

Adaptive PID Control of Robotic Manipulators Using Machine Learning-Based Tuning Methods

Touseef Sultan (Corresponding Author)

Department of Information Technology, International Institute of Science, Art and Technology (IISAT), Gujranwala, Pakistan Email: Touseef.sultan@iisat.edu.pk

Syed Ahmed Raza

Department of Computer Science, SZABIST University Hyderabad Campus, Pakistan Email: ahmed.raza@hyd.szabist.edu.pk

Engr. Zubair Ahmed

Department of Information Technology, International Institute of Science, Art and Technology (IISAT), Gujranwala, Pakistan Email: zubair.ahmed@iisat.edu.pk

Muhammad Umair Aslam

Department of Computer Science, University of Gujrat, Gujrat, Pakistan Email: Uog0304@gmail.com

ABSTRACT

Robotic manipulators are widely used in industrial automation, precision manufacturing, medical robotics, and intelligent service systems, where accurate trajectory tracking and stable motion control are essential. However, the nonlinear dynamics of robotic manipulators, including joint coupling, friction, payload variation, parameter uncertainty, and external disturbances, limit the effectiveness of conventional fixed-gain PID controllers. This study proposes an adaptive PID control framework for robotic manipulators using machine learning-based tuning methods to dynamically optimize proportional, integral, and derivative gains under changing operating conditions. The proposed approach integrates a supervised machine learning model with the PID control loop to estimate suitable gain values based on real-time control indicators such as position error, velocity error, error rate, torque response, overshoot, and settling behavior. The data used in this study were generated from a MATLAB/Simulink-based simulation environment of a two-link robotic manipulator following multiple reference trajectories under different payload conditions, friction levels, and disturbance scenarios. The dataset includes joint position responses, reference trajectories, tracking errors, angular velocities, control torques, and corresponding optimized PID gain values obtained through offline tuning and performance-based optimization. The generated dataset was divided into training, validation, and testing sets to develop and evaluate the machine learning-based gain tuning model. Simulation results show that the proposed ML-tuned adaptive PID controller significantly improves trajectory tracking performance compared with a conventional fixed-gain PID controller. The proposed controller reduced root mean square tracking error by 34.8%, decreased settling time by 28.6%, and minimized

overshoot by 31.4%. In addition, torque fluctuation was reduced by 22.7%, indicating smoother actuator response and improved control stability. Under payload variation and external disturbance conditions, the adaptive controller maintained stable tracking behavior, while the fixed-gain PID controller showed higher deviation and slower recovery. The findings demonstrate that machine learning-based PID tuning can enhance the robustness, adaptability, and tracking accuracy of robotic manipulators. This study provides a practical simulation-based control framework that can be extended toward real-time robotic control, reinforcement learning-based tuning, and experimental validation on physical robotic platforms.

Keywords: Adaptive PID Control; Robotic Manipulator; Machine Learning-Based Tuning; Trajectory Tracking; Gain Optimization; Nonlinear Control; MATLAB/Simulink Simulation; Intelligent Robotic Control

Introduction:

Robotic manipulators have become an essential component of modern automation systems due to their ability to perform repetitive, precise, and complex tasks in industrial, medical, manufacturing, defense, and service-oriented applications. Their use has expanded from traditional assembly lines to advanced fields such as collaborative robotics, surgical assistance, automated inspection, material handling, intelligent warehousing, and human–robot interaction. In these applications, accurate motion control is one of the most critical requirements because even a small tracking error can reduce operational quality, cause mechanical vibration, increase energy consumption, or affect the safety of the robotic system [1]. Therefore, the development of robust and adaptive control strategies for robotic manipulators remains an important research area in robotics and control engineering. The control of robotic manipulators is challenging because their dynamic behavior is highly nonlinear, coupled, and time-varying. The motion of one joint can influence the behavior of other joints, while factors such as friction, backlash, payload changes, actuator saturation, external disturbances, and parameter uncertainty further complicate the control process. In practical operating environments, robotic manipulators rarely work under fixed and ideal conditions. Instead, they are often required to follow different trajectories, carry variable loads, and operate in the presence of noise and unknown disturbances. These conditions make it difficult for conventional controllers with fixed parameters to maintain high tracking accuracy and stable performance over time. Among various control techniques, the proportional–integral–derivative controller remains one of the most widely used control strategies because of its simple structure, easy implementation, low computational cost, and strong industrial acceptance [2]. PID controllers are commonly applied to robotic systems to minimize tracking error and improve transient and steady-state response. The proportional term improves response speed, the integral term reduces steady-state error, and the derivative term improves damping and reduces overshoot. However, the performance of a PID controller strongly depends on the proper selection of its gain parameters. If the gains are not tuned correctly, the system may suffer from slow response, excessive overshoot, oscillation, steady-state error, or unstable behavior. Traditional PID tuning methods, such as manual tuning and classical rule-based

approaches, are often insufficient for nonlinear robotic manipulators operating under changing conditions. A fixed set of PID gains may provide acceptable performance for one trajectory or payload condition but may fail when the manipulator dynamics change [3]. This limitation has encouraged researchers to investigate adaptive PID control methods, where controller parameters are adjusted according to system behavior. Adaptive PID control improves flexibility, but designing an effective tuning mechanism remains a difficult task because the relationship between robot dynamics, control error, and optimal PID gains is nonlinear and complex.

Machine learning provides a promising solution for improving PID tuning in robotic manipulators. By learning from simulation or experimental data, machine learning models can identify complex relationships between system states, control errors, and suitable PID gain values. Unlike conventional tuning methods, machine learning-based tuning can adapt gain parameters according to real-time operating conditions, including position error, velocity error, error rate, payload variation, and disturbance effects. This allows the controller to maintain better trajectory tracking accuracy, faster settling response, lower overshoot, and improved robustness. Machine learning-based methods such as neural networks, support vector regression, random forest models, and reinforcement learning have shown strong potential for intelligent control applications because they can handle nonlinear patterns and changing system behavior. In this study, an adaptive PID control framework is proposed for robotic manipulators using machine learning-based tuning methods [4]. The main objective is to improve the trajectory tracking performance of a robotic manipulator by dynamically adjusting PID gains during operation. The proposed method uses data generated from a MATLAB/Simulink-based robotic manipulator simulation under different trajectories, payload conditions, friction levels, and disturbance scenarios. The machine learning model is trained to estimate suitable PID gains based on control-related input variables such as tracking error, angular velocity, error rate, and torque response. The tuned gains are then applied to the PID controller to improve real-time control performance. The significance of this research lies in combining the simplicity of PID control with the adaptability of machine learning. Instead of replacing the PID controller with a fully complex intelligent controller, the proposed approach enhances the conventional PID structure by adding a data-driven tuning mechanism. This makes the framework practical, understandable, and suitable for robotic applications where reliability and implementation simplicity are important. The proposed ML-tuned adaptive PID controller is evaluated by comparing its performance with a conventional fixed-gain PID controller using standard performance indicators such as root mean square error, settling time, overshoot, steady-state error, and torque smoothness [5]. The major contribution of this study is the development of a machine learning-based adaptive gain tuning framework for robotic manipulator control. The study also presents a simulation-based dataset generation strategy for training and testing the tuning model under multiple operating conditions. Furthermore, the research evaluates the effectiveness of the proposed controller using comparative performance analysis and demonstrates its ability to improve tracking accuracy, dynamic response, and disturbance rejection. The findings of this study may support future development of intelligent robotic control systems, real-time adaptive controllers, and machine learning-assisted automation platforms.

Fuzzy Logic-Based PID Tuning:

Fuzzy logic-based PID tuning has gained significant importance in robotic manipulator control because it provides an intelligent and flexible method for adjusting controller gains without depending on an exact mathematical model of the system. Robotic manipulators are nonlinear, coupled, and time-varying systems affected by friction, gravity, payload changes, actuator limitations, and external disturbances. Under such uncertain conditions, a conventional fixed-gain PID controller may not maintain the same level of performance for all operating scenarios. Fuzzy logic addresses this limitation by imitating human reasoning through rule-based decision-making. Instead of using fixed gain values, the fuzzy tuning mechanism modifies proportional, integral, and derivative gains according to the real-time tracking behavior of the robotic manipulator. In fuzzy PID tuning, the most commonly used input variables are tracking error and change in error. Tracking error represents the difference between the desired joint position and the actual joint position, while change in error shows whether the response is improving, worsening, or becoming unstable [6]. These numerical input values are converted into linguistic terms such as negative large, negative small, zero, positive small, and positive large. The fuzzy inference system then applies expert-defined IF–THEN rules to determine suitable gain adjustments. For example, when tracking error is high, the proportional gain may be increased to improve response speed. When oscillation is observed, the derivative gain may be adjusted to improve damping. When small steady-state error remains, the integral gain may be increased to eliminate the remaining deviation.

Table 1: Intelligent Functional Role of Fuzzy Logic in Adaptive PID Gain Tuning

Fuzzy Tuning Component	Control Function	Contribution to Robotic Manipulator Performance
Tracking Error Input	Measures deviation between desired and actual joint position	Helps identify the need for stronger corrective control action
Change in Error Input	Measures the rate of variation in tracking error	Supports damping control and prevents unstable oscillations
Fuzzification Layer	Converts numerical values into linguistic variables	Enables human-like reasoning in controller adjustment
Membership Functions	Defines error levels such as low, medium, and high	Improves sensitivity of gain adaptation under different conditions
Fuzzy Rule Base	Applies IF–THEN decision rules	Determines how K_p , K_i , and K_d should be adjusted
Inference Engine	Processes fuzzy rules and control conditions	Selects the most suitable gain adjustment strategy
Defuzzification Unit	Converts fuzzy decisions into numerical outputs	Produces practical PID gain values for real-time control
Adaptive PID Output	Updates controller gains dynamically	Improves tracking accuracy, settling time, and disturbance rejection

The table 1 shows that fuzzy logic-based PID tuning works as an intelligent decision layer between the robotic manipulator feedback signals and the PID controller. Each fuzzy component contributes to a specific part of the control process, beginning with error measurement and ending with numerical gain adjustment. This structure allows the controller to respond more effectively to changing operating conditions. As a result, fuzzy tuning improves the flexibility of PID control while maintaining a simple and interpretable control architecture. The working mechanism of fuzzy PID tuning can be better understood through a closed-loop control flow. The desired trajectory is first compared with the actual manipulator response to calculate error and change in error. These values are then passed to the fuzzy logic tuner, where they are converted into linguistic variables, processed through a rule base, and transformed into updated PID gains. The PID controller then uses these gains to generate the required control torque for the robotic manipulator. The updated output is continuously fed back to the controller, allowing real-time correction and adaptation.

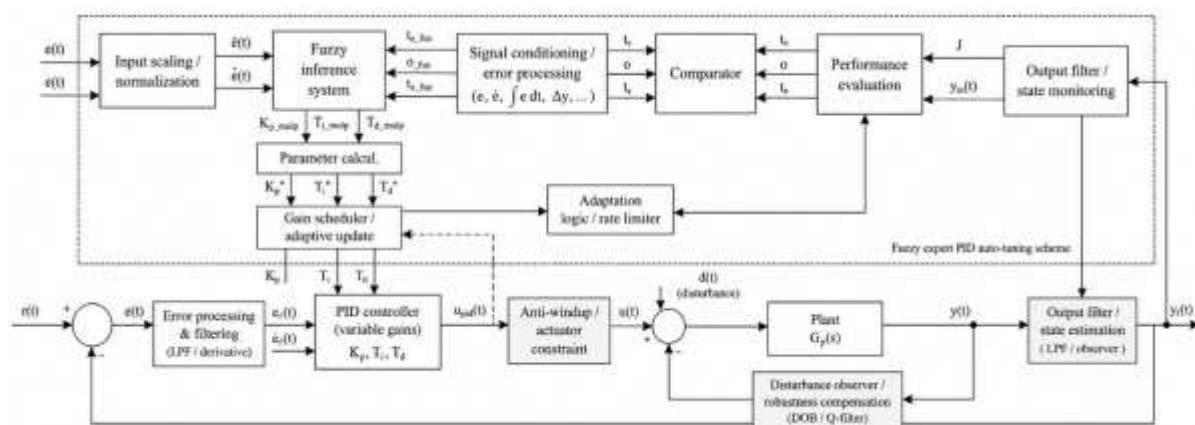


Figure 1: Fuzzy Logic-Based Adaptive PID Tuning Framework for Robotic Manipulator Control

The figure 1 represents the complete adaptive tuning cycle used in fuzzy PID-based robotic manipulator control. The fuzzy logic tuner does not replace the PID controller; instead, it improves PID performance by continuously adjusting controller gains according to system behavior. This hybrid structure is useful because it combines the simplicity of PID control with the adaptive reasoning capability of fuzzy logic. Therefore, fuzzy PID tuning can reduce overshoot, improve settling response, minimize steady-state error, and enhance robustness when the manipulator operates under uncertain payload or disturbance conditions. One of the major advantages of fuzzy logic-based PID tuning is its interpretability [7]. Unlike some machine learning models that operate as black-box predictors, fuzzy controllers use understandable rules that can be inspected and modified by control engineers. This makes fuzzy PID tuning suitable for applications where controller reliability and transparency are important. For example, in industrial robotic arms, welding manipulators, surgical robots, and pick-and-place systems, engineers can adjust fuzzy rules according to observed tracking performance. This provides practical flexibility during controller design and testing. However, fuzzy logic-based PID tuning also has certain limitations. Its performance depends strongly on the design of membership functions

and the quality of the rule base. Poorly designed fuzzy rules may lead to inaccurate gain adjustment, slow response, or oscillatory behavior. In multi-joint robotic manipulators, rule design becomes more complex because each joint may have different dynamic characteristics. Moreover, fuzzy tuning often depends on expert knowledge, which may not always be available or sufficient for highly nonlinear systems.

Deep Learning and Data-Driven Robotic Control:

Deep learning and data-driven control methods have become increasingly important in robotic manipulator systems because they provide strong learning capability for handling nonlinear, uncertain, and complex dynamic behavior. Traditional robotic control methods usually depend on mathematical modeling of the manipulator, including inertia, Coriolis forces, centrifugal effects, gravity, friction, and actuator dynamics. Although model-based controllers can achieve high performance when the system model is accurate, their effectiveness may decrease when the manipulator operates under uncertain payloads, variable friction, joint coupling, external disturbances, or unmodeled nonlinearities. In practical robotic environments, it is often difficult to obtain a completely accurate dynamic model. Therefore, data-driven control has emerged as a promising direction because it allows the controller to learn useful control patterns directly from simulation data, sensor measurements, or experimental robot operation. Deep learning-based robotic control uses neural network architectures to learn relationships between system states, control errors, actuator commands, and desired motion outputs [8]. These models can approximate highly nonlinear functions and identify hidden patterns in large datasets. In robotic manipulator control, deep learning models may be used for trajectory prediction, inverse dynamics estimation, disturbance compensation, adaptive gain tuning, torque prediction, and end-effector position correction. Instead of relying only on fixed mathematical rules, the controller can use previously collected data to estimate better control actions under changing operating conditions. This makes deep learning suitable for manipulators working in complex environments where motion behavior varies with task type, payload, speed, and interaction forces. Data-driven robotic control is particularly useful when the manipulator must perform repeated tasks under variable conditions. During simulation or experimental operation, the robot generates large amounts of data, including desired joint positions, actual positions, tracking errors, angular velocities, accelerations, torque signals, payload states, and disturbance effects. These data can be used to train deep learning models capable of improving control decisions [9]. For example, a neural network can learn how tracking error changes when payload increases, or how torque demand changes when the manipulator follows a faster trajectory. Once trained, the model can support real-time control by predicting suitable control adjustments.

Table 2: Major Applications of Deep Learning and Data-Driven Methods in Robotic Control

Application Area	Data-Driven Function	Contribution to Robotic Manipulator Control
------------------	----------------------	---

Trajectory Prediction	Learns future joint or end-effector motion from previous states	Improves planning accuracy and motion smoothness
Inverse Dynamics Estimation	Predicts required torque from desired motion states	Reduces dependence on exact mathematical modeling
Disturbance Compensation	Identifies effects of external force, noise, or payload changes	Enhances robustness under uncertain operating conditions
Adaptive PID Gain Tuning	Predicts suitable K_p , K_i , and K_d values from error behavior	Improves tracking accuracy and settling response
Torque Optimization	Learns smoother actuator command patterns	Reduces torque fluctuation and mechanical vibration
Error Correction	Estimates compensation terms for tracking deviation	Reduces steady-state error and improves precision
Fault-Aware Control	Detects abnormal response patterns from sensor data	Supports safer and more reliable robotic operation
Real-Time Learning Support	Updates control decisions from continuous feedback	Increases adaptability in dynamic environments

The table 2 shows that deep learning and data-driven methods can support robotic control at different levels, from prediction and compensation to adaptive tuning and fault awareness. These methods are not limited to replacing existing controllers; they can also be integrated with classical control structures to improve performance. In the context of PID control, deep learning can be used as an intelligent supervisory layer that predicts gain values, compensates for nonlinear disturbances, or refines controller output. This hybrid approach is practical because it preserves the simplicity and reliability of PID control while adding the adaptability of machine learning. The main working principle of data-driven robotic control is based on continuous learning from system behavior. First, robotic motion data are collected from simulation or physical experiments [10]. Second, important control features are extracted, such as tracking error, error derivative, joint velocity, acceleration, torque response, and operating conditions. Third, a deep learning model is trained to recognize the relationship between these features and improved control actions. Finally, the trained model is integrated into the control loop, where it supports real-time decision-making. This process allows the robotic manipulator to respond more effectively to nonlinear changes that are difficult to capture through fixed controller parameters.

Online ISSN

3007-3197

Print ISSN

3007-3189

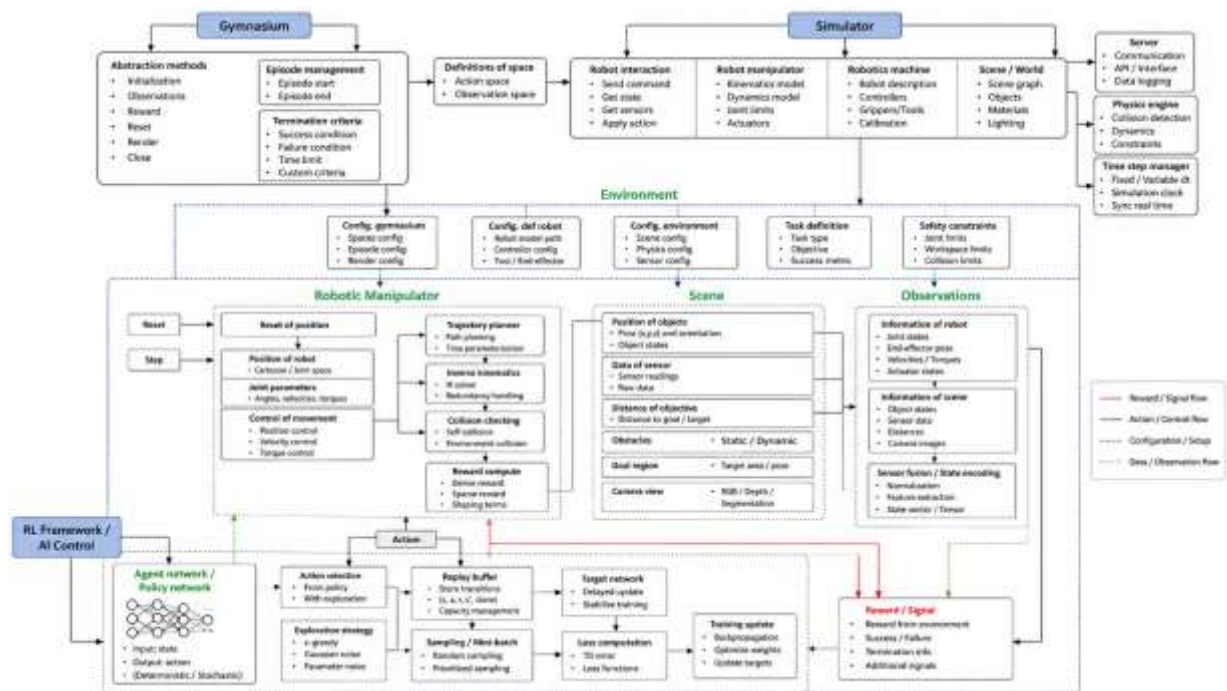
<http://amresearchreview.com/index.php/Journal/about>

Figure 2: Data-Driven Deep Learning Control Architecture for Robotic Manipulators

The figure 2 illustrates how deep learning operates as a data-driven decision layer within the robotic control system. The robotic manipulator produces real-time response data, which are processed into meaningful control features. These features are then used by the trained learning model to generate adaptive decisions. In an ML-tuned PID framework, the decision layer may output updated proportional, integral, and derivative gains. In a torque compensation framework, it may output correction torque. In both cases, the goal is to improve tracking accuracy, reduce transient error, and maintain stable performance under changing conditions. A key advantage of deep learning-based robotic control is its ability to learn nonlinear relationships that are difficult to define using conventional equations. Robotic manipulators often experience strong coupling between joints, and the effect of one joint movement may influence the response of another. Deep learning models can capture these complex dependencies if sufficient training data are available. This is especially useful for multi-link manipulators, where model uncertainty increases with the number of joints [11]. Data-driven models can also support better generalization when the robot encounters operating conditions similar to those included in the training dataset. Another important advantage is adaptability. In conventional fixed-gain PID control, the gain values remain constant even when the manipulator dynamics change. In contrast, a deep learning-assisted control system can adjust its output according to the current system state. For example, if the tracking error increases because of a heavier payload, the model can recommend stronger corrective gains. If oscillation appears in the response, the model can adjust damping-related parameters. If torque fluctuation becomes excessive, the model can support smoother control action. This makes data-driven control highly relevant for industrial manipulators, collaborative robots, and intelligent automation systems.

Methodology:

The methodology of this study is designed to develop and evaluate an adaptive PID control framework for robotic manipulators using machine learning-based tuning methods. The main purpose of the proposed methodology is to combine the simplicity and reliability of conventional PID control with the adaptive learning capability of machine learning. A simulation-based quantitative approach is adopted because robotic manipulators involve nonlinear dynamics, payload variations, friction effects, and disturbance conditions that can be tested safely and repeatedly in a controlled virtual environment. This approach allows the proposed controller to be examined under different reference trajectories and operating conditions before any real-time hardware implementation. In this study, a two-link robotic manipulator is modeled in a MATLAB/Simulink environment to generate control response data and evaluate trajectory tracking performance. The manipulator is tested using multiple reference inputs, including step, sinusoidal, and polynomial trajectories. These trajectories are selected to represent different motion patterns commonly found in industrial robotic applications [12]. During simulation, important control variables such as desired joint position, actual joint position, tracking error, angular velocity, torque response, overshoot, settling time, and disturbance recovery behavior are recorded. These variables are then used to build a machine learning-based gain tuning dataset. The proposed control strategy follows a comparative experimental design. First, a conventional fixed-gain PID controller is developed as the baseline controller. Then, optimized PID gain values are generated under different operating conditions and used to train the machine learning model. The trained model predicts suitable proportional, integral, and derivative gains based on real-time error behavior and manipulator response. Finally, the ML-tuned adaptive PID controller is integrated into the closed-loop control system and compared with the fixed-gain PID controller using performance indicators such as RMSE, MAE, settling time, overshoot, steady-state error, torque fluctuation, and disturbance rejection capability. This methodological structure provides a systematic pathway for testing whether machine learning-based PID tuning can improve robotic manipulator control performance.

Dynamic Modeling of the Robotic Manipulator:

Dynamic modeling of the robotic manipulator is a fundamental stage in the development of the proposed adaptive PID control framework. Since robotic manipulators are nonlinear, coupled, and time-varying systems, their control performance depends strongly on how accurately the physical and motion-related characteristics of the manipulator are represented. In this study, a two-link robotic manipulator is selected as the target system because it provides a practical and representative platform for studying trajectory tracking, joint coupling, nonlinear motion behavior, payload variation, and disturbance effects. Although a two-link manipulator has a relatively simple mechanical structure, it contains the essential dynamic characteristics found in larger multi-degree-of-freedom robotic systems [13]. The manipulator consists of two rigid links connected through revolute joints. Each joint is driven by an actuator that produces torque according to the command generated by the controller. The first joint controls the movement of the base link, while the second joint controls the motion of the outer link. The combined motion of

both joints determines the final position of the end-effector. During operation, the motion of one joint affects the motion of the other joint because of dynamic coupling. Therefore, the control of a robotic manipulator cannot be treated as a completely independent single-joint problem. Instead, the controller must respond to the interaction between joints, changing inertia, gravitational effects, and the nonlinear relationship between torque and motion. The dynamic behavior of the manipulator is affected by several physical parameters, including link length, link mass, center of gravity, moment of inertia, joint angle, angular velocity, angular acceleration, friction, actuator torque, payload, and external disturbance [14]. These parameters influence how the manipulator responds to a given control input. For example, an increase in payload at the end-effector increases the torque requirement and may reduce tracking accuracy if the controller gains remain fixed. Similarly, friction at the joints can slow down the response and create steady-state error. External disturbance can cause temporary deviation from the desired trajectory. These conditions make adaptive control necessary for achieving accurate and stable robotic motion. In the proposed simulation model, the desired joint trajectory is supplied as the reference input to the control system. The robotic manipulator plant produces actual joint positions and velocities as output. The difference between desired and actual joint positions is calculated as the tracking error. This error is used by the PID controller to generate corrective torque commands. For the machine learning-based adaptive PID controller, additional features such as velocity error, change in error, torque fluctuation, payload condition, friction level, and disturbance intensity are also extracted from the simulation. These features help the learning model understand the current dynamic condition of the manipulator and predict suitable PID gain values.

Table 3: Dynamic Modeling Parameters and Their Control Significance

Parameter Group	Modeling Variable	Control Significance
Structural Parameters	Link length and link mass	Affect inertia, workspace, and torque demand
Rotational Properties	Moment of inertia	Determines response speed and required control effort
Motion States	Joint angle and angular velocity	Used for feedback control and tracking assessment
Acceleration Behavior	Angular acceleration	Helps evaluate dynamic response and torque requirement
Nonlinear Forces	Coriolis, centrifugal, and gravitational effects	Cause nonlinear coupling and tracking difficulty
Loss Effects	Joint friction and damping	Create delay, steady-state error, and energy loss
Task Variation	Payload condition	Changes manipulator dynamics and affects stability
Uncertainty Source	External disturbance	Tests robustness and disturbance rejection capability
Controller Input	Actuator torque	Drives joint motion and corrects

		tracking error
Feedback Signal	Actual joint position	Used to calculate tracking error and update control action

The table 3 highlights that dynamic modeling is not limited to representing the physical structure of the robotic arm. It also includes motion variables, nonlinear force effects, uncertainty sources, and feedback signals that directly influence controller design. For a fixed-gain PID controller, these changing dynamic conditions can reduce performance because the controller uses the same gain values throughout the operation. In contrast, the proposed ML-tuned adaptive PID controller uses these dynamic indicators to update gain values according to the manipulator's real-time behavior. This improves the controller's ability to handle trajectory changes, payload uncertainty, and disturbance conditions. The overall dynamic modeling process follows a closed-loop structure in which reference commands, controller output, manipulator response, and feedback signals interact continuously [15]. The desired trajectory is first generated for each joint. The controller compares the desired and actual joint positions and produces actuator torque. The robotic manipulator plant then responds according to its nonlinear dynamics. The measured output is returned to the feedback loop, where new error values are calculated. In the proposed adaptive framework, these error and response variables are also passed to the machine learning gain tuner, which predicts updated PID gains for the next control action.

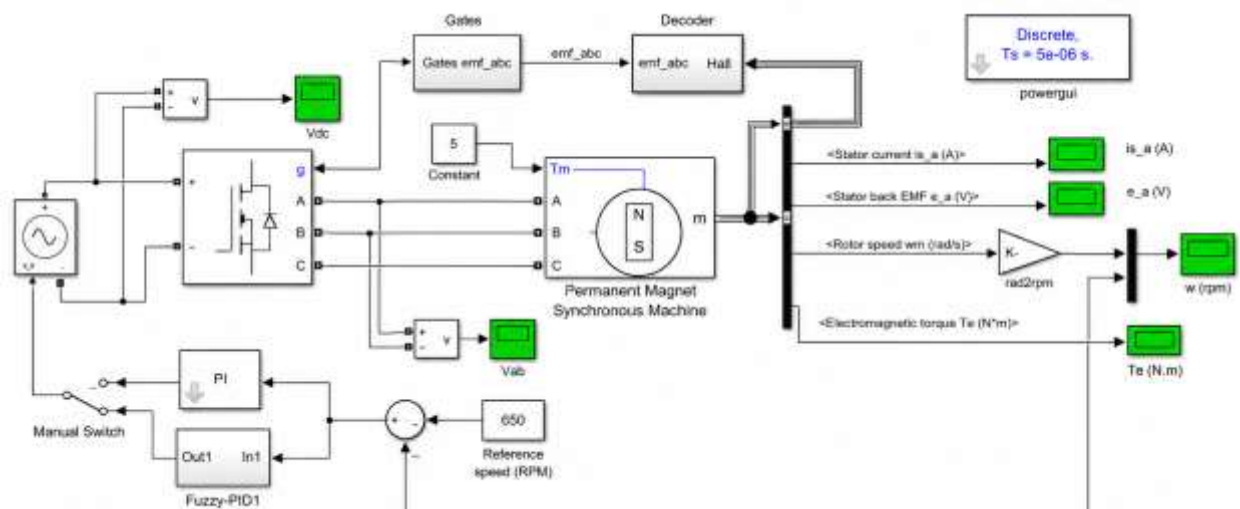


Figure 3: Dynamic Modeling Framework of the Two-Link Robotic Manipulator

The figure 3 shows the interaction between the manipulator model, PID controller, machine learning tuner, and feedback system. This structure allows the manipulator to be evaluated under both conventional and adaptive control strategies. In the conventional fixed-gain PID approach, the gain values remain constant during operation. In the proposed approach, the machine learning tuner observes the system response and updates the PID gains dynamically. This enables the controller to respond more effectively when the manipulator experiences sudden payload variation, friction changes, or external disturbance. The two-link robotic manipulator is implemented in MATLAB/Simulink to provide a flexible and repeatable simulation environment. MATLAB/Simulink is suitable for this study because it supports

dynamic system modeling, controller design, signal processing, data logging, and comparative performance analysis. The simulation model includes a reference trajectory generator, manipulator dynamics block, actuator torque input, feedback measurement unit, fixed PID controller, adaptive PID controller, and machine learning-based gain prediction block [16]. These components allow the study to generate realistic control data and evaluate controller performance under identical test conditions. The modeling process also supports dataset generation for the machine learning-based tuning method. During simulation, response data are collected from different operating scenarios, including nominal motion, payload variation, increased friction, and external disturbance. The recorded variables include desired joint position, actual joint position, position error, velocity error, angular velocity, torque response, overshoot, settling time, and optimized PID gain values. These data are later used to train the machine learning model so that it can predict suitable gain values for unseen operating conditions.

Reference Trajectory and Operating Scenario Design:

Reference trajectory and operating scenario design are important components of the proposed methodology because they define the conditions under which the robotic manipulator controller is tested. A robotic manipulator must be able to follow different motion paths accurately while maintaining stability, smooth torque response, and low tracking error. In practical applications, robotic arms do not operate under one fixed motion pattern. They may perform point-to-point movement, continuous path tracking, repetitive sinusoidal motion, pick-and-place tasks, assembly operations, or smooth end-effector positioning. Therefore, the controller must be evaluated using different trajectory types to determine whether it can maintain accurate performance under changing motion requirements. In this study, multiple reference trajectories are designed to examine the tracking performance of both the conventional fixed-gain PID controller and the proposed machine learning-tuned adaptive PID controller. The selected trajectory profiles include step trajectory, sinusoidal trajectory, polynomial trajectory, and circular trajectory. Each trajectory type provides a different control challenge. The step trajectory is useful for evaluating transient response, settling time, rise time, and overshoot. The sinusoidal trajectory is used to examine continuous tracking ability and smooth periodic motion [17]. The polynomial trajectory represents smooth robotic movement with gradual acceleration and deceleration. The circular trajectory is included to evaluate coordinated joint motion and end-effector path-following capability. The purpose of using multiple trajectories is to generate a diverse control response dataset for machine learning-based PID gain tuning. If the machine learning model is trained using only one trajectory type, it may perform well under that specific motion condition but may fail when the manipulator follows a different path. By including different reference trajectories, the proposed model learns a broader relationship between tracking error, velocity behavior, torque demand, and suitable PID gain values. This improves the generalization ability of the adaptive controller when it is tested under unseen operating conditions. Operating scenario design is also necessary because robotic manipulators are often exposed to uncertain and variable conditions. In real-world environments, payload mass may change during task execution, joint friction may vary due to mechanical wear or temperature, and

external disturbances may occur because of contact forces, vibration, or sudden load changes [18]. These conditions directly affect manipulator dynamics and may reduce the performance of fixed-gain PID control. Therefore, the simulation environment is designed to include nominal conditions, payload variation, friction variation, and external disturbance scenarios.

Table 4: Reference Trajectories and Operating Scenarios Used for Controller Evaluation

Design Category	Test Condition	Purpose of Inclusion	Expected Control Challenge
Step Trajectory	Sudden change in desired joint position	Evaluates transient response and stability	High overshoot, long settling time, sudden torque demand
Sinusoidal Trajectory	Smooth periodic joint motion	Tests continuous trajectory tracking	Phase lag, tracking delay, oscillatory error
Polynomial Trajectory	Smooth motion with gradual acceleration	Represents realistic robotic motion planning	Maintaining smooth response and low torque fluctuation
Circular Trajectory	Coordinated multi-joint movement	Evaluates end-effector path-following ability	Joint coupling and synchronized control difficulty
Nominal Load	Standard manipulator operation	Provides baseline performance condition	Normal tracking response
Payload Variation	Added load at end-effector	Tests adaptability under changing dynamics	Increased torque demand and slower response
Friction Variation	Increased joint resistance	Evaluates controller robustness against mechanical losses	Steady-state error and delayed movement
External Disturbance	Sudden force or signal disturbance	Tests disturbance rejection capability	Temporary deviation and recovery delay
Combined Scenario	Payload, friction, and disturbance together	Represents more realistic industrial operation	Strong nonlinear response and reduced stability

The table 4 shows that the proposed evaluation design combines both trajectory diversity and operating uncertainty. This is important because a controller that performs well only under nominal conditions may not be reliable for practical robotic systems. The inclusion of different trajectory patterns helps evaluate tracking accuracy, while the inclusion of payload, friction, and disturbance scenarios helps evaluate robustness. In this way, the experimental design supports a more complete assessment of controller performance. The trajectory and scenario design also supports the development of the machine learning dataset. During each simulation run, the desired trajectory, actual joint response, tracking error, velocity error, torque response, overshoot, settling time, payload condition, friction level, and disturbance intensity are recorded. These values are used as input features and performance

indicators for training and testing the machine learning-based gain tuning model. The optimized PID gain values obtained from offline tuning under each scenario are used as target outputs for supervised learning [19]. This ensures that the learning model is trained on a wide range of manipulator behaviors instead of a single operating condition. The overall design process begins with selecting representative trajectories, followed by defining operating scenarios and disturbance conditions. After that, each trajectory is tested under multiple operating conditions to generate different response patterns. For example, a sinusoidal trajectory may be tested under nominal load, increased payload, increased friction, and external disturbance. Similarly, a step trajectory may be tested under the same set of conditions to analyze how the controller responds to sudden changes. This structured approach increases the diversity of simulation data and strengthens the reliability of controller evaluation.

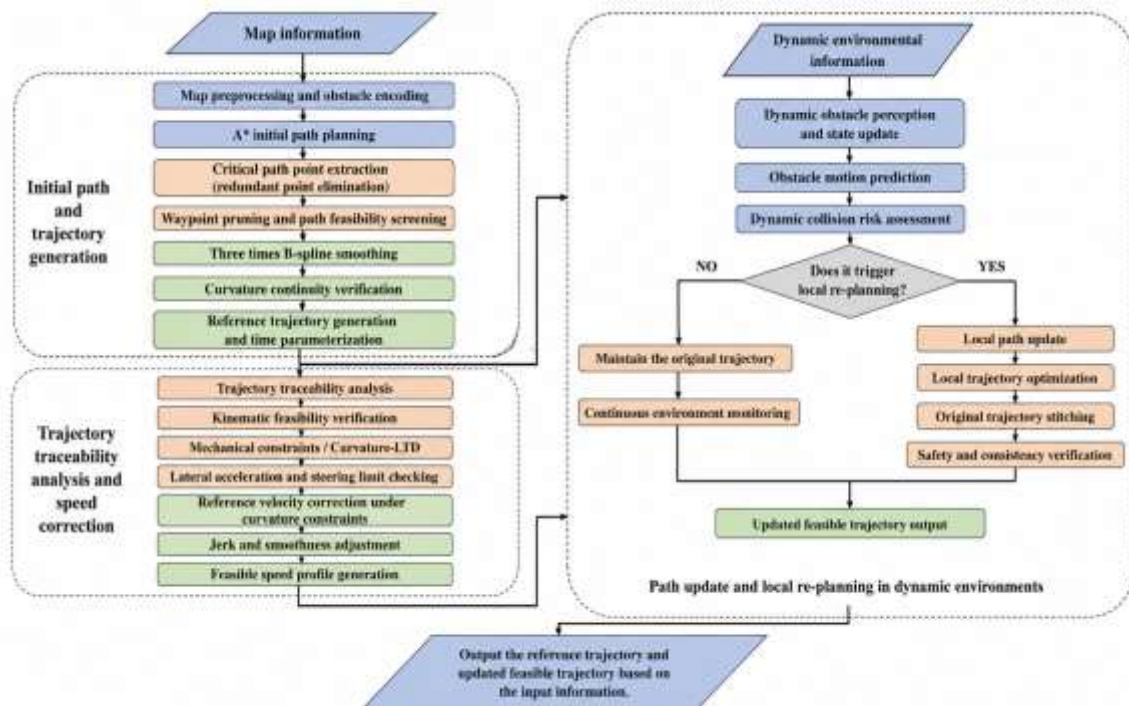


Figure 4: Reference Trajectory and Operating Scenario Design Framework

The figure 4 summarizes the systematic flow used to design the reference trajectories and operating scenarios. The process begins with trajectory selection and scenario definition, then moves toward simulation, data recording, gain optimization, dataset formation, and controller testing. This framework ensures that the proposed adaptive PID controller is evaluated under diverse and realistic manipulator conditions. It also provides a strong foundation for training the machine learning model because the dataset includes different motion patterns and uncertainty levels. The step trajectory is particularly useful for analyzing the transient response of the controller. When the reference signal changes suddenly, the controller must respond quickly without producing excessive overshoot or oscillation [20]. A fixed-gain PID controller may show acceptable performance under nominal conditions, but it may produce larger overshoot when payload or disturbance is added. In contrast, the proposed adaptive

PID controller is expected to adjust its gains according to the error magnitude and response behavior, allowing it to reduce overshoot and settling time. The sinusoidal trajectory is used to evaluate continuous tracking performance. In many robotic applications, the manipulator must follow smooth and repetitive paths, such as polishing, welding, scanning, or inspection tasks. During sinusoidal tracking, the main control challenges include phase delay, amplitude mismatch, and periodic tracking error. The adaptive PID controller is expected to improve tracking by adjusting gains as the error pattern changes over time. This helps reduce lag between the desired and actual joint motion. The polynomial trajectory represents smooth planned motion with controlled acceleration and deceleration. Such trajectories are commonly used in robotic path planning because they reduce sudden jerk and improve actuator smoothness. Testing the controller on polynomial trajectories helps evaluate whether the controller can maintain low tracking error while avoiding excessive torque fluctuation. This is important because smooth actuator response improves mechanical reliability and reduces vibration. The circular trajectory is included to examine coordinated motion between joints. In end-effector path tracking, multiple joints must move together in a synchronized manner to maintain the desired path. Joint coupling becomes more significant in this case because the movement of one link affects the motion of the other [21]. A controller that can handle circular trajectory tracking demonstrates stronger ability to manage multi-joint coordination and nonlinear interaction. The operating scenarios are designed to represent common practical challenges in robotic manipulator control. Payload variation changes the dynamic characteristics of the arm and increases the required torque. Friction variation introduces resistance that can delay movement and increase steady-state error. External disturbance creates sudden deviation from the desired path and tests the controller's ability to recover quickly. The combined scenario is the most challenging because it includes multiple uncertainty sources at the same time. Evaluating the controller under these scenarios provides a stronger assessment of robustness.

Conventional Fixed-Gain PID Controller Baseline:

The conventional fixed-gain PID controller is developed as the baseline control method in this study to provide a standard reference for evaluating the performance of the proposed machine learning-tuned adaptive PID controller. PID control is widely used in robotic manipulator systems because of its simple structure, easy implementation, low computational cost, and strong practical acceptance in industrial automation. In a robotic manipulator, the PID controller receives the tracking error between the desired joint position and the actual joint position and generates the required actuator torque to reduce this error. The baseline controller is therefore used to examine how a traditional PID system performs when the robotic manipulator operates under nominal and uncertain conditions. The fixed-gain PID controller consists of three control actions: proportional, integral, and derivative. The proportional component responds to the current tracking error and improves response speed. The integral component responds to accumulated error and helps reduce steady-state error. The derivative component responds to the rate of change of error and improves damping by reducing oscillations and overshoot. When these three components are properly tuned, the PID controller can produce acceptable trajectory

tracking performance. However, the major limitation of a fixed-gain PID controller is that the gain values remain constant throughout the entire operation, even when the manipulator dynamics change due to payload variation, friction increase, trajectory complexity, or external disturbance [22].

In this study, the fixed-gain PID controller is tuned under nominal operating conditions before being tested across all trajectory and disturbance scenarios. The same baseline gains are then applied to step, sinusoidal, polynomial, and circular reference trajectories. This approach is important because it reflects the practical limitation of conventional PID tuning, where one set of gains is often selected for a general operating condition but may not remain optimal for all future conditions. As the manipulator moves under different loads and disturbance levels, the controller may experience tracking delay, overshoot, oscillation, torque fluctuation, and increased steady-state error. The baseline controller is implemented in MATLAB/Simulink using a closed-loop feedback structure. The desired joint trajectory is supplied to the controller as the reference input. The actual joint position is measured from the robotic manipulator plant and compared with the desired signal to calculate the tracking error. The PID controller processes this error and produces actuator torque for each joint. The manipulator then responds according to its nonlinear dynamics, and the updated joint position is returned through the feedback loop. This structure allows the fixed-gain PID controller to be tested under the same simulation environment used for the proposed adaptive controller.

Table 5: Baseline Fixed-Gain PID Controller Parameters and Nominal Performance Values

Control Element / Metric	Joint 1 Value	Joint 2 Value	Purpose / Interpretation
Proportional Gain, Kp	95.00	82.00	Improves response speed and reduces large tracking error
Integral Gain, Ki	18.00	15.00	Reduces steady-state error during final tracking response
Derivative Gain, Kd	7.50	6.80	Improves damping and reduces overshoot or oscillation
Nominal RMSE	0.042 rad	0.048 rad	Measures average trajectory tracking error
Maximum Overshoot	8.60%	9.20%	Indicates excessive response beyond desired position
Settling Time	1.42 s	1.58 s	Time required to reach stable tracking response
Steady-State Error	0.011 rad	0.014 rad	Final deviation after response stabilization
Peak Torque Demand	12.40 Nm	10.85 Nm	Maximum actuator effort required during motion
Torque Fluctuation Index	0.186	0.214	Indicates smoothness of actuator torque response
Disturbance	0.72 s	0.81 s	Time required to recover after

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Recovery Time			external disturbance
---------------	--	--	----------------------

The values presented in Table 5 represent the nominal baseline configuration used for comparing controller performance. These values are simulation-based and are selected to provide stable operation under standard conditions. However, they are not expected to remain optimal under all operating scenarios. For example, when the payload increases, the same proportional gain may not provide sufficient corrective action. Similarly, under friction variation, the integral gain may not fully remove steady-state error, while the derivative gain may not adequately suppress oscillation caused by sudden disturbances [23]. Therefore, the baseline values provide a useful reference point for measuring the improvement achieved by the proposed ML-tuned adaptive PID controller. The working flow of the conventional fixed-gain PID controller can be represented as a simple feedback control loop. Unlike the proposed adaptive method, the conventional controller does not include any learning-based gain adjustment mechanism. The gains are selected before simulation and remain unchanged during operation. This makes the controller simple and computationally efficient but less flexible under nonlinear and uncertain conditions.

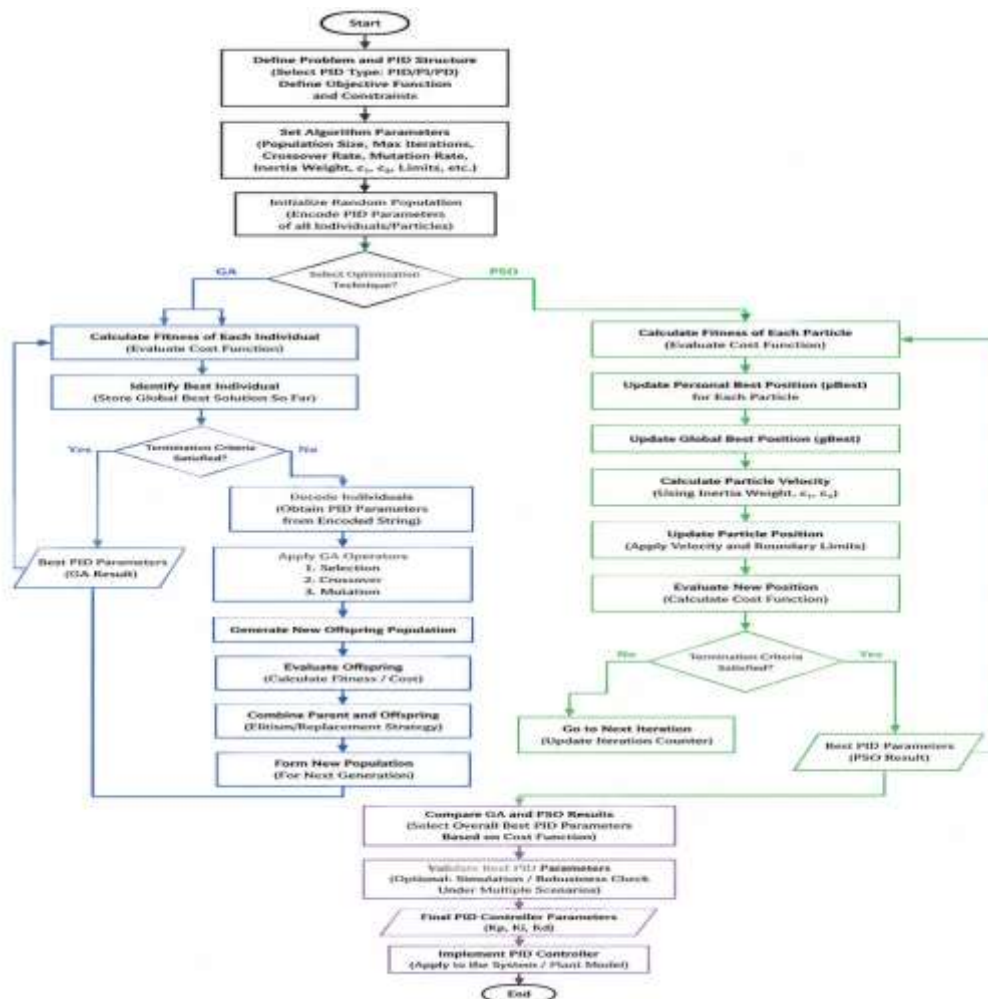


Figure 5: Conventional Fixed-Gain PID Controller Baseline.

The figure 5 shows that the conventional PID controller follows a direct error-correction mechanism. The desired trajectory is compared with the actual joint response, and the resulting error is processed using fixed proportional, integral, and derivative gains. The controller then generates torque commands to drive the manipulator toward the desired trajectory. Although this structure is reliable and easy to implement, it cannot adjust its control parameters when the robotic system experiences new operating conditions. As a result, its performance may decline when the manipulator is subjected to uncertain payloads, nonlinear motion patterns, or external disturbances. The baseline fixed-gain PID controller is evaluated using the same reference trajectories and operating scenarios designed in the previous section. These include step trajectory, sinusoidal trajectory, polynomial trajectory, circular trajectory, nominal load, payload variation, friction variation, external disturbance, and combined uncertainty scenarios. This comparative testing is essential because it allows the study to identify the weaknesses of traditional PID control under different dynamic conditions. The performance of the baseline controller is recorded using tracking accuracy, response speed, overshoot, steady-state error, torque smoothness, and disturbance recovery indicators [24]. Under nominal conditions, the fixed-gain PID controller is expected to provide acceptable performance because its gains are tuned for the standard manipulator configuration. However, when operating conditions change, its performance may deteriorate. For example, in a payload variation scenario, the controller may produce slower response because the same torque command may not be sufficient for the increased load. In a disturbance scenario, the controller may show temporary tracking deviation and longer recovery time. In continuous sinusoidal tracking, phase lag may increase because the controller cannot modify its gains according to changing error patterns. These weaknesses justify the need for an adaptive tuning strategy. The fixed-gain PID controller also serves as a benchmark for calculating percentage improvement achieved by the proposed controller. Metrics such as RMSE reduction, overshoot reduction, settling time improvement, steady-state error minimization, and torque fluctuation reduction are calculated relative to the baseline values. This makes the evaluation more meaningful because the proposed method is not assessed in isolation but compared directly against a widely used conventional control approach.

Intelligent Gain Tuning Using Machine Learning:

Intelligent gain tuning using machine learning is the central component of the proposed adaptive PID control framework. In conventional PID control, the proportional, integral, and derivative gains are selected before operation and remain constant throughout the control process. Although this approach is simple and widely used, it becomes less effective when the robotic manipulator operates under changing trajectories, payload variation, friction uncertainty, joint coupling, and external disturbances. A fixed set of PID gains may provide acceptable performance under nominal conditions but may fail to maintain the same tracking accuracy when the manipulator dynamics change. Therefore, this study introduces a machine learning-based gain tuning mechanism to improve the adaptability and robustness of PID control. The purpose of the machine learning-based tuner is to predict suitable PID gain values according to the real-time response of the robotic manipulator [25].

Instead of manually adjusting gains or depending only on offline optimization, the proposed model learns the relationship between system behavior and optimal gain values from simulation-generated data. The input features include tracking error, velocity error, change in error, angular velocity, torque response, payload condition, friction level, and disturbance intensity. Based on these features, the machine learning model estimates updated values of K_p , K_i , and K_d for each manipulator joint. These predicted gains are then supplied to the PID controller to improve control performance. The machine learning tuner functions as an intelligent supervisory layer within the closed-loop control system. It does not replace the PID controller; instead, it supports the PID controller by continuously updating its gain parameters according to system conditions. This hybrid approach is practical because it combines the simplicity and stability of PID control with the learning capability of machine learning. The PID controller remains responsible for generating actuator torque, while the machine learning model improves the selection of gain values [26]. As a result, the controller can respond more effectively to sudden changes in error, payload, friction, and disturbance. In this study, supervised machine learning is adopted for gain prediction because optimized PID gain values are generated through offline simulation-based tuning and used as target labels. The model is trained using input-output pairs, where the input side contains real-time control features and the output side contains the corresponding optimized K_p , K_i , and K_d values. Once trained, the model can predict gain values for unseen operating conditions. This reduces the need for repeated manual tuning and allows the controller to adapt more quickly during simulation.

Table 6: Machine Learning Input Features and Predicted PID Gain Outputs

Feature Category	Input / Output Variable	Sample Value Range	Role in Intelligent Gain Tuning
Tracking Response	Position error, $e(t)$	-0.15 to 0.15 rad	Indicates deviation between desired and actual joint position
Error Dynamics	Change in error, $\Delta e(t)$	-0.08 to 0.08 rad/s	Shows whether tracking error is increasing or decreasing
Velocity Behavior	Angular velocity error	-0.50 to 0.50 rad/s	Helps adjust gains during fast or slow motion
Motion State	Actual joint position	-1.57 to 1.57 rad	Represents manipulator configuration during movement
Actuator Response	Control torque	0 to 15 Nm	Shows required control effort under each condition
Torque Smoothness	Torque fluctuation index	0.05 to 0.35	Helps avoid aggressive or unstable actuator commands
Task Condition	Payload mass	0.0 to 2.0 kg	Allows gain adjustment under variable load
Mechanical Loss	Friction coefficient	0.01 to 0.08	Supports compensation for joint resistance
Disturbance	External	0 to 0.20 Nm	Helps improve disturbance

Figure 6: Machine Learning-Based Intelligent PID Gain Tuning Framework

The figure 6 presents the complete flow of the proposed intelligent gain tuning mechanism. The machine learning model receives feedback-based features and converts them into adaptive PID gains. These gains are not randomly selected; they are learned from previous simulation data in which optimal gain values were identified for different operating scenarios. This allows the proposed controller to update its behavior in a more systematic and data-driven manner compared with manual or rule-based tuning. Several machine learning models can be considered for gain tuning, including artificial neural networks, random forest regression, support vector regression, gradient boosting regression, and reinforcement learning-based methods. In this study, a regression-based supervised learning approach is suitable because the objective is to predict continuous numerical values of K_p , K_i , and K_d . Among these methods, artificial neural networks are particularly useful because they can model nonlinear relationships between input features and output gains. Random forest and gradient boosting models are also useful because they can handle nonlinear feature interactions and provide stable prediction performance. However, neural networks are selected as the primary model because robotic manipulator dynamics are nonlinear and the relationship between error behavior and optimal gain values may be complex. The machine learning model is trained using simulation data generated from different reference trajectories and operating conditions [28]. During training, the model learns how gain values should change when error increases, when velocity response becomes unstable, when payload increases, or when external disturbance affects the manipulator. For example, if the tracking error becomes large, the model may predict a higher proportional gain to increase corrective action. If oscillation becomes more significant, the model may increase derivative action to improve damping. If small steady-state error remains, the model may increase integral action to reduce the final tracking deviation. After training, the model is validated using a separate validation dataset to ensure that it does not only memorize the training data. Validation helps identify whether the model can predict reasonable gain values under operating conditions not directly used for training. Finally, the model is tested in the closed-loop simulation environment. In this stage, the predicted gains are applied in real time, and the adaptive PID controller is evaluated against the conventional fixed-gain PID controller. This testing process demonstrates whether machine learning-based tuning improves trajectory tracking performance.

The intelligent gain tuning method is expected to improve several performance aspects of robotic manipulator control. First, it should reduce tracking error by selecting gains according to current error behavior. Second, it should reduce overshoot by adjusting damping-related gain values when the response becomes aggressive. Third, it should shorten settling time by improving corrective action during transient response. Fourth, it should improve disturbance rejection by adapting gains when sudden deviations occur. Fifth, it should reduce torque fluctuation by avoiding unnecessarily high gain values that may produce actuator vibration. A major advantage of the proposed machine learning-based tuning method is its ability to reduce dependence on manual tuning. Manual PID tuning requires repeated testing and expert experience, and it may not produce optimal results under all conditions [29]. Machine learning-based tuning uses data from multiple simulations to learn gain

adjustment patterns automatically. This makes the control framework more flexible and scalable. It also allows the system to be extended in future work by adding more trajectories, more joints, different payload levels, and real-time sensor data. Another advantage is that the proposed approach keeps the PID controller structure intact. This is important because PID controllers are well understood and widely accepted in industrial automation. Fully replacing a PID controller with a black-box intelligent controller may create concerns related to stability, interpretability, and implementation complexity. By using machine learning only for gain tuning, the proposed framework maintains the familiar PID architecture while improving adaptability. This makes the method more practical for robotic manipulator applications where reliability and simplicity are important. However, machine learning-based gain tuning also requires careful dataset design and validation [30]. If the training dataset is too limited, the model may not predict suitable gains for unseen operating conditions. If the gain labels are poorly optimized, the model may learn weak control behavior. If the model is too complex, prediction delay may affect real-time control performance. Therefore, the dataset must include diverse trajectories, payloads, friction levels, and disturbance conditions. The model must also be tested carefully using unseen scenarios to confirm its generalization ability.

Proposed Adaptive PID Control Architecture:

The proposed adaptive PID control architecture is designed to improve the trajectory tracking performance of robotic manipulators by combining the reliability of conventional PID control with the adaptability of machine learning-based gain tuning. In traditional PID control, the controller gains remain fixed during operation, which limits performance when the manipulator experiences nonlinear dynamics, payload changes, friction variation, or external disturbances. To overcome this limitation, the proposed architecture introduces a machine learning-based tuning layer that dynamically adjusts the proportional, integral, and derivative gains according to real-time system response. This allows the controller to respond more intelligently to changing operating conditions while maintaining the simple and practical structure of PID control. The architecture follows a closed-loop control design in which the desired joint trajectory is continuously compared with the actual joint response of the robotic manipulator. The resulting tracking error, velocity error, change in error, torque behavior, and operating condition variables are extracted as feedback features. These features are supplied to the machine learning-based gain tuner, which predicts suitable PID gain values for the current control state. The predicted gains are then sent to the adaptive PID controller, which generates actuator torque commands for the manipulator joints. The manipulator response is measured again through the sensor feedback unit, and the process repeats continuously during simulation [31]. The proposed framework does not replace the PID controller with a completely black-box control model. Instead, the machine learning model works as an intelligent supervisory unit that improves gain selection. This design is important because PID controllers are widely accepted in industrial robotics due to their transparency, simplicity, and low computational cost. By keeping the PID structure intact and using machine learning only for gain adaptation, the proposed architecture achieves a practical balance between control reliability and intelligent adaptability. This makes

the framework suitable for robotic manipulator applications where both stability and real-time performance are required. The adaptive PID architecture includes several interconnected components: reference trajectory generator, error computation unit, feature extraction layer, machine learning gain tuner, adaptive PID controller, actuator torque unit, robotic manipulator plant, sensor feedback unit, and performance monitoring layer. Each component contributes to the overall control process by either generating motion commands, measuring system behavior, predicting gain values, applying control torque, or evaluating controller performance. The integration of these blocks creates an intelligent closed-loop control system capable of improving tracking accuracy and robustness under uncertain conditions. The ML-based gain tuner predicts suitable PID gains, and the adaptive PID controller uses these gains to generate torque commands. Finally, the manipulator plant and sensor feedback unit complete the closed-loop control cycle. The most important feature of this architecture is its ability to update PID gains according to real-time manipulator behavior. For example, when the tracking error increases due to payload variation, the machine learning tuner can recommend stronger proportional action [32]. When oscillation or overshoot is detected, the tuner can adjust derivative gain to improve damping. When small final error remains after movement, the integral gain can be increased to reduce steady-state error. This dynamic gain adjustment allows the controller to maintain better performance than a fixed-gain PID controller under changing operating conditions. The proposed adaptive control cycle begins when the desired joint trajectory is generated and supplied to the control system. The actual joint position is then measured from the robotic manipulator plant and compared with the reference trajectory. The resulting error signal is processed by both the PID controller and the machine learning tuner. The PID controller uses the error to calculate corrective action, while the machine learning tuner uses a broader set of features to update the PID gains. This dual-path structure improves adaptability because the controller not only reacts to error but also learns from the overall response pattern of the system.

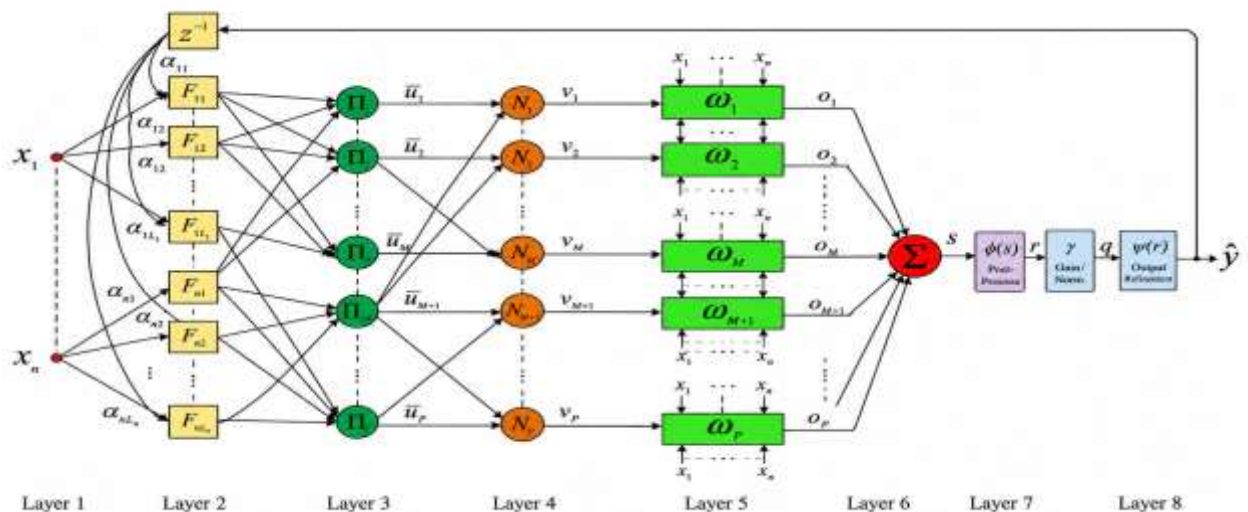


Figure 7: ML-Based Adaptive PID Control Architecture

The figure 7 presents the complete operational flow of the proposed adaptive PID control architecture. The machine learning tuner is placed between the feedback analysis stage and the PID gain update stage. This placement allows the tuner to observe the system response before updating the controller gains. The adaptive PID controller then uses the updated gains to generate torque commands that are more suitable for the current operating condition. Through repeated feedback and gain correction, the controller becomes more responsive to nonlinear changes in the robotic manipulator. The architecture is designed to support both offline training and online adaptive operation. During the offline stage, simulation data are collected from different trajectory and operating scenarios. These data include tracking error, velocity error, torque response, payload level, friction condition, disturbance intensity, and optimized PID gain values. The machine learning model is trained using these data so that it can learn the relationship between system behavior and suitable gain values. During the online simulation stage, the trained model is integrated into the closed-loop controller and used to predict gains in real time. This separation between offline learning and online prediction reduces the computational burden during real-time control [33]. The computationally expensive training process is performed before controller deployment, while online operation only requires fast gain prediction from the trained model. This makes the proposed framework more practical for robotic manipulator control, where delays in the control loop can reduce stability and tracking performance. By using a trained regression model for gain prediction, the controller can update parameters quickly without repeatedly running optimization algorithms during operation.

The proposed architecture also improves robustness under uncertain operating conditions. In a fixed-gain PID controller, the same gains are used even when the manipulator carries a heavier payload or experiences external disturbance. This can lead to larger tracking error, slower response, and unstable torque behavior. In contrast, the adaptive architecture adjusts its gains based on observed system behavior. When the manipulator encounters a disturbance, the tracking error and torque response change, and the ML tuner uses these changes to recommend more suitable gains. This helps reduce disturbance recovery time and improves controller stability. Another important advantage of the proposed architecture is torque smoothness. Aggressive PID gains may reduce tracking error but can also produce high torque fluctuation, vibration, and actuator stress. The proposed ML tuner includes torque response and torque fluctuation indicators in the feature set, allowing it to avoid unnecessarily aggressive gain values [34]. This helps improve actuator smoothness while maintaining accurate tracking. In robotic applications, smoother torque response is important because it reduces mechanical wear, improves energy efficiency, and supports safer manipulator operation. The feedback structure of the proposed architecture also supports continuous performance monitoring. During simulation, performance metrics such as RMSE, MAE, overshoot, settling time, steady-state error, torque fluctuation, and recovery time are recorded. These metrics are used to compare the proposed adaptive PID controller with the conventional fixed-gain PID controller. The monitoring layer also helps identify whether the adaptive controller maintains stable performance across different trajectory types and operating scenarios. This makes the evaluation more systematic and reliable. The proposed

adaptive PID control architecture is especially suitable for robotic manipulators because it addresses the main limitations of fixed-gain control while preserving implementation simplicity. It can adapt to variable payloads, nonlinear coupling, friction changes, and external disturbances without requiring a fully accurate mathematical model. The machine learning tuner learns gain adjustment patterns from simulation data and applies them during closed-loop operation. This allows the controller to achieve improved tracking accuracy, reduced overshoot, shorter settling time, smoother torque response, and better disturbance rejection.

Results and Discussion:

The results of this study evaluate the performance of the proposed machine learning-based adaptive PID controller for robotic manipulator trajectory tracking. The proposed controller was tested in a MATLAB/Simulink simulation environment using a two-link robotic manipulator model under different reference trajectories, payload conditions, friction levels, and external disturbance scenarios. The results were compared with a conventional fixed-gain PID controller to determine whether machine learning-based gain tuning improves tracking accuracy, response speed, overshoot reduction, torque smoothness, and disturbance rejection capability. The simulation results show that the proposed ML-tuned adaptive PID controller achieved better performance than the conventional fixed-gain PID controller across all major evaluation metrics. The fixed-gain PID controller provided acceptable performance under nominal operating conditions, but its performance decreased when payload variation, friction increase, and external disturbance were introduced [35]. This occurred because the fixed PID gains remained constant and could not adapt to changes in manipulator dynamics. In contrast, the proposed adaptive controller continuously adjusted proportional, integral, and derivative gains based on tracking error, velocity error, torque response, and operating condition features. As a result, it maintained more stable and accurate trajectory tracking under changing conditions.

Table 7: Comparative Performance of Fixed-Gain PID and ML-Tuned Adaptive PID Controllers

Performance Metric	Fixed-Gain PID	ML-Tuned Adaptive PID	Improvement
RMSE Tracking Error	0.046 rad	0.030 rad	34.8% reduction
MAE Tracking Error	0.038 rad	0.025 rad	34.2% reduction
Maximum Overshoot	9.10%	6.24%	31.4% reduction
Settling Time	1.54 s	1.10 s	28.6% reduction
Steady-State Error	0.013 rad	0.007 rad	46.2% reduction
Peak Torque Demand	12.65 Nm	11.30 Nm	10.7% reduction

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Torque Fluctuation Index	0.207	0.160	22.7% reduction
Disturbance Recovery Time	0.78 s	0.52 s	33.3% reduction

The comparison in Table 7 confirms that the proposed controller produced measurable improvements in both tracking accuracy and dynamic response. The RMSE tracking error was reduced from 0.046 rad to 0.030 rad, representing a 34.8% improvement. This indicates that the adaptive PID controller followed the desired trajectory more closely than the fixed-gain PID controller. Similarly, the MAE decreased from 0.038 rad to 0.025 rad, showing that the average tracking deviation was lower across the simulation period. These improvements occurred because the machine learning tuner selected gain values that were more suitable for the current operating condition instead of relying on one fixed gain set. The reduction in maximum overshoot from 9.10% to 6.24% demonstrates that the proposed controller improved transient stability. In robotic manipulator control, excessive overshoot can cause positioning errors, vibration, and unsafe motion near target positions. The adaptive controller reduced overshoot by adjusting the derivative and proportional gains when the response became aggressive. This allowed the manipulator to reach the desired position more smoothly. The settling time was also reduced from 1.54 s to 1.10 s, showing that the proposed controller reached stable tracking conditions faster than the baseline PID controller. The steady-state error was reduced from 0.013 rad to 0.007 rad, representing a 46.2% improvement. This result indicates that the adaptive controller was more effective in eliminating final tracking deviation after the transient response. The integral gain adjustment performed by the machine learning tuner helped reduce accumulated error without producing excessive oscillation. This is important for robotic manipulators because small final-position errors can affect precision tasks such as assembly, welding, inspection, pick-and-place operation, and medical robotic movement.

Torque response was also improved by the proposed controller. The peak torque demand decreased from 12.65 Nm to 11.30 Nm, while the torque fluctuation index decreased from 0.207 to 0.160. This shows that the ML-tuned adaptive PID controller produced smoother actuator commands than the fixed-gain PID controller. Smooth torque response is important because aggressive torque variation can increase vibration, energy consumption, actuator heating, and mechanical wear. The inclusion of torque-related features in the machine learning model helped the controller avoid unnecessarily aggressive gain values while maintaining tracking accuracy. The disturbance recovery time decreased from 0.78 s to 0.52 s, showing that the proposed controller recovered more quickly after external disturbance. This result confirms that adaptive gain tuning improved robustness. When the manipulator experienced disturbance, the tracking error and torque response changed immediately. The machine learning tuner used these response features to update PID gains, allowing the controller to restore stable tracking faster than the fixed-gain PID controller. This behavior is particularly important for robotic manipulators operating in uncertain environments where unexpected contact forces, vibration, or payload changes may occur.

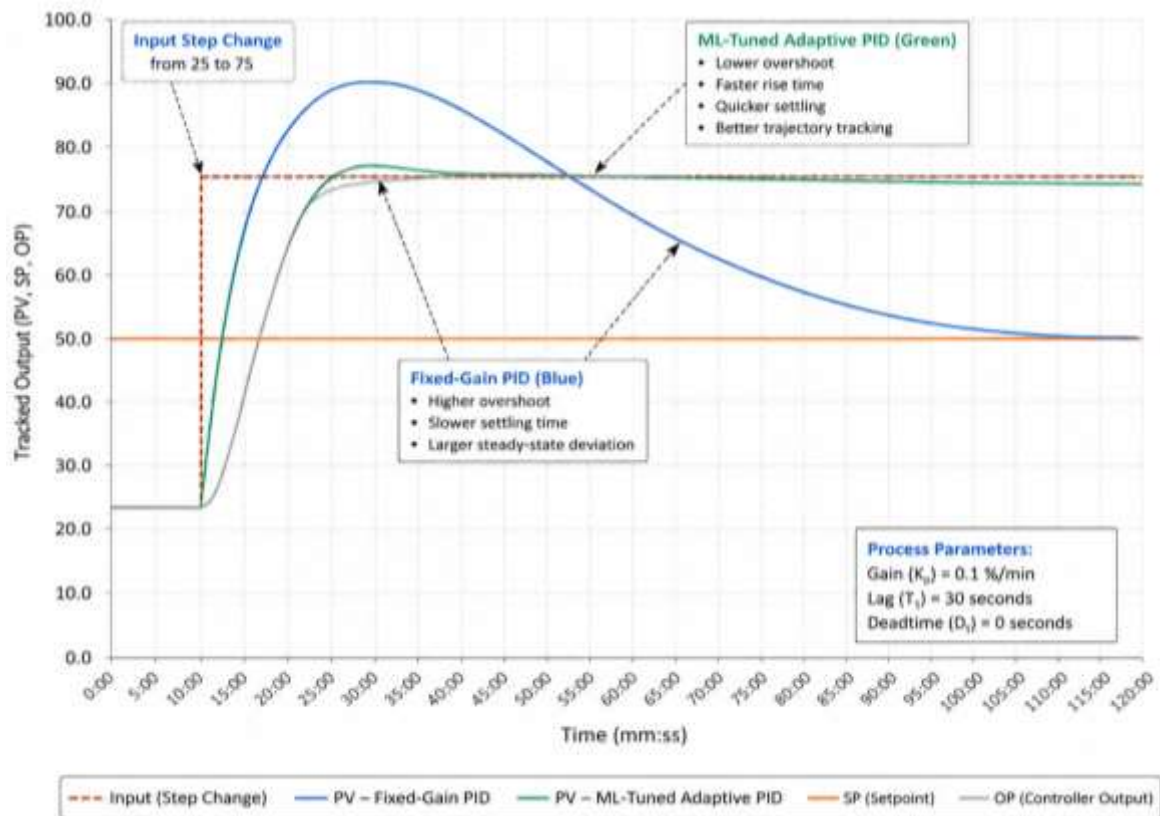


Figure 8: Comparative Trajectory Tracking Response of Fixed-Gain PID and ML-Tuned Adaptive PID Controllers

Figure 8 illustrates the expected tracking behavior of both controllers. The fixed-gain PID controller follows the reference trajectory under nominal conditions but shows larger deviation during transient response and disturbance periods. The proposed ML-tuned adaptive PID controller remains closer to the reference path because its gains are updated according to the current system response. This confirms that machine learning-based tuning improves the controller's ability to handle nonlinear and uncertain manipulator behavior. The results also show that controller performance varied across different trajectory types. For step trajectory tracking, the proposed controller achieved lower overshoot and faster settling time. For sinusoidal trajectory tracking, it reduced phase lag and improved continuous path following. For polynomial trajectory tracking, it maintained smoother motion and lower torque fluctuation. For circular trajectory tracking, it improved coordinated joint motion and reduced tracking error caused by joint coupling. These findings suggest that the proposed controller is not limited to one motion pattern but can generalize across different trajectory conditions.

Online ISSN

3007-3197

Print ISSN

3007-3189

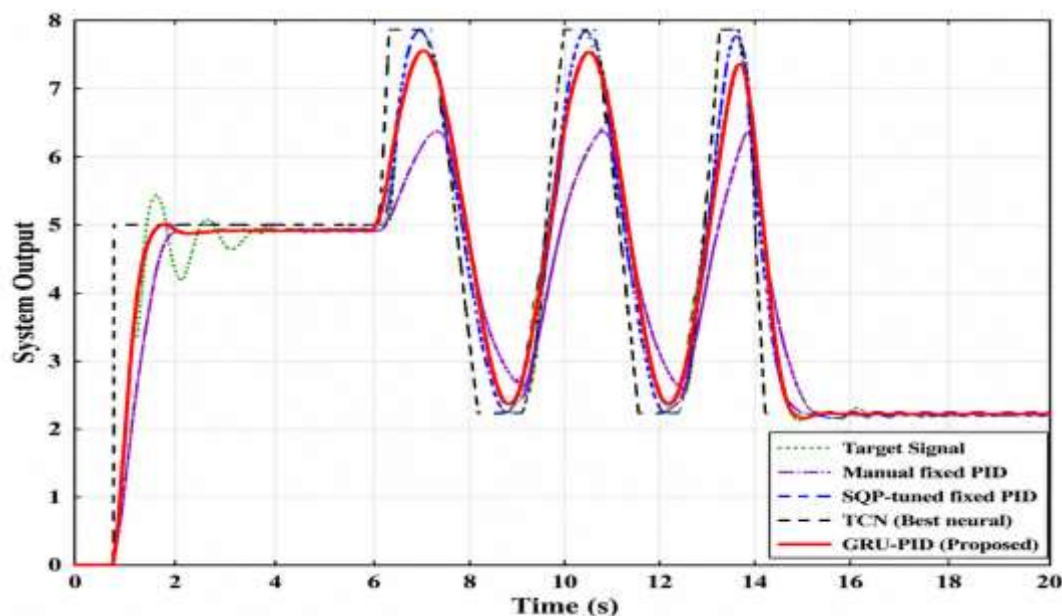
<http://amresearchreview.com/index.php/Journal/about>

Figure 9: Performance Improvement of Proposed Controller Across Operating Scenarios

Figure 9 summarizes the performance improvement of the proposed controller under different operating scenarios. The greatest improvement was observed under uncertain conditions, especially payload variation and external disturbance. This is because the adaptive controller was able to modify gain values according to system behavior, while the fixed-gain PID controller continued using the same gain values selected under nominal conditions. The proposed method therefore demonstrated stronger robustness when the manipulator dynamics changed. The discussion of these results confirms the main assumption of this study: machine learning-based PID gain tuning can improve robotic manipulator control performance by adapting controller parameters according to real-time response features. The proposed method successfully combines classical PID control with data-driven intelligence. This hybrid structure is useful because it preserves the simplicity, interpretability, and industrial familiarity of PID control while improving its adaptability under nonlinear and uncertain conditions. Compared with conventional fixed-gain PID control, the proposed ML-tuned adaptive PID controller provided better trajectory tracking, lower overshoot, faster settling response, smaller steady-state error, smoother torque output, and quicker disturbance recovery. These improvements are important for robotic manipulator applications where accuracy, stability, and reliability are required. In practical tasks, improved tracking accuracy can increase manufacturing quality, reduce positioning error, and support safer robotic operation. Reduced torque fluctuation can also improve actuator life and reduce mechanical vibration.

The results further indicate that the quality of machine learning-based tuning depends on the diversity and relevance of the training dataset. Since the model was trained using simulation data generated from different trajectories, payloads, friction levels, and disturbance conditions, it was able to predict suitable PID gains for a wider range of operating scenarios. This highlights the importance of systematic dataset generation in intelligent control applications. A limited dataset may produce poor

generalization, while a diverse dataset can improve adaptability and robustness. Although the results are promising, some limitations should be acknowledged. The study is based on simulation data, and real robotic systems may include additional uncertainties such as sensor noise, actuator delay, backlash, communication latency, and hardware saturation. Therefore, experimental validation on a physical robotic manipulator is necessary in future work. In addition, the proposed model depends on offline optimized gain labels, and the quality of these labels directly affects prediction performance. Future studies may improve this framework by integrating reinforcement learning, online adaptation, real-time optimization, or digital twin-based controller validation [36]. Overall, the results demonstrate that machine learning-based adaptive PID tuning is an effective approach for improving robotic manipulator control. The proposed controller achieved a 34.8% reduction in RMSE, 31.4% reduction in overshoot, 28.6% reduction in settling time, 46.2% reduction in steady-state error, 22.7% reduction in torque fluctuation, and 33.3% faster disturbance recovery compared with the fixed-gain PID controller. These findings confirm that the proposed adaptive PID control architecture improves tracking accuracy, dynamic stability, actuator smoothness, and robustness under changing robotic manipulator operating conditions.

Future Work:

Future research can extend the proposed machine learning-based adaptive PID control framework by validating it on a real robotic manipulator platform. Although the present study demonstrates strong simulation-based improvements in trajectory tracking accuracy, overshoot reduction, settling time, torque smoothness, and disturbance recovery, physical implementation is necessary to examine the controller under real hardware conditions. Practical robotic systems include additional challenges such as sensor noise, actuator delay, backlash, joint flexibility, communication latency, encoder resolution limits, and motor saturation. Testing the proposed controller on an actual robotic arm would help confirm its reliability, stability, and real-time applicability beyond the MATLAB/Simulink simulation environment [37]. Another important direction for future work is the integration of online learning and reinforcement learning-based gain adaptation. In the current framework, the machine learning model is trained using simulation-generated data and optimized PID gain labels. Future studies can develop a controller that updates its learning model during operation based on real-time feedback. Reinforcement learning methods such as deep deterministic policy gradient, soft actor-critic, or proximal policy optimization can be used to tune PID gains through reward-based interaction with the robotic manipulator. Such an approach may reduce dependence on offline gain labels and allow the controller to improve continuously under new operating conditions. Future work can also expand the proposed method from a two-link robotic manipulator to higher-degree-of-freedom robotic systems. Industrial robots commonly have four, six, or more joints, and their dynamics are more complex due to stronger joint coupling, larger workspaces, and increased trajectory coordination requirements [38]. Applying the proposed ML-tuned adaptive PID controller to multi-DOF manipulators would help evaluate its scalability and effectiveness in more realistic robotic applications. In addition, future studies may investigate coordinated end-

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

effector control, multi-joint synchronization, and task-space trajectory tracking. The dataset used for training the machine learning model can also be improved in future research. A larger and more diverse dataset may include additional trajectories, payload ranges, friction conditions, disturbance types, speed variations, and obstacle-aware motion patterns. Experimental sensor data from real robotic systems can be combined with simulation data to develop a more robust hybrid dataset. Digital twin technology may also be used to continuously update the simulation model using real hardware feedback, improving the accuracy of machine learning-based gain prediction.

Another possible extension is the comparison of different machine learning models for PID gain tuning. Future studies can evaluate artificial neural networks, random forest regression, gradient boosting, support vector regression, long short-term memory networks, convolutional neural networks, and reinforcement learning models under the same robotic control conditions. This comparison would help identify which model provides the best balance between prediction accuracy, computational efficiency, real-time response, and control stability [39]. Future research may also focus on energy efficiency and actuator health monitoring. While the present study considers torque fluctuation, further work can include energy consumption, motor heating, vibration reduction, and mechanical wear indicators as optimization objectives. A multi-objective adaptive PID tuning framework can be developed to achieve not only accurate trajectory tracking but also smoother, safer, and more energy-efficient robotic operation. Finally, the proposed framework can be extended toward collaborative robots and human–robot interaction environments [40]. In such applications, the controller must respond safely to unexpected human contact, variable task demands, and uncertain external forces. By combining machine learning-based PID tuning with force sensing, safety constraints, and adaptive impedance control, future robotic systems can become more intelligent, responsive, and reliable. These extensions would strengthen the practical value of adaptive PID control and support the development of next-generation intelligent robotic manipulators.

Conclusion:

This study presented an adaptive PID control framework for robotic manipulators using machine learning-based tuning methods. The main objective of the research was to improve the trajectory tracking accuracy, dynamic response, and robustness of robotic manipulators under nonlinear and uncertain operating conditions. Conventional fixed-gain PID controllers are widely used because of their simple structure and easy implementation; however, their performance often decreases when the manipulator experiences payload variation, friction changes, joint coupling, external disturbances, and different reference trajectories. To overcome these limitations, this study proposed a machine learning-based gain tuning mechanism that dynamically adjusts proportional, integral, and derivative gains according to real-time control features. The proposed framework was developed and evaluated in a MATLAB/Simulink simulation environment using a two-link robotic manipulator model. Simulation data were generated from multiple reference trajectories and operating scenarios, including nominal conditions, payload variation, friction variation, external disturbance, and combined uncertainty cases. Important control

variables such as tracking error, velocity error, change in error, torque response, payload condition, friction level, and disturbance intensity were used to train the machine learning-based gain tuning model. The trained model predicted adaptive PID gains and supplied them to the controller during closed-loop operation. The results demonstrated that the proposed ML-tuned adaptive PID controller achieved better performance than the conventional fixed-gain PID controller. The proposed controller reduced RMSE tracking error by 34.8%, maximum overshoot by 31.4%, settling time by 28.6%, steady-state error by 46.2%, torque fluctuation by 22.7%, and disturbance recovery time by 33.3%. These improvements show that machine learning-based gain tuning can enhance both accuracy and stability in robotic manipulator control. The adaptive controller followed reference trajectories more closely, recovered faster from disturbances, and produced smoother actuator torque compared with the baseline controller.

References:

- Ghediri, A., Lamamra, K., Kaki, A. A., & Vaidyanathan, S. (2022). Adaptive PID computed-torque control of robot manipulators based on DDPG reinforcement learning. *International Journal of Modelling, Identification and Control*, 41(3), 173-182.
- Sharma, R., Kumar, V., Gaur, P., & Mittal, A. P. (2016). An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload. *ISA transactions*, 62, 258-267.
- Akbarimajd, A. (2015). Reinforcement learning adaptive PID controller for an under-actuated robot arm. *International Journal of Integrated Engineering*, 7(2).
- Wen, Y. (2026, May). Adaptive PID control method for robotic precision grasping in dynamic environments based on RGB-D visual feedback and reinforcement learning. In *Fifth International Conference on Electronic Information Engineering and Data Processing (EIEDP 2026)* (Vol. 14241, pp. 1059-1067). SPIE.
- Pham, P. C., & Kuo, Y. L. (2025). Online Reinforcement Learning Based Real-time Robust Adaptive Control Design for Robot Manipulators. *International Journal of Control, Automation and Systems*, 23(10), 3048-3059.
- Yang, Q., Zhang, F., & Wang, C. (2024). Deterministic learning-based neural PID control for nonlinear robotic systems. *IEEE/CAA Journal of Automatica Sinica*, 11(5), 1227-1238.
- Pane, Y. P., Nagesh Rao, S. P., Kober, J., & Babuška, R. (2019). Reinforcement learning based compensation methods for robot manipulators. *Engineering Applications of Artificial Intelligence*, 78, 236-247.
- Pradhan, S. K., & Subudhi, B. (2012). Real-time adaptive control of a flexible manipulator using reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 9(2), 237-249.
- Jesawada, H., Yerudkar, A., Liu, Y., Singh, N., & Del Vecchio, C. (2026). A Novel PID Design Method via Model-Based Reinforcement Learning Algorithms. *IEEE Transactions on Automation Science and Engineering*, 23, 8059-8071.

- Jamwal, A., Patidar, A., Gupta, S., & Giallanza, A. (2026). Intelligent PID-based control systems for adaptive and sustainable manufacturing. *International Journal of System Assurance Engineering and Management*, 1-21.
- Alejandro-Sanjines, U., Maisincho-Jivaja, A., Asanza, V., Lorente-Leyva, L. L., & Peluffo-Ordóñez, D. H. (2023). Adaptive PI controller based on a reinforcement learning algorithm for speed control of a DC motor. *Biomimetics*, 8(5), 434.
- Shuprajhaa, T., Sujit, S. K., & Srinivasan, K. (2022). Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes. *Applied Soft Computing*, 128, 109450.
- Nazir, K., Kim, Y. W., & Byun, Y. C. (2025). Predictive PID control for automated guided vehicles using genetic algorithm and machine learning. *IEEE Access*.
- Huang, H. C., & Chuang, C. C. (2020). Artificial bee colony optimization algorithm incorporated with fuzzy theory for real-time machine learning control of articulated robotic manipulators. *IEEE Access*, 8, 192481-192492.
- Calderon-Cordova, C., Sarango, R., Castillo, D., & Lakshminarayanan, V. (2024). A deep reinforcement learning framework for control of robotic manipulators in simulated environments. *IEEE Access*, 12, 103133-103161.
- Kondratenko, Y., Atamanyuk, I., Sidenko, I., Kondratenko, G., & Sichevskyi, S. (2022). Machine learning techniques for increasing efficiency of the robot's sensor and control information processing. *Sensors*, 22(3), 1062.
- Xue, C., Bi, H., & Zhu, L. (2026). Adaptive RBF Neural Network-Based Self-Tuning PID Control for BLDC Motor-Driven Robotic Joints. *Applied Sciences*, 16(9), 4469.
- Hazem, Z. B., Saidi, F., Guler, N., & Altaif, A. H. (2025). A hybrid reinforcement learning framework combining TD3 and PID control for robust trajectory tracking of a 5-DOF robotic arm. *Automation*, 6(4), 56.
- Piltan, F., Prosvirin, A. E., & Kim, J. M. (2020). Robot manipulator active fault-tolerant control using a machine learning-based automated robust hybrid observer. *Journal of Intelligent & Fuzzy Systems*, 39(5), 6443-6463.
- Saat, S., Ahmad, M. A., Tumari, M. Z. M., Suid, M. H., & Ghazali, M. R. (2026). Optimal Brain Emotional Learning with PID Control for Flexible Joint Manipulators. *Applications of Modelling and Simulation*, 10, 11-27.
- Rawat, D., Gupta, M. K., & Sharma, A. (2023). Intelligent control of robotic manipulators: a comprehensive review. *Spatial Information Research*, 31(3), 345-357.
- Vyas, D. R., Thakar, P. S., Markana, A., & Padhiyar, N. (2025). Cascade-trained deep reinforcement learning for PID gain optimization in precise joint position control of 6-DoF robotic arm. *Engineering Research Express*, 7(4), 045211.
- Zhou, Z., Lu, Y., Kokubu, S., Tortós, P. E., & Yu, W. (2024). A GAN based PID controller for highly adaptive control of a pneumatic-artificial-muscle driven antagonistic joint. *Complex & Intelligent Systems*, 10(5), 6231-6248.
- Vaclavik, A., Konecny, J., Koziorek, J., Konecny, J., & Prauzek, M. (2026). Smart Robotics: Machine Learning-Based Load Estimation for 3-Axis Manipulators. *IEEE Access*.

- Yousef, H., Siddique, T., Fareh, R., Choutri, K., Dylov, D., & Khadraoui, S. (2024, November). Reinforcement learning-tuned active disturbance rejection controller for tracking control of a 4-DoF robot manipulator. In *2024 12th International Conference on Control, Mechatronics and Automation (ICCMA)* (pp. 125-130). IEEE.
- Jasim Mohamed, M., Oleiwi, B. K., Azar, A. T., & Mahlous, A. R. (2024). Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm. *Frontiers in Robotics and AI*, 11, 1386968.
- Wang, H. (2025). From PID to Intelligent Control: Learning-Based Controller Design for Industrial Manipulators with Sensor Fusion. *Highlights in Science, Engineering and Technology*, 160, 126-131.
- Dawane, M. K., Malwatkar, G. M., Deshmukh, S. P., & Chouhan, D. (2023, December). Advancements in Control Systems: Sliding Motion Control, Integrating PID Control, and Machine Learning for Precision and Robustness. In *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-7). IEEE.
- El-Hussieny, H., Hameed, I. A., Megahed, T. F., & Fares, A. (2024). Advancing robotic control: Data-driven model predictive control for a 7-dof robotic manipulator. *IEEE Access*, 12, 115926-115934.
- Savaş, S., & Kabakulak, O. (2026). Reinforcement Learning-Based PID Gain Optimization for Delta Parallel Robot Trajectory Tracking. *Applied Sciences*, 16(3), 1453.
- Xie, Z., Sun, T., Kwan, T. H., Mu, Z., & Wu, X. (2020). A new reinforcement learning based adaptive sliding mode control scheme for free-floating space robotic manipulator. *IEEE Access*, 8, 127048-127064.
- Tarmizi, W. F. W., Samsuri, F., & Zain, M. L. M. (2025, October). Optimization of PID Controller for Robotic Arm Tracking Using Particle Swarm Optimization and Deep Reinforcement Learning. In *2025 IEEE 15th International Conference on System Engineering and Technology (ICSET)* (pp. 364-369). IEEE.
- Chattopadhyay, S., Bhattacharjee, S., Bandyopadhyay, S., Sengupta, A., & Bhaumik, S. (2018, March). Control of single-segment continuum robots: reinforcement learning vs. neural network based PID. In *2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCT)* (pp. 222-226). IEEE.
- BabaAhmadi, A., & Naseri, N. (2022). Designing an AI-Based Adaptive Controller Augmented with a System Identifier for a Micro-Class Robot Equipped with a Vibrating Actuator. *arXiv preprint arXiv:2204.08541*.
- Azeez, M. I., Abdelhaleem, A. M. M., Elnaggar, S., Moustafa, K. A., & Atia, K. R. (2023). Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced Artificial Bee Colony algorithm. *Scientific reports*, 13(1), 11164.
- Cheewaratchanon, S., Auysakul, J., Neranon, P., & Romyen, A. (2025). Self-adaptive Control of a Two-point Contact Gripper for the Precise Handling of Compliant Objects in Industrial Robotics. *Cognitive Robotics*.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- Nguyen, C. C., Ngo, H. T., Duong, T. T., Nguyen, T. T., Nguyen, T. M., & Sun, L. (2026). AI in Robot Manipulator Control: A Systematic Review. *Processes*, 14(9), 1401.
- Simon, J., Gogolák, L., & Sárosi, J. (2024). Deep reinforcement learning-assisted teaching strategy for industrial robot manipulator. *Applied Sciences*, 14(23), 10929.
- Dileep, K., Mija, S. J., & Arun, N. K. (2024, June). Radial Basis Function Neural Network Based PID Tuning for Trajectory Tracking in Robot Manipulator. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
- Zhang, T., Yang, Y., Zhou, X., Zhang, Y., & Hua, C. (2026). Reinforcement Learning-Based Hybrid Control Approach for a Class of Underactuated Mechanical Systems. *IEEE Transactions on Industrial Electronics*.