APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 10, 2026, 108-119

Adaptive PD-Based Control of Robot Manipulators with Gravity Compensation and RBF Neural Networks

Ahmed Alkamachi[1,*]

[1]Mechatronics Engineering Department, Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

[*]Corresponding author: ahmed78@kecbu.uobaghdad.edu.iq

Submitted 03 March 2026; Revised 02 May 2026; Accepted 05 May 2026; Available online 09 May 2026.

Abstract: Robotic manipulators play an important role in automation, which requires precise control on their motions. The control of two degrees of freedom (2DOF) robotic arms is challenged by nonlinearities, joint coupling, parametric uncertainty, and external disturbances. This work proposes an efficient hybrid controller for a 2DOF robotic arm, which combines a gravity compensated proportional derivative (PD+G) controller with an adaptive radial basis function neural network (RBFNN) compensator that learn from the error signal and its derivative. The nominal robotic manipulator model along with the proposed controller, is implemented in simulation software. The proposed controller is evaluated through detailed simulations, which provide an initial validation before future experimental implementation. The validation includes overlapped step inputs to clearly reveal the coupling between joints and sinusoidal reference trajectories to examine the controller's ability for continuous tracking. The robustness is examined by applying a step disturbance superimposed on the torque and changes in model parameters. Compared with the PD+G controller, the proposed PD+G+RBFNN control attains improved settling time for the first joint and comparable performance for the second joint, and less overshoot, while the adaptive NN effectively compensates for nonlinearities, unknown dynamics, and disturbances. Quantitative assessment of the system with controllers using several performances indices confirms the superiority of the proposed controller in all test scenarios. The highest significant improvement is achieved when applying sinusoidal tracking, with reductions of more than 90% in the performance indices as compared with the PD+G benchmark. The results show that the proposed configuration is practical, robust, and suitable for controlling nonlinear, coupled manipulators with the existence of uncertainties and disturbances.

Keywords: 2DOF robotic manipulator; Adaptive control; Gravity compensator; Joint coupling; PD control; RBF neural network.

1. INTRODUCTION

Robotic manipulators are used in the automation of production processes to save effort and time. They are used for assembly, welding, painting, and packaging, assuring high precision and efficiency. They also find applications in medicine, scientific research, and the space industry [1, 2]. Classical proportional derivative (PD)/ proportional integral derivative (PID) controllers represent the basic in manipulator control due to their simplicity and ease of tuning [3, 4], but their performance degrades in the presence of nonlinearity, states coupling, parameter uncertainty, and disturbances [5, 6]. Robotic manipulators are nonlinear systems that are highly susceptible to strong nonlinearities and external disturbances [7]. This leads to the development of adaptive and intelligent methods that can approximate unknown dynamics and compensate for disturbances.

In recent years, researches based on neural networks (NNs), combined with classical control, have gained particular attention [8]. Soriano *et al*. proposed PD controller compensated with cascaded neural network and validated it on a two-degrees of freedom (2DOF) manipulator. Comparing it with classical PD, the NN solution acts as a nonlinear compensation over the classical controller and improves tracking quality [9]. Wu *et al*. proposed an integration between backstepping scheme and adaptive NN control to improve manipulator performances and cope with its uncertainties and time varying constraints. The work demonstrates a typical pattern: the NN approximates unknown nonlinearities, and the correction term provides robustness [10]. The authors in [11] introduced an "adaptive bias" radial basis function neural network (RBFNN) to enhance the control performance against large payloads, which was confirmed on simulation-based analysis. Recent work extends this research line to dual networks + sliding mode control (SMC) or structures with perturbation observers, aimed at improving robustness under unknown kinematics parameters [8]. Shen *et al*. designed an adaptive RBFNN–Computed Torque Control (RBFNN–CTC) for a manipulator system with a Strobeck friction model. The adaptive RBFNN algorithm is developed to estimate the unmodeled dynamics online and pass it to the CTC, hence achieving improved tracking accuracy as operating conditions change [12]. Similarly, in exoskeleton applications, CTC compensated by RBFNN has been shown to improve gait tracking of the coupled system [13]. Furthermore, robust/NN–CTC solutions are emerging in [14], confirming the usefulness

of NNs for compensating model errors and external forces while maintaining high robustness.

Reviews and articles synthesizing of recent publications confirm that hybrid NN adopted with classical/adaptive law approaches promote tracking tasks under uncertainty and disturbances, and that the design of adaptation rules contributes to ensure stability and performance [15]. Also worth noting that [16] presented adaptive NNs for manipulators with external disturbances, as well as extended state observer to estimate the link velocities, demonstrating the control ability of RBFNNs beyond the classic 2DOF framework. In a related study, authors in [17] developed a hybrid control scheme that integrates RBFNN with PD controller to approximate and compensate for gravity uncertainties. Their approach focuses on improving the accuracy of step input regulation by learning the static gravity terms of the manipulator.

It is important to distinguish the proposed control architecture from the existing literature, specifically the work in [17] which also utilizes a PD and RBFNN combination. While [17] employs the RBFNN specifically to approximate the unknown gravity vector in a point-to-point regulation task, this work utilizes the RBFNN as a compensator for lumped uncertainties, including dynamic coupling and unmodeled friction. Furthermore, the network inputs in [17] are the joint states, whereas the proposed controller utilizes an error driven approach with inputs derived from the tracking error. This allows for superior performance in continuous trajectory tracking (e.g., sinusoidal references) rather than simple set point regulation. Additionally, while [17] relies on a robust switching term to handle approximation errors, our approach focuses on a smooth adaptive law that reduces chattering while maintaining stability in the presence of inter joint dynamics.

Unlike most existing RBFNN based manipulator controllers that rely on computed torque or backstepping schemes, the proposed controller integrates a gravity compensated PD controller with RBFNN compensator driven by the system error signal. This structure maintains the simplicity of the PD controller while enabling online compensation of coupling effects and uncertainties without requiring full model knowledge. The proposed controller resolves the limitation documented in reviewed literatures which is the inability of fixed structure classical controllers to compensate online for coupled joint dynamics, gravity estimation errors, and external disturbances simultaneously. From theoretical point of view, the proposed structure is superior not just by the numerical performance but because the filtered error Lyapunov formulation guarantees uniform ultimate boundedness (UUB) of all closed loop signals, including time varying reference trajectories. This is a stronger theoretical guarantee than the asymptotic stability results available for set point controllers, and it explicitly suits sinusoidal and continuously varying desired trajectories.

The novelty in the stability analysis lies specifically in:

- Deriving the closed loop error dynamics in terms of desired angle, position error and filtered tracking errors without introducing additional reference variables, therefore ensuring that the desired trajectory first and second derivatives are included within the lumped uncertainty.
- Proving that the RBFNN weight adaptation law and the PD dissipative term together cancel all cross terms in derivative of the candidate Lyapunov function. This will lead to a clean uniform ultimate boundedness (UUB) bound that quantifies convergence as a function of RBFNN approximation accuracy.

Significant theoretical advances have integrated with PD/PID controller to address nonlinearities and guaranteeing stability. Advanced PID control structures, including nonlinear PID scheme and observer based adaptive variants, have extended the operating region of classical controllers under uncertainty [3][4]. These theoretical foundations push toward the use of Lyapunov stability analysis in the present work, where the closed loop error dynamics under the proposed PD+G+RBFNN scheme are shown to be bounded regardless of the existence of modeling uncertainties and external disturbances.

The proposed controller configuration is examined under multiple challenging scenarios to evaluate its effectiveness and robustness. Specifically, (1) the effect of dynamic coupling between joints is clearly demonstrated by applying step inputs at different times; (2) the robustness under model uncertainty (parameter drifts) and external disturbances is tested; (i3) PD+G and the proposed PD+G+RBFNN are compared in multiple scenarios, including sinusoidal tracking. The simulation setup is designed to include quantitative performance metrics such as the integrated time weighted absolute error (ITAE), integral of squared error (ISE), integral of absolute error (IAE), root mean squared error (RMSE) indices, settling time, and overshoot.

The remainder of this paper is organized as follows: In Section 2, the structure of the robotic manipulator utilized in this paper is introduced. In Section 3, the design process of the PD controller with a gravity compensator and proposed RBFNN compensation scheme is presented. Section 4 provides several test scenarios to show the superiority of the proposed controller against the PD+G controller. Finally, a conclusion is given in Section 5.

2. PLANT DYNAMICS

The plant dynamics for the 2DOF robotic manipulator is derived using the Euler-Lagrange formalism, which is widely utilized to describe robotic systems. The system composed of two rigid links which are linked to each other by revolving joints as shown in Figure 1. These links are actuated independently by joint torques (τ_1 and τ_2). The robot operates in a vertical plane so that the effect of gravity is taken into account during modeling. The effect of viscous friction is also considered into the model. The general equation of motion of 2DOF robotic arm is given by [18]:

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + B\dot{\theta} + \tau_d = \tau \tag{1}$$

where $\theta = [\theta_1 \ \theta_2]^T$ represents the joint angles, $\dot{\theta}$, and $\ddot{\theta}$ are their first and second derivatives respectively, and $\tau = [\tau_1 \ \tau_2]^T$ is the joint input torques vector. $D(\theta)$ is the inertia matrix, $C(\theta,\dot{\theta})$ denotes the Coriolis and centrifugal matrix in the equation, $G(\theta)$ is the gravity vector, B is viscous friction coefficients matrix, and τ_d is the unknown disturbance vector.

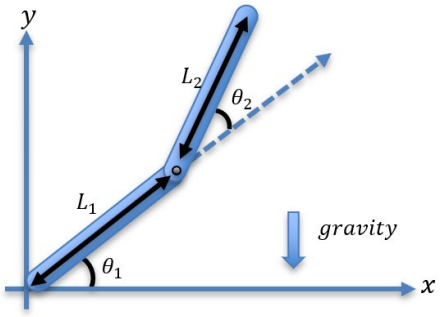


Figure 1. Schematic drawing of a 2DOF robotic arm.

The matrices in the dynamic equation are defined as follows [19]:

$$D(\theta) = \begin{bmatrix} M_1 d_1^2 + I_1 + M_2(L_1^2 + d_2^2 + 2L_1 d_2 cos\theta_2) & M_2(d_2^2 + L_1 d_2 cos\theta_2) + I_2 \\ M_2(d_2^2 + L_1 d_2 cos\theta_2) + I_2 & M_2 d_2^2 + I_2 \end{bmatrix} \tag{2}$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -M_2 L_1 d_2 \, sin\theta_2 \, \dot{\theta}_2 & -M_2 L_1 d_2 \, sin\theta_2 \, (\dot{\theta}_1 + \dot{\theta}_2) \\ M_2 L_1 d_2 \, sin\theta_2 \, \dot{\theta}_1 & 0 \end{bmatrix} \tag{3}$$

$$G(\theta) = \begin{bmatrix} g(M_1 d_1 + M_2 L_1) cos\theta_1 + M_2 d_2 g \cos(\theta_1 + \theta_2) \\ M_2 d_2 g \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \tag{5}$$

where M_1 and M_2 are the masses of the links, L_1 and L_2 are the link lengths, d_1 and d_2 are the distances from the joints to the centers of mass of arm 1 and 2 respectively, I_1 and I_2 are the moments of inertia, g is gravitational acceleration, and b_1 and b_2 are viscous friction coefficients.

This mathematical model is widely used in the robotics literature for analysis, control design, and simulation of manipulator systems [20-22]. Each component of the robot's dynamic model, $D(\theta)$, $C(\theta, \dot{\theta})$, $G(\theta)$, and B, contains both known and unknown terms. The known parts are determined by the robot's nominal physical parameters. However, due to uncertainties such as modeling errors, parameter variations, and external disturbances especially when masses, link lengths, or center of mass locations are incorrectly estimated, additional unknown components appear in these matrices. The parameter values utilized in this work are listed in Table 1.

3. CONTROLLER DESIGN

Robotic manipulators suffer coupling effect between the two arms, nonlinearities, parameter uncertainties, and external disturbances. Therefore, controlling robotic joints with high accuracy and robustness is challenging. Classical control methods such as PD controllers are characterized by its simplicity and good stability attributes but may exhibit limited role in the presence of modeling uncertainties or external disturbances. Adaptive neural network controllers, on the other hand, offer real time learning and have the ability to compensate unknown or varying system parameters and disturbances.

In this work, a mixed control strategy that combines a gravity compensated PD controller with an adaptive RBFNN controller is proposed. The PD+G compensator provides an essential feedback mechanism, guarantee essential system stability with successful tracking action under nominal conditions. On the other hand, the RBFNN part cooperates with the PD+G controller to represent an adaptive section to estimate and compensate for uncertainties, external disturbances, and unmodeled dynamics. This hybrid controller is expected to considerably improve control performance by reducing steady state errors, and improving the system's ability to adapt to uncertain parameters and surrounding disturbances successfully.

3.1 PD Controller with Gravity Compensator

One of the most widely used methods for controlling robotic manipulators is the PD controller. It is preferred in robotic applications due to its simple structure, ease of implementation, and good dynamic performance. Furthermore, the absence of integral term, helps in ensuring stability in manipulator systems and preventing excessive oscillations. The PD control law used for the 2DOF robotic manipulator is expressed as follows:

$$\tau_{PD}(t) = K_p \, e(t) + K_d \, \dot{e}(t) \tag{6}$$

where $\tau_{PD}(t)$ is the controller output torque vector, K_p is the proportional gain matrix, K_d is the derivative gain matrix, $e(t) = \theta_d(t) - \theta(t)$ is the error vector between the desired joint angle vector, $\theta_d(t)$ and the measured joint angle vector, $\theta(t)$. $\dot{e}(t)$ is the time derivative of the position error.

Table 1. Parameters of the 2DOF planar robotic manipulator.

Parameter	Description	Value	Parameter	Description	Value
M_1	Mass of link 1	1.0 kg	I_1	Moment of inertia of link 1	0.02 kgm^2
M_2	Mass of link 2	1.0 kg	I_2	Moment of inertia of link 2	0.02 kgm^2
L_1	Length of link 1	1.0 m	g	Gravitational acceleration	9.81 m/s^2
L_2	Length of link 2	1.0 m	b_1	Viscous friction coefficient at joint 1	0.05 Nms/rad
d_1	Distance to center of mass of link 1	0.5 m	b_2	Viscous friction coefficient at joint 2	0.05 Nms/rad
d_2	Distance to center of mass of link 2	0.5 m			

Table 2. PD controller parameters.

PD parameter	Value	PD parameter	Value
K_p	[49.8072; 18.4379]	K_d	[16.0914; 19.6671]

The proportional part $K_p\ e(t)$ of PD controller is used to reduce the position error to zero, while the derivative part $K_d\ \dot{e}(t)$ is used to deny excessive oscillation and overshoot, resulting in fast and stable response. The PD controller parameters are generally adjusted and can be effectively determined using optimization methods considering the system dynamics and desired control performance. Optimization methods ensure optimal controller tuning while achieving desired performance [23, 24]. The tuned parameters for the PD controller are shown in Table 2.

During robotic arm movement, a large portion of the control torque is used to compensate for nonlinear disturbances such as gravity, inertia, and friction. In case the gravity compensation is not used, when the manipulator is energized, it could fall down due to its weight, causing serious damage to its structure and parts. Furthermore, when trajectory tracking begins, the manipulator must first overcome its own gravity and then gradually approach the reference trajectory [17, 25]. Auxiliary gravity compensator is used to cancel the effect of gravity as much as possible by adding feedforward to the controller torque $G(\theta)$ (or a nominal estimate $\hat{G}(\theta)$) and suppress the remaining error (mainly due to inertial and coupling effects) with simple feedback. The baseline control law, consisting of a PD controller with gravity compensation, is defined as:

$$\tau_{PD+G}(t) = K_p\ e(t) + K_d\ \dot{e}(t) + \hat{G}(\theta) \tag{7}$$

A nominal gravity model $\hat{G}(\theta)$ is used for compensation. Practically, modeling errors or parameter variations may introduce gravity mismatch. Such effects are included in the lumped uncertainty term and are compensated by the adaptive RBFNN component. In this study, the PD controller with gravity compensator (PD+G) (as shown in Figure 2) is used to ensure the basic stability and nominal performance of the system. It is then adapted with a RBFNN to more effectively compensate for system uncertainties and external disturbances.

2.1 Adaptive RBFNN Compensator

Adaptive compensation for unmodeled uncertainties and disturbances in robotic systems is proposed for the aim to improve control performance. In this work, a RBFNN shown in Figure 3 is used to estimate and compensate for these uncertainties in real time. The structure of the RBFNN consists of a set of localized Gaussian basis functions, the output of which is a linear combination of these functions weighted by adaptive weights. The input to the network is $x = [e \quad \dot{e}]^T$, which represents the input vector. It includes the position error $e(t)$ and its time derivative $\dot{e}(t)$, which captures the dynamic state of the tracking error.

Hidden layer consists of five neurons, while output layer has two outputs that represent the torque supplied to the joint actuators. The small number of neurons was selected for the hidden layer in order to keep low computational overhead and enable real time implementation of the controller. Although increasing the number of neurons may improve the approximation accuracy of the neural network, it also increases the computational demand. Therefore, a limited number of neurons was adopted to achieve a balance between control accuracy and computational efficiency. In addition to the selection of neurons number, the centers of the Gaussian basis functions must also be properly selected. The Gaussian centers were nominated to cover the expected range of tracking errors and their derivatives. This configuration was found to provide stable learning behavior during simulations. The Gaussian function for each neuron i is defined as:

$$\emptyset_i(x) = exp\left(-\frac{\|x-c_i\|^2}{2\sigma_i^2}\right) \tag{8}$$

where c_i are the centers of the Gaussian functions, and σ_i is the width (variance) of the i_{th} neuron.

Now, $\tau_{NN}(t) = [\tau_{NN1}(t) \quad \tau_{NN2}(t)]^T$ is the RBFNN output torque vector which will be added to the PD+G controller output. The unknown dynamics, including modeling uncertainties, coupling effects, friction, and external disturbances, are collected into a lumped uncertainty term defined as:

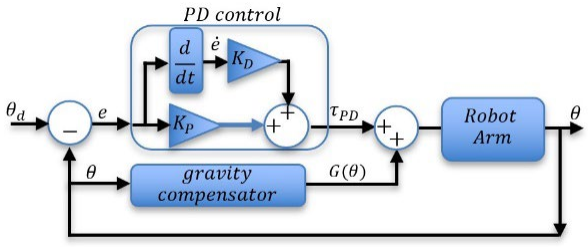


Figure 2. A block diagram of PD+G control of robotic arm.

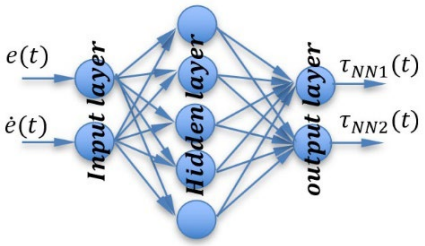


Figure 3. Neural network structure.

$$\Delta(\theta, \dot{\theta}, t) \tag{9}$$

It represents the total unmodeled dynamics affecting the system. The RBF neural network is used to approximate this lumped uncertainty as:

$$\tau_{NN}(t) = W^T \emptyset(x) \tag{10}$$

where W is the vector of adaptive weights that are updated online using an adaptation law as:

$$W(t + \delta t) = W(t) + \alpha e(t) \emptyset(x) \tag{11}$$

with δt is the change in time, and α being the adaptation gain. The error vector $e(t)$ is replaced by a PD combination vector and (11) becomes as follows:

$$W(t + \delta t) = W(t) + \alpha PD(t) \emptyset(x) \tag{12}$$

where $PD(t) = [PD_1(t) \quad PD_2(t)]^T$ is the PD signal which is used instead of $e(t)$ for the following reasons:

- PD takes into account both position and velocity errors, so it gives a better gradient and soft convergence.
- PD has units of torque, so the NN learns the unknown and the disturbing torque; and as a result, the required PD torque effort decreases.

The initial values of the RBF weights (Listed in Table 3) are set based on the researcher's experience and then they will be updated over the system's running process by the update law.

In this work, a joint control law is used, which combines the output of the PD+G controller with an adaptive signal generated by the RBFNN neural network as shown in Figure 4. The final control signal for the two-joint manipulator is described as:

$$\tau = \tau_{PD+G}(t) + \tau_{NN}(t) \tag{13}$$

where $\tau_{PD}(t)$ is the output of the PD controller, $\tau_{NN}(t)$ is the signal from the RBFNN adaptive compensator, and $G(\theta)$ represents the direct compensation of gravity forces. This approach allows for efficient trajectory tracking and compensation for unknown perturbations and model uncertainties.

4.2 Lyapunov Stability Analysis

Lyapunov analysis is used to establish the stability of the proposed PD+G+RBFNN control scheme.

Theorem: Consider the nonlinear dynamics of the 2DOF robotic manipulator given in (1) and the control law defined in (13). Assume the desired trajectory $\theta_d(t)$is smooth with bounded derivatives, i.e., $\| \dot{\theta}_d \| \leq \bar{v}$ and $\| \ddot{\theta}_d \| \leq \bar{a}$ for all $t \geq 0$. If the RBFNN approximation error is bounded and the gain matrices K_d and Γ are positive definite, then all closed-loop signals remain bounded and the tracking error converges to a compact set, whose bound is derived in (36), showing its dependence on the approximation error and control gains.

The joint tracking error and its first and second derivatives are defined as:

$$e(t) = \theta_d(t) - \theta(t)$$

$$\dot{e}(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$$

$$\ddot{e}(t) = \ddot{\theta}_d(t) - \ddot{\theta}(t) \tag{14}$$

and the filtered tracking error is:

$$s(t) = \dot{e}(t) + \Lambda e(t) \tag{15}$$

Table 3. RBF initial setting.

PD parameter	Value
c	$\begin{bmatrix} 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 \end{bmatrix}$
σ	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
α	0.01

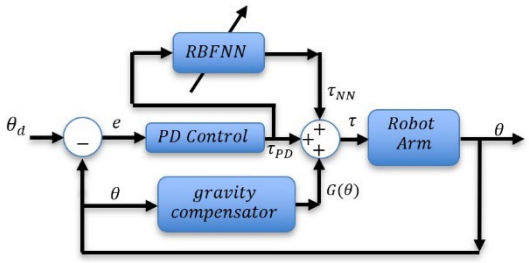


Figure 4. Block diagram of the proposed controller.

Differentiating (15) yields:

$$\dot{s}(t) = \ddot{e}(t) + \Lambda\dot{e}(t) = \left(\ddot{\theta}_d - \ddot{\theta}\right) + \Lambda\dot{e}(t) \tag{16}$$

Equation (16) shows that $\dot{s}$ contains $\ddot{\theta}_d$, which comes from time varying reference trajectories. Now, the following equation can be obtained from (16):

$$\dot{e} = s - \Lambda e \tag{17}$$

Then the actual joint velocity is:

$$\dot{\theta} = \dot{\theta}_d - \dot{e} = (\dot{\theta}_d + \Lambda e) - s \tag{18}$$

Multiplying $\dot{s} = \left(\ddot{\theta}_d - \ddot{\theta}\right) + \Lambda\dot{e}(t)$ by $D(\theta)$ and substituting the manipulator dynamics (1) yields:

$$D(\theta)\dot{s} = D(\theta)\ddot{\theta}_d + D(\theta)\Lambda\dot{e} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + B\dot{\theta} + \tau_d - \tau \tag{19}$$

Substituting (18) into the Coriolis term:

$$C(\theta,\dot{\theta})\dot{\theta} = C(\theta,\dot{\theta})\left[(\dot{\theta}_d + \Lambda e) - s\right] = C(\theta,\dot{\theta})(\dot{\theta}_d + \Lambda e) - C(\theta,\dot{\theta})s \tag{20}$$

and substituting (20) back in (19) gives:

$$D(\theta)\dot{s} = \underbrace{D(\theta)\ddot{\theta}_d + D(\theta)\Lambda\dot{e} + C(\theta,\dot{\theta})(\dot{\theta}_d + \Lambda e) + \left[G(\theta) - \hat{G}(\theta)\right] + B\dot{\theta} + \tau_d}_{\Delta(\theta,\dot{\theta},t)} - C(\theta,\dot{\theta})s - \tau \tag{21}$$

where $\Delta(\theta,\dot{\theta},t)$ represents the lumped uncertainty which includes all terms that are not cancelled by the nominal control law:

$$\Delta(\theta,\dot{\theta},t) = D(\theta)\ddot{\theta}_d + D(\theta)\Lambda\dot{e} + C(\theta,\dot{\theta})(\dot{\theta}_d + \Lambda e) + \left[G(\theta) - \hat{G}(\theta)\right] + B\dot{\theta} + \tau_d \tag{22}$$

This term clearly includes $\dot{\theta}_d$ and $\ddot{\theta}_d$, which implies that the proof is valid for time varying trajectories including sinusoidal references. Under the assumption that $\dot{\theta}_d$ and $\ddot{\theta}_d$ are bounded, and since the closed-loop signals remain in a compact operating set, Δ is bounded thus can be approximated by RBFNN. The universal approximation capability of RBFNNs guarantees that for any continuous, bounded function Δ, on a compact domain, an ideal weight matrix Wcan be found satisfying the following equation:

$$\Delta = W^{*T}\phi(x) + \varepsilon, \| \varepsilon \| \leq \bar{\varepsilon} \tag{23}$$

where W^* is the ideal weight matrix, ε is the bounded approximation residual. Substituting the proposed control law $\tau = K_d s + \hat{G}(\theta) + W^T\phi(x)$ into error dynamics (21):

$$\begin{aligned} D(\theta)\dot{s} &= W^{*T}\phi + \varepsilon - C(\theta,\dot{\theta})s - K_d s - W^T\phi \\ &= -K_d s - C(\theta,\dot{\theta})s - \underbrace{(W - W^*)^T}_{=\tilde{W}^T}\phi(x) + \varepsilon \end{aligned} \tag{24}$$

where $\tilde{W} = W - W^*$ is the weight estimation error.

Select the following function as a Lyapunov function candidate:

$$V = \frac{1}{2}s^T D(\theta)s + \frac{1}{2}\left(\tilde{W}^T\Gamma^{-1}\tilde{W}\right) \tag{25}$$

Since $D(\theta)$ and Γ are both positive definite, $V > 0$ for all $(s, \widetilde{W}) \neq (0,0)$. Differentiating Lyapunov function in (25) leads to:

$$\dot{V} = s^T D(\theta)\dot{s} + \frac{1}{2} s^T \dot{D}(\theta) s + \widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}} \tag{26}$$

Substituting (24) in (26) yields:

$$\dot{V} = s^T \left[-K_d s - Cs - \widetilde{W}^T \phi + \varepsilon\right] + \frac{1}{2} s^T \dot{D} s + \widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}}$$

$$= -s^T K_d s - \underbrace{s^T C s + \frac{1}{2} s^T \dot{D} s}_{= 0 \text{ (skew-symmetry)}} - s^T \widetilde{W}^T \phi + s^T \varepsilon + \widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}} \tag{27}$$

where $\dot{D}(\theta) - 2C(\theta, \dot{\theta})$ is skew symmetric according to the standard property of robot manipulators, meaning that:

$$s^T \left(\dot{D} - 2C\right) s = 0 \implies \frac{1}{2} s^T \dot{D} s = s^T C s \tag{28}$$

Therefore (27) becomes:

$$\dot{V} = -s^T K_d s - s^T \widetilde{W}^T \phi + s^T \varepsilon + \widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}} \tag{29}$$

The continuous time adaptation law is:

$$\dot{W} = \Gamma \, \phi(x) \, s^T \tag{30}$$

Since W^* is constant, $\dot{\widetilde{W}} = \dot{W} = \Gamma \phi s^T$,

$$\widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}} = \widetilde{W}^T \Gamma^{-1} \Gamma \, \phi \, s^T = \widetilde{W}^T \phi \, s^T = s^T \widetilde{W}^T \phi \tag{31}$$

Substituting back into (29), the weight estimation terms cancel exactly:

$$\dot{V} = -s^T K_d s - s^T \widetilde{W}^T \phi + s^T \widetilde{W}^T \phi + s^T \varepsilon \tag{32}$$

$$\dot{V} = -s^T K_d s + s^T \varepsilon \tag{33}$$

Applying the Cauchy Schwarz inequality $s^T \varepsilon \leq \| s \| \bar{\varepsilon}$ makes (33) as follows:

$$\dot{V} \leq -\lambda_{\min}(K_d) \| s \|^2 + \| s \| \bar{\varepsilon} \tag{34}$$

Therefore $\dot{V} < 0$ whenever

$$\| s \| > \frac{\bar{\varepsilon}}{\lambda_{\min}(K_d)} \tag{35}$$

This establishes uniform ultimate boundedness (UUB) of $s(t)$. Since $s = \dot{e} + \Lambda e$ and Λ is positive definite, s being bounded implies $e(t)$ and $\dot{e}(t)$ are also bounded. The tracking error converges to:

$$\mathrm{E} = \left\{ e : \| e \| \leq \frac{\bar{\varepsilon}}{\lambda_{\min}(K_d) \, \lambda_{\min}(\Lambda)} \right\} \tag{36}$$

The size of this set shrinks as K_d or the RBFNN approximation accuracy improves. Furthermore, the adaptive weights $W(t)$ remain bounded for all time.

4. SIMULATION RESULTS

To verify the effectiveness of the proposed PD+G controller with an RBFNN compensator, detailed simulations were conducted in the simulation environment. The manipulator model and the proposed control algorithm were implemented taking into account nonlinear dynamics, gravity forces, and viscous friction. Step signal and sine function signals were used to examine the tracking quality of the target trajectory, allowing for both the response to sudden changes and continuous tracking of reference inputs. Simulation setting, such as time step, and initial conditions, were selected to mimic real and stable test conditions. Joint angles were observed and recorded for further analysis.

To evaluate the controller's effectiveness, comparative simulations were also conducted using the PD+G controller, demonstrating the improvement in control quality after adding the adaptive RBFNN element. In order to quantitatively evaluate the controller performance, ITAE, ISE, IAE, RMSE indices and the transient characteristics of the unit step response were used in this study. The performance indices are expressed by the following equations:

$$ITAE = \int_0^T t|e(t)|dt \tag{37}$$

$$ISE = \int_0^T e^2(t)dt \tag{38}$$

$$IAE = \int_0^T |e(t)|dt \tag{39}$$

$$RMSE = \sqrt{\frac{1}{T}\int_0^T e^2(t)dt} \tag{40}$$

where t represents time and $e(t)$ represents the error (the difference between the reference and the system output).

4.1 Step Response Test

In this test, a step input is applied to both joints θ_1 and θ_2, with staggered start up times in order to highlight the dynamic coupling effect between the two joints. This configuration makes it possible to observe how the control of each joint influences the other. The two control strategies, the classical PD regulator with gravity compensator and the proposed controller combining PD+G with an adaptive compensator by RBFNN, were evaluated under these conditions. The results in Table 4 show that the PD+G controller has longer settling time, with higher overshoot values, in particular on the first joint.

The recorded performance indices in Table 4 explicitly shows the superiority of the proposed controller in terms of lower ITAE, ISE, IAE, and RMSE. A noticeable coupling effect when using PD+G controller is shown in Figure 5. When θ_1 start to follow the desired angle at $t = 0$, there is a noticeable effect on θ_2, and when θ_2 start to follow its desired at $t = 5$ sec, θ_1 is disturbed from its steady value. On the other hand, the combined PD+G with RBFNN controller significantly overcome the coupling effect between the joints' angles by providing the required control signal. These observations demonstrate that the addition of the adaptive compensator allows the system to better compensate for cross dynamics, thus ensuring a better overall performance and a better robustness.

4.2 Sinusoidal Pattern Test

In the second test, sinusoidal signals were used as the desired angles of the joints. They are defined as:

$$\theta_{1_{desired}} = 0.5\sin(2\pi t) \tag{41}$$

$$\theta_{2_{desired}} = 0.5\sin(2\pi t + \pi/2) \tag{42}$$

This kind of input allows to assess the ability of the controller to track continuous, time-varying trajectories, which is important in real robotic applications, where manipulative movements are not one-time jumps, but smooth and dynamic changes.

Simulation result for this test is shown in Figure 6, while the performance indices are reported in Table 5. The analysis of the quantitative results for both controllers showed that the PD+G controller with RBFNN compensator achieves lower ITAE, ISE, IAE, and RMSE values compared to the PD+G controller itself. This means that the proposed controller provides better tracking quality, reducing tracking error over time, and consistent accuracy. The sinusoidal test therefore confirms the proposed controller's ability to cope with dynamic trajectory changes, which is important for applications requiring precise manipulator movement.

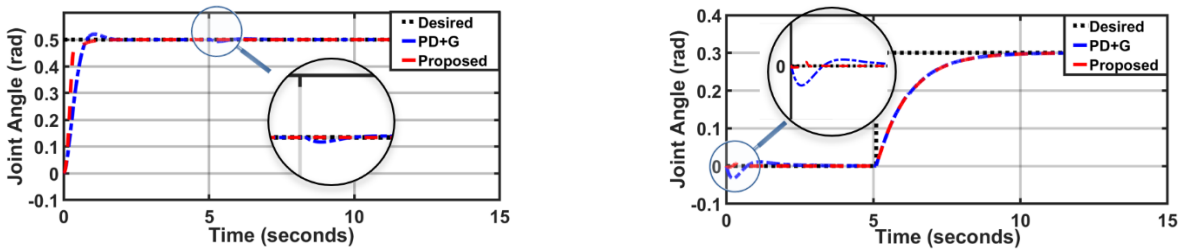
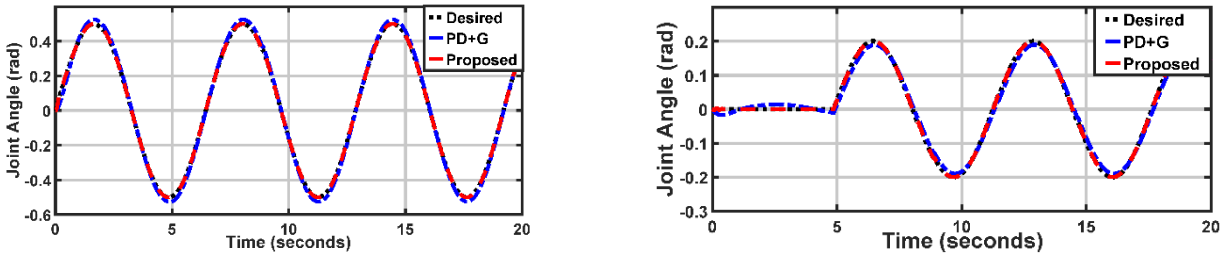


Figure 5. Step response plots for the joints angle: (a) Joint θ_1; (b) Joint θ_2.

Table 4. Performance comparison for step trajectory tracking.

Controller	Joint Angle	Settling Time (s)	Overshoot (%)	ITAE	ISE	IAE	RMSE
PD+G	θ_1 θ_2	1.382 4.125	4.14 0.00	2.038	0.108	0.534	0.085
Proposed	θ_1 θ_2	0.822 4.168	0.07 0.00	1.963	0.094	0.4745	0.079

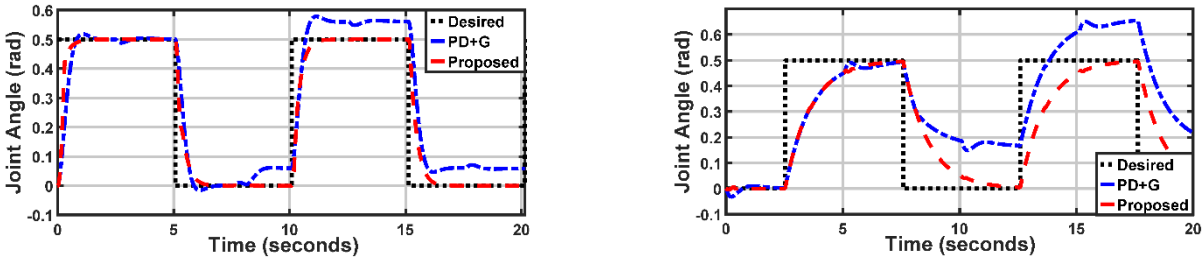


(a) (b)

Figure 6. Joints angle response for sinusoidal pattern test: (a) Joint θ_1; (b) Joint θ_2.

Table 5. Performance comparison for sinusoidal trajectory tracking.

ITAE		ISE		RMS error		IAE	
PD+G	Proposed	PD+G	Proposed	PD+G	Proposed	PD+G	Proposed
5.06	0.35	0.0096	0.0003	0.022	0.004	0.531	0.049



(a) (b)

Figure 7. Joints angle response under disturbance test: (a) Joint θ_1; (b) Joint θ_2.

Table 6. Performance comparison for disturbance rejection test.

ITAE		ISE		RMS error		IAE	
PD+G	Proposed	PD+G	Proposed	PD+G	Proposed	PD+G	Proposed
52.47	27.04	1.109	0.724	0.236	0.190	4.46	2.698

4.3 Disturbance Rejection Test

The third test involved applying a disturbance torque in the form of a step signal with an amplitude of $[3, 3]$ N.m. at $t = 8$ sec, which is added to the control signal supplied to the manipulator model. This test aims to evaluate the ability of the control system to cope with external unexpected disturbances. The results in Figure 7 show that the proposed combined controller effectively overcomes the effects of perturbation, restoring accurate trajectory tracking. At the same time, the classical PD+G controller cannot cope with disturbances, which manifests itself in significant deviations from the set position and a deterioration in control quality.

The effect of cross coupling between the joints is clear when using PD+G controller. After the step input on θ_2 applied, the response of θ_1 slightly changes, indicating feedback effects. In contrast, the proposed hybrid controller, which includes an adaptive compensator, attenuates the effects of this coupling. It better compensates for the inter joint dynamics, resulting in smoother and more accurate tracking for both joints.

This test confirms that the use of an adaptive neural network compensator significantly improves the stability of the system and its ability to withstand unexpected external influences. These findings are demonstrated numerically by the performance indices in Table 6. It shows the prevalence of the proposed controller across all performance indices, which further confirms the controller superior tracking accuracy, faster convergence, and enhanced overall control performance compared to the benchmark methods.

4.4 Model Uncertainty Test

In this test, arbitrary changes are made to the dynamic model parameters of the robot arm and compared the system performance with both the conventional PD+G controller and the proposed adaptive controller. The variations in parameters range from 10% to 50% as listed in Table 7. The results shown in Figure 8 reveal that the PD+G controller is more sensitive to these parameter changes, resulting in higher errors and poorer tracking performance.

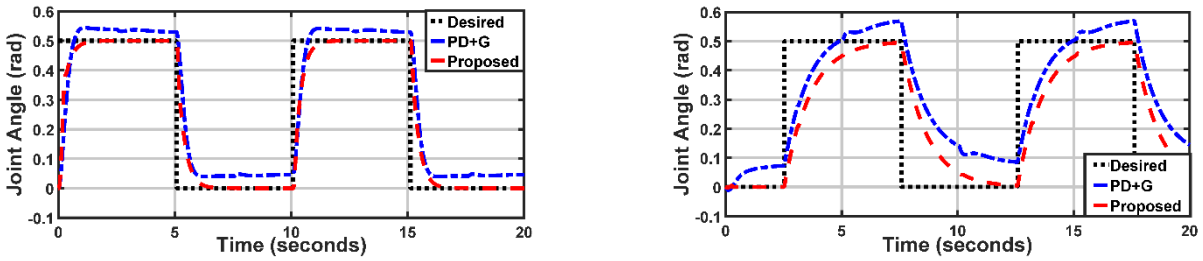


Figure 8. Joints angle response under model uncertainty test: (a) Joint θ_1; (b) Joint θ_2.

Table 7. System parameters: nominal versus perturbed values.

Parameter	Nominal	Perturbed	Parameter	Nominal	Perturbed
M_1	1.0 kg	1.2 kg	d_2	0.5 m	0.4 m
M_2	1.0 kg	0.8 kg	I_1	0.02 kg. m^2	0.025 kg. m^2
L_1	1.0 m	0.9 m	I_2	0.02 kg. m^2	0.015 kg. m^2
L_2	1.0 m	1.2 m	b_1	0.05 N. m. s/rad	0.035 N. m. s/rad
d_1	0.5 m	0.6 m	b_2	0.05 N. m. s/rad	0.025 N. m. s/rad

Table 8. Performance comparison for model uncertainty test.

ITAE		ISE		RMS error		IAE	
PD+G	Proposed	PD+G	Proposed	PD+G	Proposed	PD+G	Proposed
41.74	27.05	0.885	0.701	0.210	0.188	3.939	2.684

On the other hand, the proposed controller, thanks to its adaptive part, better compensates for model uncertainties, thus demonstrating more stable system behavior with lower tracking errors. In parallel, the calculated ITAE, IAE, ISE, and RMSE values support this conclusion as illustrated quantitatively by Table 8. This demonstrates that the proposed controller significantly improves system performance by reducing both the total error and the persistence of the error. Consequently, despite the uncertainties in the model parameters, the proposed controller provides more reliable and precise control of the robot arm.

4.5 Performance Summary

To provide a plain comparison between the conventional PD+G controller and the proposed PD+G+RBFNN controller, the percentage of improvement in the main performance indices through all test scenarios are summarized in Table 9. The quantitative results presented in Table 9 demonstrate the effectiveness of the proposed controller under different operating conditions.

For the step tracking, the improvements are relatively simple, which is expected since step inputs represent a relatively simple control task where even classical controllers can achieve acceptable performance. Nevertheless, the proposed method still provides good improvement across all indices, indicating enhanced transient response and reduced overall error. Concerning sinusoidal tracking, the proposed controller shows a significant enhancement, that exceeding 90% in ITAE, ISE, and IAE, and reaching 81.82% in RMSE. This highlights the strong capability of the controller in handling time varying continuous desired trajectories, where the adaptive neural network compensates for dynamic nonlinearities and modeling uncertainties. For disturbance rejection, the improvements are also substantial, indicating better error suppression and faster recovery after disturbance application. Under model uncertainty, the proposed controller maintains robust performance, achieving improvements up to 35.19% (ITAE) and 31.86% (IAE). This confirms the controller robustness against parameter variations.

Table 9. Final performance summary table.

Test	% Improvement			
	ITAE	ISE	IAE	RMSE
Step response	3.68	12.96	11.14	7.06
Sinusoidal tracking	93.08	96.88	90.77	81.82
Disturbance rejection	48.47	34.70	39.50	19.49
Model uncertainty	35.19	20.79	31.86	10.48

5. CONCLUSION

Systematic simulations with overlapped time-shifted step signals (to expose coupling between joints), sinusoidal references, perturbations on the torque, and model deviations permanently show that PD+G along with RBFNN controller provides lower tracking errors, shorter settling time of the first link with nearly identical settling performance for the second link, and less overshoot than a traditional PD+G controller. This is quantitatively confirmed with several indices including ITAE, ISE, IAE, and RMSE where the adaptive NN compensator effectively lessen the effect of uncertain components that the PD part cannot handle alone. Furthermore, tests with deviating parameter values illustrate robustness to model errors. The strength of the method lies in taking into account joint coupling and treating it as a disturbance-like. The RBFNN motivated adaptation law ensures well defined updates of NN weights and contributes to stable tracking in the presence of uncertainties. The limitations mainly concern the choice of control and adaptation parameters (e.g. PD gain and learning rate), which can affect transients and control effort. Future work may include experimental validation on a physical platform and extension to higher DOF. Overall, the results show that the proposed strategy constitutes a robust and practical alternative to classical PD control for nonlinear, coupled manipulators with model and disturbance uncertainties.

In practical implementations, the delay in transmitting signals between sensors, controllers, and actuators may degrade the control performance of the proposed scheme. In the present work, ideal signal transmission without delay is considered. However, for real world applications, time delays can degrade tracking accuracy and even lead to unstable closed loop system if not properly addressed. Addressing time delay effects in the proposed PD+G+RBFNN scheme opens a valuable direction for future experimental work.

ACKNOWLEDGEMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

[1] Husainy, S. Mangave and N. Patil, A review on robotics and automation in the 21st century: Shaping the future of manufacturing, healthcare, and service sectors, *Asian Review of Mechanical Engineering*, 12, 2023, 41–45.

[2] Alkamachi and Y. G. K. Abboosh, Modelling and control of cable driven robotic arm using Maplesim, *Advances in Electrical and Electronic Engineering*, 22, 2024, 271–280.

[3] J. Wu and C. Peng, Observer-based adaptive event-triggered PID control for networked systems under aperiodic DoS attacks, *International Journal of Robust and Nonlinear Control*, 32, 2022, 2536–2550.

[4] Zhao, D. Wang and W. Xue, Beyond linear limits: Design of robust nonlinear PID control, *Automatica*, 173, 2025, 112075.

[5] Q. Zhou, S. Zhao, H. Li, R. Lu and C. Wu, Adaptive neural network tracking control for robotic manipulators with dead zone, *IEEE Transactions on Neural Networks and Learning Systems*, 30, 2019, 3611–3620.

[6] S. Ajwad, O. Iqbal and J. Iqbal, Hardware realization and PID control of multi-degree of freedom articulated robotic arm, *Mehran University Research Journal of Engineering and Technology*, 34, 2015, 1–12.

[7] M. S. Ahmed, A. H. M. Mary and H. H. Jasim, Robust computed torque control for uncertain robotic manipulators, *Al-Khwarizmi Engineering Journal*, 17, 2021, 22–28.

[8] X. Yang, Z. Zhao, Y. Li, G. Yang, J. Zhao and H. Liu, Adaptive neural network control of manipulators with uncertain kinematics and dynamics, *Engineering Applications of Artificial Intelligence*, 133, 2024, 107935.

[9] L. A. Soriano, E. Zamora, J. M. Vazquez-Nicolas, G. Hernández, J. A. Barraza Madrigal and D. Balderas, PD control compensation based on a cascade neural network applied to a robot manipulator, *Frontiers in Neurorobotics*, 14, 2020, 577749.

[10] Y. Wu, R. Huang, X. Li and S. Liu, Adaptive neural network control of uncertain robotic manipulators with external disturbance and time-varying output constraints, *Neurocomputing*, 323, 2019, 108–116.

[11] Q. Liu, D. Li, S. S. Ge, R. Ji, Z. Ouyang and K. P. Tee, Adaptive bias RBF neural network control for a robotic manipulator, *Neurocomputing*, 447, 2021, 213–223.

[12] X. Shen, K. Zhou, R. Yu and B. Wang, Design of adaptive RBFNN and computed-torque control for manipulator joint considering friction modeling, *International Journal of Control, Automation and Systems*, 20, 2022, 2340–2352.

[13] J. Narayan, M. Abbas, B. Patel and S. K. Dwivedy, Adaptive RBF neural network-computed torque control for a pediatric gait exoskeleton system: An experimental study, *Intelligent Service Robotics*, 16, 2023, 549–564.

[14] Y. Han, H. Ma, Y. Wang, D. Shi, Y. Feng, X. Li, Y. Shi, X. Ding and W. Zhang, Neural network robust control based on computed torque for lower limb exoskeleton, *Chinese Journal of Mechanical Engineering*, 37, 2024, 37.

[15] K. Guo and Y. Pan, Composite adaptation and learning for robot control: A survey, *Annual Reviews in Control*, 55, 2023, 279–290.

[16] H. Zhang, Y. Zhao, Y. Wang and L. Liu, Adaptive neural network control of robotic manipulators with input constraints and without velocity measurements, *IET Control Theory & Applications*, 18, 2024, 1232–1247.

[17] H. Zhang and W. Bu, H. Zhang, M. Du, G. Wu and W. Bu, PD control with RBF neural network gravity compensation for manipulator, *Engineering Letters*, 26, 2018, 236–244.

[18] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, New York: Wiley, 2020.

[19] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson Education India, 2009.
[20] W. K. Sa'id, B. I. Kazem and A. M. Manaty, Hybrid controller for a single flexible link manipulator, *The Journal of Engineering*, 18, 2012, 1242–1254.
[21] K. H. Mahmoud, A.-N. Sharkawy and G. T. Abdel-Jaber, Development of safety method for a 3-DOF industrial robot based on recurrent neural network, *Journal of Engineering and Applied Science*, 70, 2023, 44.
[22] Z. Téczely and B. Kiss, Switching linear parameter varying control for a robotic manipulator, *Acta IMEKO*, 14, 2025, 1–8.
[23] P. Kulkarni, O. Kulkarni and J. K. Sayyad, Tuning of a robotic arm using pid controller for robotics and automation industry, *6th International Conference on Energy, Power and Environment (ICEPE)*, Shillong, India, 2024, 1–6.
[24] A. Tawfeeq, M. Y. Salloom and A. Alkamachi, A self-balancing platform on a mobile car, *International Journal of Electrical and Computer Engineering (IJECE)*, 12, 2022, 5911– 5922.
[25] J. Huang, C. Yang and J. Ye, Nonlinear PD controllers with gravity compensation for robot manipulators, *Cybernetics and Information Technologies*, 14, 2014, 141–150.