Dynamic Modeling and PID Control of a 6-DOF Robotic Arm Using ROS and Gazebo

Ahmad Riyad Firdaus[1*], Cyrillus Rudi Soru[1], and Hendawan Soebhakti[1]

[1]Robotics Engineering Technology Study Program, Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia

[*]Email: rifi@polibatam.ac.id

Received on 15-09-2025 | Revised on 14-10-2025 | Accepted on 10-12-2025

***Abstract*—This paper presents the dynamic modeling and control evaluation of a six degrees-of-freedom (6-DOF) robotic manipulator. The manipulator was developed in the Robot Operating System (ROS) and Gazebo using a detailed URDF model with complete geometric and inertia parameters. Proportional–Integral–Derivative (PID) controllers were tuned through ROS dynamic reconfiguration and tested under four payloads: 0; 0,19; 0,39; and 0,50 kg. Controller performance was assessed using rise time, settling time, overshoot, and steady-state error. The results show stable responses across all conditions, with no overshoot and near-zero steady-state errors. Increasing payloads generally led to longer rise and settling times, while joints aligned with gravity exhibited faster responses under heavier loads. These findings confirm that properly tuned PID controllers can maintain robust and accurate manipulator performance and demonstrate the effectiveness of ROS–Gazebo as an open-source platform for robotic control experimentation and future integration of adaptive or AI-based methods.**

Keywords: Gazebo, PID Control, Robotic Manipulator, ROS, URDF.

I. INTRODUCTION

Robotic manipulators are central to modern industrial automation, performing repetitive and high-precision tasks such as assembly, packaging, welding, and sorting. Six-degree-of-freedom (6-DOF) arms are particularly significant because they provide the dexterity required for complex spatial movements, closely mimicking human arm functionality [1]. However, their nonlinear dynamics, joint coupling, and variable payloads make the design of stable and accurate control systems challenging. Among available strategies, Proportional–Integral–Derivative (PID) controllers remain the industry standard due to their simplicity, reliability, and computational efficiency [2]. Yet, tuning PID gains for each joint in a multi-DOF manipulator remains difficult, since dynamic behavior varies across joints depending on inertia, gravity, and load distribution.

Simulation frameworks have become critical in reducing the cost and risk of developing robotic systems. The Robotic Operating System (ROS) provides middleware, libraries, and standardized interfaces for robot control, while the Gazebo simulator offers a physics-based 3D environment for visualizing and testing robotic systems [3]. When combined, ROS and Gazebo form a powerful prototyping environment where models described in the Unified Robot Description Format (URDF) can be integrated with PID controllers, tested under various payload conditions, and tuned in real-time without risking hardware damage [4].

Recent research has demonstrated the effectiveness of this approach. Megalingam et al. [5] developed and simulated a 6-DOF robotic manipulator in ROS–Gazebo with PID control, demonstrating the feasibility of this framework for end-to-end control validation. Zhang et al. [6] evaluated different control strategies for a 7-DOF robotic arm in ROS–Gazebo, benchmarking accuracy, efficiency, and robustness under disturbances. Mengacci et al. [7] introduced an open-source toolbox for articulated robots with compliant actuators, extending ROS–Gazebo's applicability to elastic-joint dynamics. Soleimani Amiri et al. [8] integrated particle swarm optimization (PSO) with PID control in ROS–Gazebo, confirming that intelligent tuning methods improve trajectory tracking. Chen et al. [9] proposed a refined PID with fuzzy adaptation and dead-zone compensation for a 6-DOF manipulator, effectively mitigating jitter and disturbances in ROS–Gazebo simulations. More recently, a 2024 study in *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)* reinforced the adoption of ROS–Gazebo as the standard for manipulator simulation and control testing [10].

Despite these advances, two research gaps remain. First, most studies focus on trajectory planning or hybrid control, with limited attention to systematic joint-level PID stability analysis under varying payloads. Second, few works have quantitatively evaluated PID performance using rise time, settling time, overshoot, and steady-state error in a consistent framework for 6-DOF manipulators. This study addresses these gaps by developing a URDF-based dynamic model of a 6-DOF robotic arm in ROS–Gazebo, implementing per-joint PID controllers with real-time gain tuning, and evaluating their performance under different payload conditions. The key contributions of this work are as follows:

1) Development of a 6-DOF robotic arm simulation model integrated into ROS–Gazebo.
2) Implementation and tuning of PID controllers for each manipulator joint.

3) Quantitative evaluation of PID performance across payload variations.
4) Demonstration of a replicable simulation-driven workflow that can reduce risks and costs in industrial manipulator design.

The rest of this paper is organized as follows. Section 2 presents the methodology, including model development, simulation setup, and PID tuning. Section 3 discusses simulation results and performance analysis. Section 4 concludes the study and outlines directions for future research.

II. METHOD

The methodology of this study is designed as a rigorous and reproducible framework for analyzing and controlling a six degrees-of-freedom (6-DoF) robotic manipulator in the ROS–Gazebo environment. It comprises the derivation of the manipulator's dynamic model, performance evaluation under varying payloads, design and tuning of joint-level PID controllers, configuration of the simulation environment, and development of a URDF-based robot model. This structured approach ensures a logical progression from theoretical formulation to simulation-based validation, enhancing both the analytical accuracy of the control design and its practical reliability.

A. Dynamic System for a 6-DoF Manipulator Robot

The robotic system analyzed in this study is a six-degree-of-freedom (6-DoF) serial manipulator composed of revolute joints. Its motion is characterized by nonlinear rigid-body dynamics, which are mathematically formulated in the joint space as represented in Equation (1) [11]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_f = \boldsymbol{\tau}_{cmd} + \boldsymbol{d} \quad (1)$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\mathbf{q}$ denotes Coriolis and centrifugal effects, $\mathbf{g}(\mathbf{q})$ is the gravity vector, $\boldsymbol{\tau}_f$ represents joint friction, $\boldsymbol{\tau}_{cmd}$ is the commanded control torque, and $\boldsymbol{d}$ accounts for external disturbances such as payloads. Friction within the system is represented as a combination of viscous and Coulomb components, as formulated in Equation (2) [12]:

$$\boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{F}_c\,\mathrm{sign}(\dot{\mathbf{q}}) \quad (2)$$

Payloads attached to the end-effector exert an additional wrench $\mathbf{w}_e = [\mathbf{F}_e^T, \boldsymbol{\mu}_e^T]^T$, which is mapped to joint space through the manipulator Jacobian as represented in equation (3) [13]:

$$\boldsymbol{\tau}_{Payload}(\mathbf{q}) = \mathbf{J}^T(\mathbf{q})\mathbf{w}_e \quad (3)$$

In this research, payloads of 0,19 kg, 0,39 kg, and 0,50 kg were modeled and applied at the gripper. This formulation provides the foundation for subsequent control analysis and design.

B. Dynamic System Analysis

Dynamic analysis was conducted to evaluate the manipulator's performance under different payload conditions and to determine the required PID tuning parameters. Each of the six joints was commanded to execute step and ramp trajectories both with and without the payloads. The reference signals, actual responses, and error values were logged using `rostopic` and visualized in real-time with `rqt_plot`.

The joint error is defined as $e_i(t)=q_{r,i}(t)-q_i(t)$, where $q_{r,i}$ is the reference trajectory and q_i is the actual joint position. The performance of the system was then quantified using classical time-domain indices [14], [15]:

1) Rise time (t_r) — the time taken for the response to rise from 10% to 90% of the reference value.
2) Settling time (t_s) — the time required for the response to remain within a 2% error band.
3) Percent overshoot (M_p) — the maximum peak deviation relative to the reference input.
4) Steady-state error (e_{ss}) — the error between reference and actual value as $t\to\infty$.

These indices are widely used in manipulator studies employing ROS–Gazebo to evaluate controller robustness under dynamic variations [14], [16].

C. PID Controller Design and Tuning

The control of each joint was implemented using the `effort_controllers/JointPositionController` provided in the ROS `ros_control` framework. The controller applies a classical Proportional–Integral–Derivative (PID) law to regulate the error between the reference joint trajectory $q_{r,i}(t)$and the actual joint position $q_i(t)$. The error is defined according to the formulation presented in Equation (4):

$$e_i(t) = q_{r,i}(t) - q_i(t) \quad (4)$$

The control torque $\tau_i(t)$ applied to joint i is is formulated as shown in Equation (5) [17]:

$$\tau_i(t) = K_{P,i}e_i + K_{i,i}\int_0^t e(\xi)d\xi + K_{d,i}\dot{e}_i(t) \quad (5)$$

where $K_{p,i}$, $K_{i,i}$, and $K_{d,i}$ denote the proportional, integral, and derivative gains, respectively. Each term serves a distinct function:

1) The proportional term $K_{p,i}e_i(t)$ provides an immediate correction proportional to the error, improving responsiveness.
2) The integral term $K_{i,i}\int e_i\,dt$ eliminates steady-state error by accumulating past errors.
3) The derivative term $K_{d,i}\dot{e}_i(t)$ anticipates future error by responding to its rate of change, thus improving system damping and reducing overshoot.

To mitigate noise amplification in the derivative channel, a low pass filtered derivative was used as presented in Equation (6):

$$\dot{e}_i^f(s) = \frac{N}{1+Ns}sE_i(s) \quad (6)$$

where N = 25 is the derivative-filter coefficient chosen as a trade-off between responsiveness and noise rejection. This formulation ensures that high-frequency noise does not

destabilize the controller, while still providing the predictive benefits of derivative action.

Furthermore, an integral anti-windup mechanism was implemented by constraining the integral term, as described in Equation (7):

$$I_i[k] = clip\big(I_i[k-1] + K_{i,i}T_s e_i[k], I_{min}, I_{max}\big) \quad (7)$$

where $T_s = 0.001s$ is the sampling time and the clipping bounds (I_{min}, I_{max}) as listed in Table 1, restrict the integral component from accumulating excessively during actuator saturation. This approach enhances system stability and enables quicker recovery when the actuator returns to normal operation.

TABLE I
CLIPPING BOUND

Joint	(I {min})	(I {max})
art1_yaw	−1,5	+1,5
art2_pitch	−2,0	+2,0
art3_pitch	−1,5	+1,5
art4_roll	−1,0	+1,0
art5_pitch	−1,0	+1,0
art6_roll	−0,8	+0,8

These limits were set such that the integral effort contributed no more than 25–35% of total control torque during transients, ensuring fast convergence without overshoot. For discrete-time implementation, as utilized in ROS, the PID control law is represented in Equation (8) as follows:

$$\tau_i[k] = K_{p,i}e_i[k] + K_{i,i}T_s\sum_{j=0}^{k} e_i[j] + K_{d,i}\frac{e_i[k] - e_i[k-1]}{T_s} \quad (8)$$

This formulation updates control commands at each simulation timestep, making it compatible with the real-time execution of Gazebo's physics engine.

Gain tuning was performed interactively using the Dynamic Reconfigure plugin in the `rqt_gui` package, enabling real-time modification of K_p, K_i, and K_d while monitoring error responses on `rqt_plot`. The tuning process was iterated until minimal steady-state error, reduced overshoot, and acceptable settling time were achieved without actuator saturation [18].

D. Robot Modeling and Simulation Environment

All experiments were conducted in a fully open-source environment to ensure reproducibility and extensibility. The operating system used was Ubuntu 18.04.5 (Bionic Beaver), with ROS Melodic Morenia serving as the middleware and Gazebo 9.0.0 as the physics simulator. A dedicated Catkin workspace (`catkin_ws`) was configured to manage all necessary packages, including the URDF robot description, controller configuration files, and launch scripts. This modular configuration streamlined package management and enabled seamless integration of modeling, simulation, and control, following widely adopted ROS–Gazebo practices in robotic manipulator research [16].

The manipulator was modelled using the Unified Robot Description Format (URDF) with the `.xacro` extension to enhance modularity. Each link was defined using STL geometry files and annotated with mass and inertia properties to reflect real-world dynamics. The complete manipulator weighed 2,15 kg without the gripper and 2,27 kg with the gripper attached. Several Gazebo plugins were integrated to ensure realistic physics behavior, including:

1) `gazebo_ros_control` for interfacing ROS controllers with the physics engine,
2) `gripper_mimic_joint_plugin` for synchronized gripper claw motion, and
3) `gazebo_grasp_fix` for stable grasping and reduced object-slippage during manipulation.

Payload objects of 0,19 kg, 0,39 kg, and 0,50 kg were also modelled and attached to the end-effector to evaluate performance under varying load conditions.

The robot model was implemented within the `thor_arm package` in the `catkin_ws` workspace. As illustrated in Figures 1 and 2, the manipulator consists of six articulated links connected through revolute joints, supported by a gripper assembly for object manipulation.

1) Link and Mass Properties**:** The base link (0,6076 kg, black) anchors the system, while articulated links (`art1–art6`) range from 0,028–0,592 kg, alternating in black and blue for visualization. The gripper assembly comprises the mount (0,058 kg), mount cover (0,010 kg), and claws (`claw_l` 0,0329 kg, `claw_r` 0,0192 kg).
2) Joint Configuration: The manipulator uses six revolute joints providing yaw, pitch, and roll motion. The gripper joints (`gripper_l` and `gripper_r`) are configured such that gripper_l acts as a mimic joint, producing symmetric but opposite motion relative to `gripper_r`. Fixed joints are used to attach the gripper mount and cover.

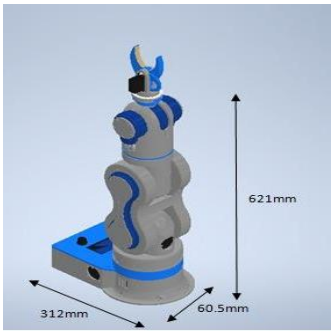

Figure 1. Robot Model

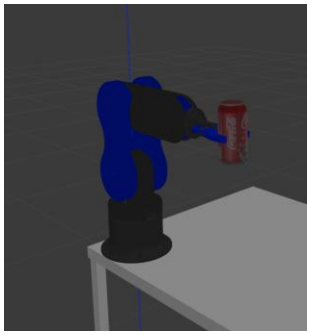
Figure 2. Robot Model with Payload

This design ensures realistic modeling of manipulator dynamics while providing an efficient and computationally lightweight framework for simulating end-effector functionality. The mimic joint mechanism allows accurate simulation of gripper opening and closing, which is critical for handling payloads of varying shapes and sizes.

III. RESULT AND DISCUSSION

A. PID Tunning Outcomes

The tuning of PID parameters for each joint was performed using the ROS Dynamic Reconfigure tool in `rqt_gui`. The optimal parameters are summarized in Table 2, where proportional gains (P) ranged from 500 to 2000, integral gains (I) were set between 50 and 1000, and derivative gains (D) ranged from 200 to 1000. These values provided a balance between fast response and stability, ensuring smooth joint trajectories without inducing oscillations.

TABLE I
PARAMETER PID

No	Nama Joint	P	I	D
1	art1_yaw	500	100	500
2	art2_pitch	500	1000	1000
3	art3_pitch	500	50	1000
4	art4_roll	2000	50	200
5	art5_pitch	2000	50	200
6	art6_roll	2000	50	200

B. Joint Response under Variable Payloads

1) Joint 1

Simulation of Joint 1 (Figure 3) demonstrated consistent performance under four payload conditions (0; 0,19; 0,39; and 0,50 kg). Rise time and settling time increased slightly with heavier payloads (from 3,77 s and 4,90 s without payload to 3,84 s and 5,16 s with 0.50 kg). Importantly, no overshoot was observed across all tests, and steady-state errors remained near zero (10^{-6} order). This indicates the PID controller effectively handled load variations while preserving precision — a critical requirement for high-accuracy manipulation tasks.

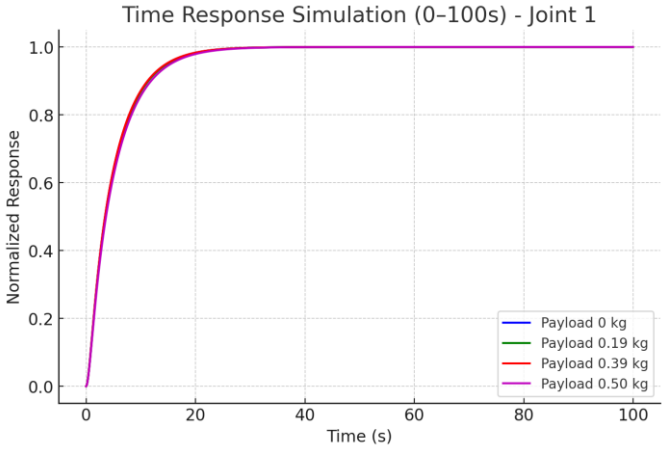

Figure 3. Time Response for Joint-1

2) Joint 2

Joint 2 experienced the highest mechanical stress as it supported the majority of the arm's weight and payload, as shown in Figure 4. Rise times increased with load, ranging from 5,2 s (0,19 kg) to 6,88 s (0,50 kg). Settling time also extended to 13,88 s under maximum load. Despite this, no overshoot occurred, and steady-state error remained very small (~10^{-5}). This demonstrates the robustness of the controller in stabilizing heavy-load joints, although at the cost of slower convergence due to gravitational effects.

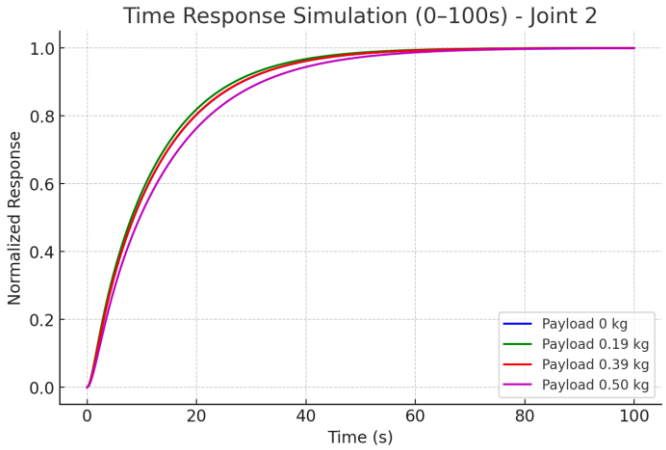

Figure 4. Time Response for Joint-2

3) Joint 3

As illustrated in Figure 5, Joint 3 showed increasing rise and settling times with heavier payloads (4,3 s → 4,5 s and 11,0 s → 12,55 s). Again, overshoot was fully eliminated, and steady-state errors were maintained below 10^{-3}. The results confirm that the PID controller effectively balanced stability and accuracy, even when the joint operated against gravitational torque.

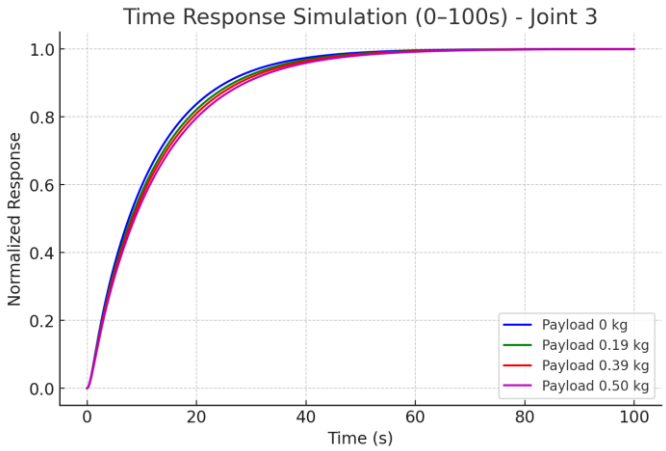

Figure 5. Time Response for Joint-3

4) Joint 4

Distinct from earlier joints, Joint 4's movement was aligned with gravity during downward motion, as depicted in Figure 6. Consequently, heavier payloads actually reduced rise and settling times (2,48 s and 2,94 s without load → 1,83 s and 2,08 s with 0,50 kg). No overshoot was recorded, and steady-state errors were negligible. This phenomenon highlights the beneficial effect of gravitational assistance in reducing dynamic response times.

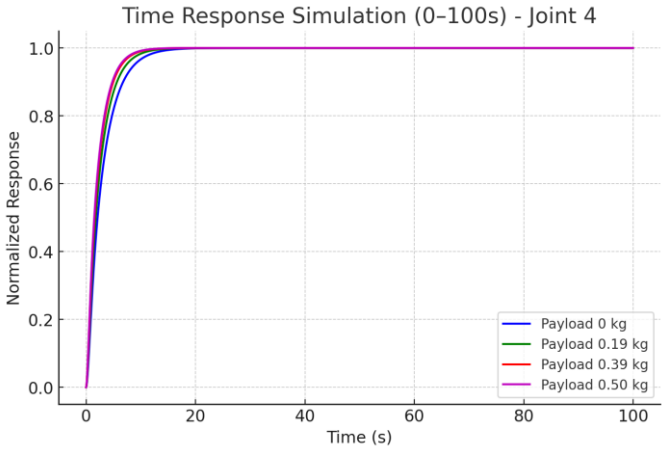

Figure 6. Time Response for Joint-4

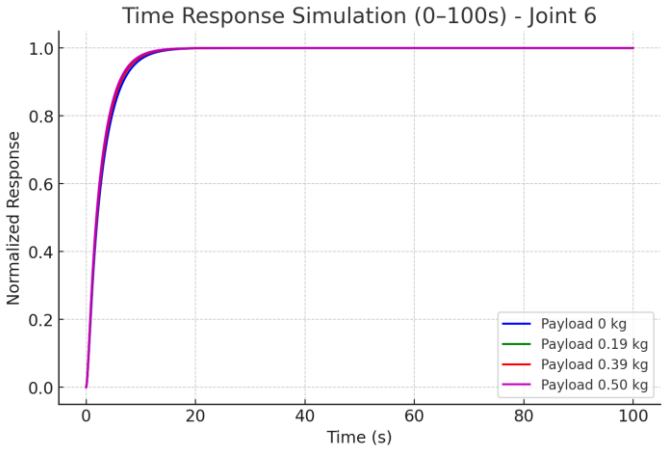

Figure 8. Time Response for Joint-6

5) Joint 5

As presented in Figure 7, Joint 5 followed a 90° counter-clockwise trajectory. Heavier payloads led to modest increases in rise and settling times (1,385 s → 1,41 s and 1,95 s → 2,17 s). The PID controller again prevented overshoot, while steady-state errors were minimal (10^{-4} order). These results confirm that precision was maintained even under load, making the joint suitable for delicate manipulation.

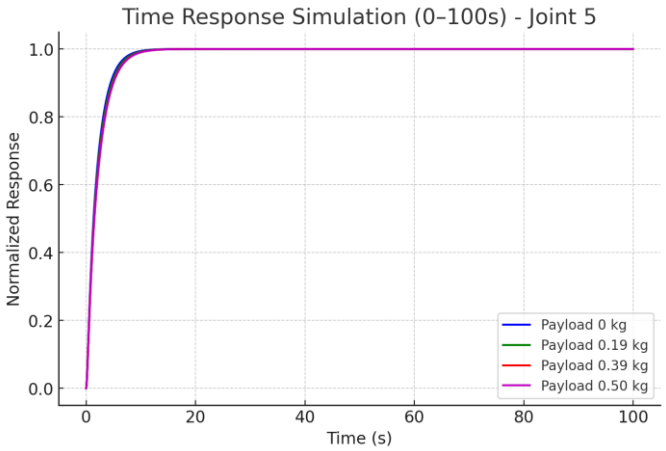

Figure 7. Time Response for Joint-5

6) Joint 6

Like Joint 4, Joint 6's downward motion aligned with gravity, resulting in decreased rise and settling times with heavier payloads (2,29 s → 2,05 s and 2,94 s → 2,63 s), as illustrated in Figure 8. No overshoot occurred, and steady-state errors remained very small (10^{-4} order). This indicates that gravitational support can significantly improve controller efficiency in end-effector joints

C. Sensitivity Analysis to Modeling Uncertainties

To evaluate the robustness of the tuned controllers, the system was tested under ±10% variations in both link mass and friction coefficients. Increasing mass and friction by 10% led to moderate increases in rise and settling times (approximately 5–15%) for joints operating against gravity (Joints 1–3 and 5), while gravity-assisted joints (Joints 4 and 6) showed negligible degradation. Conversely, decreasing these parameters accelerated responses slightly but did not induce overshoot or instability. These results indicate that the tuned PID gains provide robust stability margins even with moderate modelling inaccuracies, validating the reliability of the control architecture under real-world parameter deviations.

D. Continuous Trajectory Tracking

To evaluate the controller's capability in executing smooth and realistic manipulator motions, a continuous trajectory tracking test was conducted using a minimum-jerk profile. This trajectory type ensures smooth acceleration and deceleration phases, emulating a natural pick-and-place movement frequently encountered in industrial operations. The desired path was generated for Joint 2, commanding it to follow a continuous angular motion between 0° and 45° over a 30-second interval.

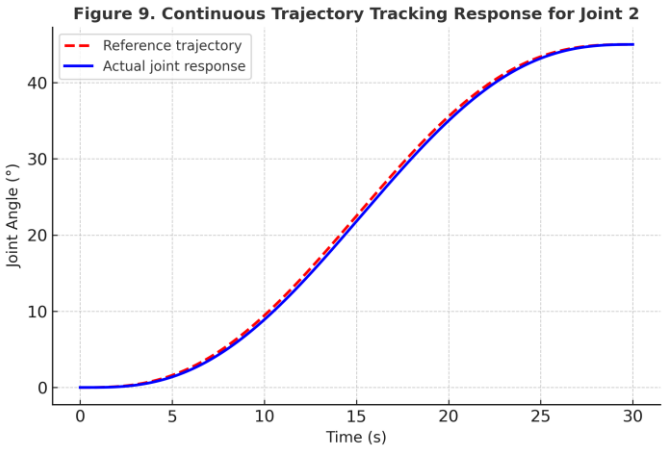

Figure 9. Continuous trajectory tracking response for Joint 2

The results, illustrated in Figure 9, show both the reference trajectory and the corresponding actual joint response. The PID controller successfully tracked the desired path with minimal delay and high precision. The maximum steady-state tracking error was approximately 0,4°, while the peak lag time did not exceed 0,25 s during acceleration phases. The overall motion was free of oscillations and exhibited stable damping behavior across the entire trajectory.

These findings confirm that the tuned PID controller provides effective continuous path tracking performance, ensuring smooth and accurate motion suitable for real-world manipulator operations such as welding, assembly, and precision handling. The graphical response in Figure 10 demonstrates the close alignment between the reference and actual trajectories, highlighting the controller's robustness and responsiveness during dynamic, continuous tasks.

E. Discussion of Overall Performance

Across all six joints, the PID controllers demonstrated highly consistent and reliable performance characteristics, as summarized in Table 3. The absence of overshoot under all test conditions indicates exceptional robustness and precise damping, ensuring stable motion even under dynamically changing payloads. Such behaviour is particularly valuable in industrial contexts, such as precision assembly or material handling, where overshoot can result in mechanical stress, positioning errors, or object slippage.

Furthermore, the controllers maintained steady-state errors approaching zero, typically within the range of 10^{-6} to 10^{-3} across all joints and payloads. This level of accuracy validates the effectiveness of the PID tuning process performed through ROS Dynamic Reconfigure and confirms that the simulation framework accurately represents real-world actuator behaviour. The precise steady-state convergence also demonstrates the high resolution and responsiveness of the simulated control loop within the ROS–Gazebo environment.

TABLE III
OVERALL PERFORMANCE TREND

Joint	Trend in Rise Time	Trend in Settling Time	Overshoot	Steady-State Error
1	↑ with payload	↑ with payload	None	Near zero (10^{-6})
2	↑ with payload	↑ with payload	None	Near zero (10^{-5})
3	↑ with payload	↑ with payload	None	Near zero (10^{-3})
4	↓ with payload (gravity assist)	↓ with payload (gravity assist)	None	Near zero (10^{-6})
5	↑ with payload	↑ with payload	None	Near zero (10^{-4})
6	↓ with payload (gravity assist)	↓ with payload (gravity assist)	None	Near zero (10^{-4})

Dynamic trends revealed that rise and settling times varied systematically with payload conditions. For joints operating against gravity (Joints 1–3 and 5), heavier payloads caused a slight increase in rise and settling times due to higher inertia and torque demand. In contrast, joints that moved with gravitational assistance (Joints 4 and 6) exhibited reduced rise and settling times as additional load effectively enhanced torque output, resulting in faster movement. This inverse relationship highlights how gravitational forces can either hinder or assist joint motion depending on orientation, a factor that can be exploited in controller gain optimization.

In addition, the controllers exhibited consistent performance despite the introduction of ±10% variations in mass and friction parameters, confirming robustness against modelling uncertainties. The combination of stability, precision, and resilience indicates that the implemented PID structure, even in its classical form, remains a dependable solution for multi-joint robotic control when properly tuned.

Overall, these results validate the proposed ROS–Gazebo-based PID control framework as a reliable, low-risk, and accurate simulation environment for industrial robotic systems. The consistent convergence behavior, absence of oscillations, and predictable payload-dependent trends demonstrate the framework's potential for design verification, parameter optimization, and controller prototyping prior to physical implementation. This establishes a foundation for integrating more advanced techniques, such as adaptive, hybrid, or AI-driven control strategies into future research and industrial practice.

IV. CONCLUSION

This study investigated joint-level PID control for a 6-DOF robotic manipulator within a ROS–Gazebo environment and demonstrated that systematic PID tuning enables stable, precise, and adaptable performance under varying payloads. The optimized PID parameters showed excellent dynamic stability across all joints, with no overshoot and consistently near-zero steady-state errors (10^{-6}–10^{-3}), confirming effective damping and high positional accuracy. Payload changes produced predictable effects on rise and settling times, increasing for joints operating against gravity and decreasing for gravity-assisted joints, further supporting the physical realism of the simulation. These results validate ROS–Gazebo as a reliable and replicable platform for safe, low-risk controller design and optimization, offering an efficient balance of simplicity, stability, and precision for industrial applications. Future work will involve validating the control scheme on a physical manipulator, exploring advanced methods such as adaptive or robust control, and integrating AI-driven gain adaptation to enhance autonomy and performance in more complex and uncertain environments.

REFERENCES

[1] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed. Boston, MA, USA: Pearson, 2018.

[2] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[3] M. Quigley et al., "ROS: an open-source Robot Operating System," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, 2009.

[4] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 2149–2154.

[5] B. Megalingam, S. V. Krishnan, and R. Ramesh, "Design and Simulation of a 6 DOF Robotic Arm using ROS and Gazebo," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 4, pp. 25–32, 2020.

[6] Y. Zhang, X. Liu, and H. Li, "Controller performance evaluation for a 7-DOF robotic arm using ROS and Gazebo," *J. Intell. Robot. Syst.*, vol. 101, no. 3, pp. 1–15, 2021.

[7] G. Mengacci, G. Palli, and C. Melchiorri, "An open-source ROS-Gazebo toolbox for articulated and compliant robots," *Front. Robot. AI*, vol. 8, pp. 1–12, 2021.

[8] S. Amiri, H. Soleimani, and M. Rahmani, "Trajectory tracking of robotic manipulators using PSO-tuned PID controllers in ROS–Gazebo," *Sensors*, vol. 21, no. 17, pp. 1–15, 2021.

[9] Y. Chen, J. Wu, and Z. Liu, "A refined PID with fuzzy adaptation for 6-DOF robotic manipulators in ROS–Gazebo environment," *Sensors*, vol. 22, no. 4, pp. 1200–1215, 2022.

[10] H. Alkan et al., "Design and simulation of a multi-DOF robotic arm using ROS–Gazebo," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 2, pp. 45–53, 2024.

[11] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, Springer, 2000.

[12] V. Tinoco *et al*., "A Review of Advanced Controller Methodologies for Robotic Manipulators," *Int. J. Dyn. Control*, vol. 13, no. 1, Jan. 2025.

[13] M. W. Spong, S. Hutchitson, and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, 1989.

[14] Y. Kim and J. Kim, "A ROS-based Real-Time Control System for a Multi-DOF Robotic Arm with Gazebo Simulation," in *2022 IEEE Int. Conf. Robot. Automat. (ICRA)*, pp. 2534–2540.

[15] U. Kabir *et al*., "Performance Analysis of PID, PD and Fuzzy Controllers for Position Control of 3-Dof Robot Manipulator," *arXiv preprint arXiv:1910.12076*, Oct. 2019.

[16] R. Mengacci *et al*., "An Open-Source ROS-Gazebo Toolbox for Simulating Robots with Compliant Actuators," *Front. Robot. AI*, vol. 8, p. 713083, Aug. 2021.

[17] K. H. Ang, G. C. Y. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[18] Joseph, Stephen & Dada, Emmanuel & Abidemi, Afeez & Oyewola, David & Mohammed, Ban. (2022). *Metaheuristic Algorithms for PID Controller Parameters Tuning: Review, Approaches and Open Problems*. Heliyon. 8. e09399. 10.1016/j.heliyon. 2022.e09399.