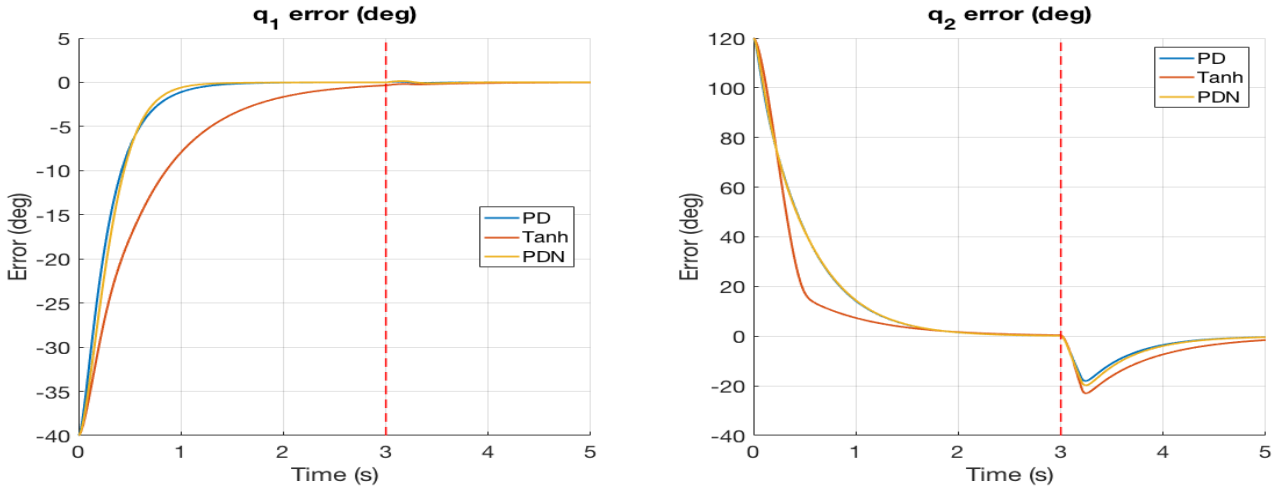

Fig. 7. Disturbance Rejection to a 6 Nm, 0.15 s Torque Pulse at Joint 2 (t = 3 s): q1 and q2 Position Errors for PD, Tanh, and PDN

For $T_s(q_2)$ (p<0.05), Tanh is fastest with median 1.117 s [1.068, 1.150], 19.0% better than the worst (PDN). Steady-state error at final time, $e_1(t_s)$, shows no significant differences (p=0.068), whereas $e_2(t_s)$ does (p<0.05); PD yields the lowest median $e_2(t_s)$ = 0.185 deg [0.068, 0.260], a 32.8% reduction relative to Tanh. Integrated error measures also differ markedly (p<0.05): Tanh achieves the smallest ISE (median 3573) and the lowest time-normalized error $\mathcal{L}_2$ (median 1191), about 8.2% better than PDN. Overall, PDN is fastest in joint-1, Tanh is fastest in joint-2 and most favorable in energy-like errors, and PD minimizes steady-state error for joint-2.

To analyze the effect of PDN self-tuning—while keeping the gains of the other controllers fixed at the values used for the (q_{d1}, q_{d2}) = (−40°, 120°) operating point—we pooled the Monte Carlo data across the three set-points (−40°, 120°), (0°, 90°), and (−20°, 140°) and applied a Kruskal–Wallis test; the results are as follows. For settling time $T_s(q_1)$, groups differ (p<0.05); PDN attains the lowest median 0.7725 s, improving by 56.2% over the worst (Tanh). For $T_s(q_2)$, groups differ (p<0.05); Tanh is best with median 1.100 s, a 19.1% improvement over the worst (PDN). For steady-state error $e_1(t_s)$, differences are not significant (p≥ 0.05). For $e_2(t_s)$, groups differ (p<0.05); PDN is best with median 0.1163 deg, a 55.9% reduction relative to the worst (Tanh). For the integral metrics, ISE and$\mathcal{L}_2$, groups differ (p<0.05); Tanh achieves the lowest medians (ISE = 3529, $\mathcal{L}_2$ = 1176), each 9.3% better than the worst (PDN). Overall, across the three operating points, PDN is fastest in joint-1 transients, Tanh is fastest in joint-2 and most favorable for energy-type errors, and PD does not dominate but remains competitive in steady-state bias for joint-2.

Therefore, we observe that, for regulation tasks, the PDN controller is competitive with PD and Tanh, even with parametric uncertainty, but its main advantage is that it does not require manual tuning of the proportional and derivative gains.

In Fig. 7, an external disturbance was applied at joint 2: a 6 Nm torque starting at t = 3 s and lasting 0.15 s (red dashed line). In q_1 the impact is minor and short-lived due to limited coupling; all controllers keep the error close to zero with only a small blip. In q_2 the disturbance causes a pronounced negative excursion (about −20 to −25 degrees), which reveals clear differences in disturbance rejection: PD shows the smallest excursion and the fastest recovery, PDN is a close second with a slightly deeper dip and slower return, and Tanh performs worst with the largest dip and the longest tail. This ranking aligns with controller structure: PD and PDN retain linear behavior to counter a torque pulse, whereas the Tanh controller soft-limits the proportional and derivative actions, reducing corrective effort.

4.3. Point-to-point trajectory tracking

Fig. 8a shows the ideal trajectory that the robot should follow, represented in the shape of an owl interpolated from 200 points and completed in 25 seconds. The PDN controller permits the robot to track this trajectory with a sampling period of 2.5 ms, corresponding to 10,000 samples, as illustrated in Fig. 8b. The Tanh and PD controllers yield similar results. Fig. 8c and Fig. 8d show the joint velocities for the Tanh and PDN controllers. In both cases, the velocities of joints q_1 and q_2 remain near the saturation limits. However, the Tanh controller achieves higher velocity values, although it stabilizes more quickly, which affects its energy consumption. The energy consumption is calculated as:

$$E(\tau, \dot{q}) = \int_0^T |\tau_1(t)\dot{q}_1(t) + \tau_2(t)\dot{q}_2(t)|dt. \quad (20)$$

Fig. 8e and Fig. 8f compare the performance regarding the $\mathcal{L}_2$ norm and energy consumption. Unlike the regulation control case, in trajectory tracking, the PDN controller shows the best performance in terms of accumulated position error, with the lowest $\mathcal{L}_2$ norm (73.48%), followed by the Tanh controller (74.12%) and then the PD (taken as the 100% reference). This shows that the PDN controller better minimizes the accumulated error. Regarding energy consumption, the PDN controller also stands out with the best performance (80.58%), followed by the PD (83.98%), and the Tanh (100%, taken as the reference). Besides reducing the error, the PDN controller is also more energy-efficient.

5. DISCUSSION

In agreement with Tab. 2, the proposed controller differs from previous variable-gain PD approaches in several key aspects. While many studies introduce variable gains as functions of state variables (e.g., position, velocity, or error) [15–23], our approach uniquely defines the proportional gain as a function of the desired position. This structural distinction has important implications:

DOI 10.2478/ama-2025-0070

– Offline Training and Practical Implementation: Unlike methods that rely on adaptive or iterative online updates [15, 17, 21], the proposed controller employs an RBF interpolation network trained offline. This design reduces online computational burden, making the method more practical for real-time applications without sacrificing performance.

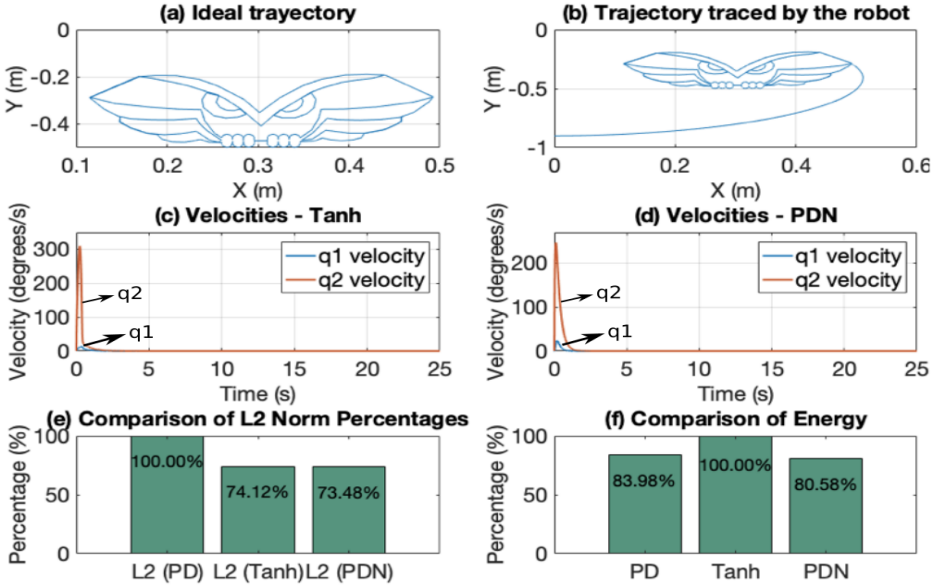

Fig. 8. Point-to-point "Owl" trajectory tracking. (a) Ideal trajectory. (b) Trajectory with the PDN controller. (c) Actuator speed with the Tanh controller. (d) Actuator speed with the PDN controller. (e) Comparison of $\mathcal{L}_2$ norms. (f) Comparison of energy consumption.

Tab. 2. Comparative table of studies on PD controllers with variable gains

Study	Controller Type	Variable Gain Structure	Stability Analysis Method	Validation Approach
[4]	PD-like with variable gains	Variable; state-, position-, and velocity-dependent; smooth functions (e.g. cos² (tanh (error+velocity)))	Lyapunov theory; global asymptotic stability; gravity compensation required	Simulation; two-DOF direct-drive robot; joint regulation; L2 norm
[15]	PD iterative neural-network learning (PDISN)	Likely variable/adaptive; neural network and iterative learning	Extended Lyapunov theories; stability type not specified	Simulation; manipulator characteristics not specified; scenario not specified
[16]	Proportional-derivative (PD)	Variable; tuned by self-organizing fuzzy algorithm	Not analyzed (no details)	Simulation; manipulator characteristics not specified; tracking control; position error metric
[17]	Self-tuning PD	Bounded, time-varying; neuro-fuzzy recurrent scheme	Lyapunov theory; semi-global exponential stability	Simulation; manipulator characteristics not specified; trajectory tracking
[18]	Nonlinear PID with fuzzy self-tuned PD gains	Variable, position-dependent; fuzzy logic	Not mentioned; global asymptotic stability; no gravity compensation	Experiments; type not specified; scenario and metrics not specified
[19]	Adaptive PD	Adaptive to gravity parameters	Not mentioned; global convergence	Simulation; three-DOF manipulator; point-to-point and tracking
[20]	PD-type robust	Variable, error-varying; parameterized by perturbing parameter	Singular perturbation theory; stability type not explicit	Physical experiment; planar two-DOF direct-drive robot; trajectory tracking
[21]	Adaptive iterative learning control (ILC)-PD	Variable; iterative learning, two iterative variables	Lyapunov theory; asymptotic convergence	Simulation; two-DOF manipulator; trajectory tracking
[22]	PD-type	Variable, state-dependent	Not mentioned; global asymptotic stability claimed	Physical experiment; two-DOF direct-drive arm; scenario not specified
[23]	Linear and nonlinear PD-type	Nonlinear functions of system states	Not mentioned; global asymptotic stability claimed	Simulation; single-link and two-DOF robots; trajectory tracking
This work	PD-like with variable gains	Variable; desired position dependent proportional gains with RBF interpolation networks trained offline	Lyapunov theory; global asymptotic stability; gravity compensation required	Simulation; two-DOF direct-drive robot; joint regulation; L2 norm, point-to-point tracking; regulation performance evaluated with parametric uncertainties and external perturbations

– Formal Stability Guarantees: Several previous works either omit stability analysis [16, 18, 19, 23] or provide only limited claims (e.g., “global convergence” without detailed proofs) [19]. In contrast, our control law is rigorously validated through Lyapunov theory, ensuring global asymptotic stability under both trajectory tracking and regulation tasks.
– Point-to-Point Tracking and Regulation without Re-tuning: Whereas prior approaches often require parameter re-adjustment depending on trajectory complexity or regulation scenarios, our method demonstrates strong performance in point-to-point tracking while also showing that regulation tasks can be achieved without re-tuning the gains.

This makes the controller particularly suitable for manipulators executing sequences of set-points, as the stability and efficiency remain robust across tasks.

– Robustness to Uncertainties and Perturbations: While some works include robustness studies under limited assumptions [20], the proposed control law explicitly evaluates performance under parametric uncertainties and external perturbations, confirming its reliability in realistic operating conditions.

6. CONCLUSIONS

Compared to the existing literature, the proposed controller combines three distinctive contributions-desired-position dependence, offline RBF training, and formal Lyapunov stability- which together enhance both theoretical guarantees and practical applicability. Its strength lies particularly in point-to-point trajectory tracking, where the error is reduced in a more energy-efficient form regarding PD and Tanh controllers. It also presents robust regulation without gain re-scheduling, aspects that are not simultaneously addressed by previous studies.

REFERENCES

1. Xu K, Wang Z. The design of a neural network-based adaptive control method for robotic arm trajectory tracking. Neural Computing and Applications. 2023;35:8785–8795. https://doi.org/10.1007/s00521-022-07646-y
2. Chávez-Olivares CA, Reyes-Cortés F, González-Galván EJ, Mendoza-Gutiérrez MO, Bonilla-Gutiérrez I. Experimental evaluation of parameter identification schemes on a direct-drive robot. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2012;226(10):1419–1431.
3. Zavala-Río A, Zamora-Gómez GI, Sanchez T, Reyes-Cortés F. A generalized proportional-derivative type scheme with multiple saturating structure for the finite-time and exponential tracking continuous control of Euler-Lagrange systems with bounded inputs. International Journal of Systems Science. 2022;54(5):945–976. https://doi.org/10.1080/00207721.2022.2153634
4. Sánchez-García B, Reyes-Cortés F, Al-Hadithi BM, Félix-Beltrán O. Global saturated regulator with variable gains for robot manipulators. Journal of Robotics and Control. 2021;2(6):1–13. https://doi.org/10.18196/jrc.26139
5. Takegaki M, Arimoto S. A new feedback method for dynamic control of manipulators. ASME Journal of Dynamic Systems, Measurement, and Control. 1981;103:119–125.
6. Elsisi M, Mahmoud K, Lehtonen M, Darwish MMF. An improved neural network algorithm to efficiently track various trajectories of robot manipulator arms. IEEE Access. 2021;9:11911–11920. https://doi.org/10.1109/ACCESS.2021.3051807
7. Azeez MI, Abdelhaleem AMM, Elnaggar S, et al. Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced Artificial Bee Colony algorithm. Scientific Reports. 2023;13:11164. https://doi.org/10.1038/s41598-023-37895-3
8. Kumar J, Kumar V, Rana KPS. Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system. Neural Computing and Applications. 2019;31:1–14. https://doi.org/10.1007/s00521-019-04215-8
9. Limón-Díaz MA, Reyes-Cortés F, González-Galván EJ. Regulación saturada con ganancia variable derivativa de robots manipuladores. Revista Iberoamericana de Automática e Informática Industrial. 2017;14(4):434–445. https://doi.org/10.1016/j.riai.2017.06.001
10. Kelly R, Santibañez V, Reyes F. On saturated-proportional derivative feedback with adaptive gravity compensation of robot manipulators. International Journal of Adaptive Control and Signal Processing. 1996;10:465–479. https://doi.org/10.1002/(SICI)1099-1115(199607)10:4/5<465::AID-ACS375>3.0.CO;2-8
11. Reyes-Cortés F, Chávez-Olivares C, González-Galván EJ. A family of hyperbolic-type explicit force regulators with active velocity damping for robot manipulators. Journal of Robotics. 2018;2018:9324623. https://doi.org/10.1155/2018/9324623
12. Sánchez-López C. Planning handwriting for a 2-DOF planar robot arm. International Journal of Mechanics and Control. 2024;25(1):115–121. https://doi.org/10.69076/jomac.2024.0016.
13. Reyes-Cortés F. Robótica: Control de Robots Manipuladores. 2nd ed. Mexico City: Alfaomega; 2024. ISBN: 9786075761251
14. Dang XB, Nguyen TT, Bae J. A novel iterative second-order neural-network learning control approach for robotic manipulators. IEEE Access. 2023;11:3280979. https://doi.org/10.1109/ACCESS.2023.3280979
15. Salas FG, Santibáñez V, Llama M. Variable gains PD tracking control of robot manipulators: stability analysis and simulations. Workshop on Autonomic Communication. 2012;1–6..
16. Armendariz J, Parra-Vega V, García-Rodríguez R, Hirai S. Dynamic self-tuning PD control for tracking of robot manipulators. Proceedings of the IEEE Conference on Decision and Control (CDC). 2012;6426562. https://doi.org/10.1109/CDC.2012.6426562
17. Sifuentes-Mijares J, Santibáñez V, Meza-Medina JL. A globally asymptotically stable nonlinear PID regulator with fuzzy self-tuned PD gains for robot manipulators. Workshop on Autonomic Communication. 2014;6936049. https://doi.org/10.1109/WAC.2014.6936049
18. Tomei P. Adaptive PD controller for robot manipulators. IEEE Transactions on Robotics and Automation. 1991;7(4):565–570. https://doi.org/10.1109/70.86088
19. González-Vázquez S, Moreno-Valenzuela J. Time-scale separation of a class of robust PD-type tracking controllers for robot manipulators. ISA Transactions. 2013;52(2):193–201. https://doi.org/10.1016/j.isatra.2012.11.007
20. Jia X, Yuan Z. Adaptive iterative learning control for robot manipulators. Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICICIS). 2010;5658818. https://doi.org/10.1109/ICICISYS.2010.5658818
21. Kelly R, Carelli R. A class of nonlinear PD-type controllers for robot manipulators. Journal of Robotic Systems. 1996;13(12):793–802. https://doi.org/10.1002/(SICI)1097-4563(199612)13:12<793::AID-ROB2>3.0.CO;2-Q
22. Liu F, Joo EM. Linear and nonlinear PD-type control of robotic manipulators for trajectory tracking. Proceedings of the IEEE Conference on Industrial Electronics and Applications. 2009;3442–3447. https://doi.org/10.1109/ICIEA.2009.5138846

Carlos Muñiz-Montero: https://orcid.org/0000-0003-4386-5066

Luis A. Sánchez-Gaspariano: https://orcid.org/0000-0002-3899-0746

Javier Lemus López: https://orcid.org/0000-0002-4058-9513