DEEP REINFORCEMENT LEARNING FOR ADAPTIVE PID TUNING IN NONLINEAR SYSTEMS: TRACKING PERFORMANCE, GAIN EVOLUTION, AND ACTUATOR CONSTRAINTS

Mrs. MANASWINI NAHAK
Assistant Professor
Department of Electrical Engineering
Roland Institute of Technology, Berhampur

Mr. GOURI SANKAR PANDA
Assistant Professor
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. PINAKI BISOYI
Assistant Professor
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. PADARABINDO PANDA
Associate Professor
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Abstract*— Tuning proportional–integral–derivative (PID) controllers for nonlinear dynamic systems requires balancing tracking accuracy, transient response, and actuator demand across changing operating conditions. This study develops a twin delayed deep deterministic policy gradient (TD3) framework that adjusts PID gains online using closed-loop observations. The implementation combines bounded gain increments, derivative filtering, anti-windup compensation, and actuator saturation. Its contribution is an integrated assessment linking learning performance to gain evolution, actuator behaviour, and independent tracking outcomes, thereby distinguishing autonomous adaptation from effective control improvement. A nonlinear second-order oscillator is used to compare the adaptive controller with a numerically tuned fixed PID under nominal conditions, parameter changes, external disturbances, measurement noise, and an unseen reference trajectory. Although training returns recover substantially, the adaptive controller exhibits persistent oscillations and repeated actuator saturation in the step-reference tests. Approximate tracking root mean square errors are 0.58, 0.64, 0.46, and 0.58 for the adaptive controller, compared with 0.23, 0.23, 0.24, and 0.23 for the fixed PID under nominal, parameter-change, disturbance, and noisy conditions, respectively. Both controllers achieve approximately 0.05 for the unseen smooth reference. The findings demonstrate task-dependent performance and expose a gap between favourable training returns and reliable closed-loop operation. They motivate further investigation of constrained adaptation, saturation-aware objectives, and supervisory safeguards before practical deployment.

Keywords— Adaptive PID control, deep reinforcement learning, twin delayed deep deterministic policy gradient, nonlinear dynamic systems, autonomous gain tuning, actuator saturation, reference tracking.

I. INTRODUCTION

Reliable control of nonlinear dynamic systems remains a fundamental challenge in modern engineering, where operating conditions change, disturbances are unavoidable, and accurate mathematical models are often difficult to obtain. Applications ranging from robotic manipulation and electric drives to chemical processing and energy conversion exhibit nonlinearities, parameter variations, and interactions that complicate controller design. These characteristics can cause a controller that performs satisfactorily at one operating point to become sluggish, oscillatory, or excessively demanding on actuators elsewhere. Achieving precise tracking while maintaining acceptable transient behaviour, disturbance rejection, and control effort therefore requires tuning strategies that can accommodate the changing dynamics of the controlled system.[1][2]

The proportional–integral–derivative (PID) controller remains an attractive choice because of its structural simplicity, modest computational requirements, and compatibility with established control hardware. Its proportional, integral, and derivative actions provide a practical means of shaping the transient response, eliminating persistent tracking errors, and improving damping. However, these benefits depend strongly on the selection of controller gains. Increasing responsiveness may amplify overshoot or control activity, whereas conservative tuning may compromise tracking speed and disturbance rejection. In nonlinear systems, these trade-offs become more pronounced because the relationship between controller gains and closed-loop behaviour varies with the operating regime. Consequently, selecting a single gain combination that delivers satisfactory performance across a broad operating range can be difficult.[3]

Conventional tuning approaches, including empirical rules, frequency-response techniques, and model-based optimisation, provide useful solutions when system behaviour can be adequately characterised. Nevertheless, tuning based on a local approximation may lose effectiveness when the plant moves beyond the conditions under which that approximation was obtained. Manual retuning also becomes burdensome when operating conditions change frequently or when several competing performance requirements must be considered. Population-based optimisation methods offer additional flexibility in searching for suitable gains, but their effectiveness depends on the optimisation objective, the evaluation environment, and the computational resources available. These

considerations motivate approaches that can learn from closed-loop experience and relate gain selection to the observed condition of the system.[4]

Reinforcement learning offers a framework for addressing this problem by treating controller tuning as a sequential decision-making task. An agent observes information about system behaviour, selects a tuning action, and receives feedback that reflects the quality of the resulting response. Through repeated interaction, the agent seeks a policy that maximises cumulative reward, allowing present tuning decisions to account for their subsequent effects. Deep reinforcement learning extends this framework through neural-network function approximation, enabling policies and value functions to represent complex relationships within continuous observation and action spaces. Existing studies have demonstrated the application of this approach to PID tuning, including experimental implementations that retain a recognisable controller structure and studies involving continuous nonlinear processes.[5]

Combining deep reinforcement learning with PID control provides a useful division of responsibilities: the PID controller generates the control input, while the learning agent determines suitable gains according to the available observations and tuning objective. Depending on the implementation, learning may identify a fixed gain combination or produce a policy that adjusts gains during operation. This flexibility allows tuning objectives to account for tracking accuracy alongside control effort, oscillation, and actuator limitations. Retaining the PID structure also supports interpretation and integration with existing control architectures, an advantage demonstrated in experimental research on reinforcement-learning-based PID tuning.[6][7]

Despite this potential, effective learning does not automatically imply reliable control. Exploratory gain changes can produce undesirable transients, and a policy trained under a limited set of conditions may perform poorly when exposed to unfamiliar dynamics or disturbances. Reward design is equally consequential: an objective dominated by tracking error may encourage aggressive control actions, while excessive penalties on control effort may produce an unnecessarily slow response. Moreover, a high accumulated reward does not, by itself, establish closed-loop stability. Research on stability-preserving PID tuning has explicitly addressed this concern by incorporating a stable baseline controller and supervisory intervention during learning, demonstrating the importance of safeguards alongside performance optimisation.[8]

The continuing research need therefore extends beyond demonstrating that deep reinforcement learning can tune PID gains. It concerns establishing when, and under which operating conditions, learned tuning provides a dependable improvement over established alternative. Addressing this need requires a clear formulation of the observations, tuning actions, and reward, together with evaluation across changes in reference signals, disturbances, and plant parameters. Tracking accuracy must be considered alongside transient response, control effort, constraint satisfaction, and the consistency of performance across repeated trials. Such an assessment is necessary to distinguish improvements associated with a particular training scenario from capabilities that generalise to broader operating conditions.[9]

Against this background, the present study investigates intelligent PID controller tuning using deep reinforcement learning for nonlinear dynamic systems. The central objective is to examine how experience-driven gain selection can improve control performance while preserving the practical advantages of the PID architecture. The study focuses on the relationship between the learning formulation and the resulting closed-loop behaviour, with particular attention to tracking performance, transient characteristics, and control effort under varying operating conditions. By examining these interconnected requirements, the work seeks to clarify the potential and practical limitations of deep reinforcement learning as a systematic approach to PID tuning for nonlinear control applications.[10]

II. LITERATURE REVIEW

2.1 Conventional PID Control and Parameter Tuning

Proportional–integral–derivative control occupies a central position in industrial automation because it combines a straightforward mathematical structure with practical implementation and relatively modest computational requirements. Its three control actions provide complementary mechanisms for responding to present error, correcting accumulated error, and shaping the transient response. However, the effectiveness of these mechanisms depends on appropriate parameter selection and careful treatment of implementation issues. Åström and Hägglund identified tuning, adaptation, and improved functionality as important directions in the continued development of PID control, establishing a foundation for subsequent research on intelligent tuning methods.[11]

Conventional tuning methods generally determine controller parameters from measured process characteristics, simplified mathematical models, or prescribed frequency-domain requirements. Their accessibility makes them useful starting points for controller design. Nevertheless, parameters selected for one operating condition may deliver an unsatisfactory response when the process dynamics change. The central challenge is therefore broader than identifying gains that minimise tracking error under nominal

conditions: it involves selecting parameters that maintain a suitable balance between responsiveness, robustness, and control effort throughout the intended operating range.[12]

2.2 Challenges of PID Control in Nonlinear Dynamic Systems

Nonlinear dynamic systems introduce difficulties that cannot always be represented adequately by a single linear approximation. Their effective gain, response speed, and interactions may vary with the system state or operating point. Consequently, a PID controller tuned around one equilibrium may exhibit increased overshoot, prolonged settling, or persistent oscillations elsewhere. Recent research on reinforcement-learning-based PID tuning explicitly recognises narrow regions of approximate linearity and variable process parameters as important obstacles to conventional controller implementation. [13]

Practical implementation adds further complications. Actuator saturation can restrict the input demanded by the controller, while integral accumulation during saturation can delay recovery. Measurement noise may also affect derivative action, and abrupt gain changes can introduce undesirable transients. These considerations make it necessary to evaluate intelligent tuning methods within the complete control loop, including actuator constraints and signal processing. A reduction in nominal tracking error alone provides insufficient evidence of suitability for a nonlinear engineering application.[14]

2.3 Fuzzy, Neural, and Evolutionary Tuning Approaches

Intelligent PID tuning has developed through several complementary research directions. Fuzzy methods use linguistic rules or membership functions to relate observed control behaviour to parameter adjustments. Neural approaches learn nonlinear relationships from data, while evolutionary methods search candidate parameter combinations according to a specified objective. Hybrid strategies combine these capabilities. For example, research on fuzzy–genetic PID autotuning demonstrates the integration of rule-based control with an optimisation procedure, while experimental work on a coupled-tank system illustrates the application of genetically tuned neuro-fuzzy control to a physical process. [15]

These approaches expand the available design space, but their capabilities should be distinguished carefully. An evolutionary algorithm may identify an effective fixed gain combination without providing a mechanism for adaptation during operation. A neural network may reproduce relationships present in its training data without reliably extrapolating to unfamiliar conditions. Similarly, fuzzy tuning depends on the suitability of its rules and membership functions. These distinctions motivate a separation between parameter optimisation, gain scheduling, and experience-driven adaptation when comparing intelligent tuning methods.[16]

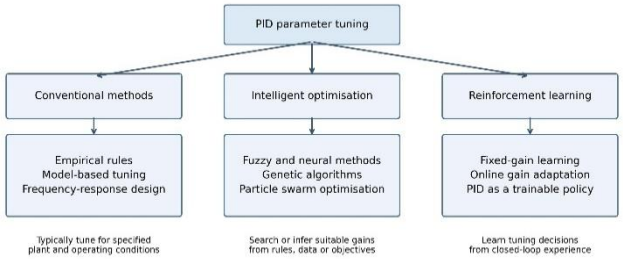


Figure 1: Conceptual classification of conventional, intelligent, and reinforcement-learning-based PID tuning approaches. The categories overlap, and hybrid implementations may combine several methods.

2.4 Reinforcement Learning for Autonomous PID Tuning

Reinforcement learning formulates controller tuning as an interaction between an agent and a dynamic environment. The agent selects an action, observes the resulting response, and receives a reward that represents the quality of its decision. In PID tuning, the action may specify controller gains or incremental gain adjustments, while observations may describe tracking error, process output, previous inputs, and relevant operating conditions. This formulation allows parameter selection to reflect the consequences of control decisions over time.[17]

Dogru and colleagues investigated autonomous PID tuning through a reinforcement learning formulation incorporating constraints, with simulation, experimental assessment, and implementation using an industrial distributed control system. Their work provides an important connection between learning algorithms and established industrial infrastructure. It also illustrates why practical relevance depends on integration, operating restrictions, and validation beyond an unconstrained numerical example. [18]

An essential distinction concerns what the learning process ultimately produces. Some methods learn a single set of gains through repeated trials; others learn a policy that adjusts gains according to changing observations. Both can use reinforcement learning, but only the latter inherently represents state-dependent gain adaptation. Accordingly, claims of autonomous or adaptive operation should be supported by the actual tuning architecture and evaluation procedure.[19]

2.5 Deep Reinforcement Learning for Continuous Gain Selection

Deep reinforcement learning extends reinforcement learning through neural-network representations of policies and value functions. This is particularly relevant when observations contain several continuous variables and controller gains are

selected from continuous ranges. Lillicrap and colleagues introduced deep deterministic policy gradient (DDPG), an actor–critic algorithm designed for continuous action spaces. Its formulation provides a methodological basis for learning continuous-valued decisions, including PID gains. However, the original study establishes a general continuous-control method rather than a PID-specific stability result. [20]

Fujimoto and colleagues subsequently introduced twin delayed deep deterministic policy gradient (TD3) to address errors associated with function approximation in actor–critic learning. The method combines twin critics, delayed policy updates, and target policy smoothing to reduce problematic value estimation and improve learning performance. These mechanisms are relevant to gain tuning because inaccurate value estimates can encourage unsuitable parameter choices. Nevertheless, improved learning behaviour must be distinguished from a formal guarantee of closed-loop stability. [21]

Soft actor–critic (SAC), developed by Haarnoja and colleagues, offers another continuous-control approach through a stochastic policy and an entropy-regularised objective. Its formulation encourages exploration while pursuing cumulative reward. For PID tuning, this creates a possible alternative to deterministic gain-selection policies. The choice between DDPG, TD3, and SAC should therefore be examined in relation to interaction cost, training consistency, exploration requirements, and deployment constraints rather than assumed universal superiority.

2.6 PID Controllers as Structured Learning Policies

A significant development in this field is the use of PID structure within the learning policy itself. Lawrence and colleagues demonstrated an experimental approach in which the PID controller serves as the trainable policy. Their study examined practical issues including hardware integration, sample efficiency, input constraints, and interpretability. Retaining a familiar controller structure allows the learned result to remain accessible to engineering inspection and compatible with conventional control equipment.

This approach differs from using a neural network to generate the actuator command directly. It also differs from a supervisory network that continually adjusts PID gains. These architectures impose different requirements on implementation and verification: learning fixed PID parameters primarily concerns the quality of the final controller, whereas continually changing gains introduces additional questions about adaptation speed, switching transients, and the behaviour of the resulting time-varying closed loop.[22]

2.7 DRL-Based PID Tuning for Nonlinear Processes

The application of deep reinforcement learning to nonlinear PID control has progressed beyond general benchmark demonstrations. The 2025 study *Tuning of PID Controllers Using Reinforcement Learning for Nonlinear System Control* investigated a TD3-based tuning structure for continuous nonlinear systems, including a strongly nonlinear biotechnological process, and compared the approach with classical tuning methods. This provides direct evidence that reinforcement-learning-based PID tuning for nonlinear systems is already an established research direction.

The significance of such studies lies in examining how learning interacts with process behaviour. However, the effectiveness of any tuning policy remains conditional on its observations, training environment, reward, and evaluation scenarios. A controller trained over a narrow operating region may achieve excellent local performance while offering limited evidence of generalisation. Consequently, nonlinear-process studies benefit from evaluations involving unfamiliar reference trajectories, parameter changes, disturbances, and measurement conditions, with clear separation between training and testing.[23]

2.8 Stability, Constraints, and Safe Exploration

Stability is a central concern because learning requires evaluating parameter choices whose consequences may initially be uncertain. Lakhani, Chowdhury, and Lu addressed this issue through a stability-preserving reinforcement learning framework incorporating a conservative baseline PID controller and supervisory intervention. Their method monitors performance during an episode and returns control to the baseline when a prescribed benchmark is violated. Simulation studies on a second-order-plus-dead-time process illustrate the role of intervention in managing exploration.

This work demonstrates the importance of integrating learning with explicit control safeguards. Nevertheless, its results should be interpreted within the assumptions and processes examined. Gain bounds, reward penalties, and baseline fallback mechanisms serve different purposes and do not automatically establish stability for an arbitrary nonlinear plant. Moreover, a stability assessment of fixed gains does not necessarily establish stability when those gains change continuously. The tuning mechanism and the controlled plant must therefore be considered together.[24]

2.9 Critical Synthesis and Research Gap

The reviewed literature establishes that reinforcement learning can support autonomous PID tuning, that deep actor–critic methods can represent continuous tuning decisions, and that structured policies can preserve the practical advantages of PID control. Experimental implementation and stability-aware tuning have also been investigated. Accordingly, the research gap cannot be defined simply as the absence of deep reinforcement learning for nonlinear PID control. A more defensible gap

concerns the extent to which a particular tuning formulation achieves reliable performance across operating conditions while respecting implementation constraints.[25]

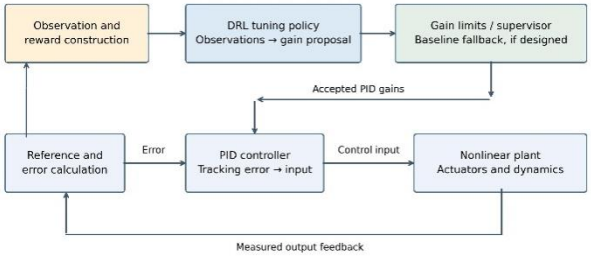


Figure 2: Conceptual framework for DRL-based adaptive PID tuning, showing observation processing, gain selection, supervisory safeguards, and output feedback. The diagram synthesises architectural concepts from the literature and does not represent a validated controller or a stability proof.

The resulting research opportunity is to develop and evaluate a clearly specified tuning framework that balances tracking accuracy, transient response, and actuator demand. Its assessment should distinguish fixed-gain optimisation from online adaptation, compare against appropriately tuned baselines, and examine sensitivity to disturbances, parameter uncertainty, and repeated training runs. Where stability guarantees are unavailable, the evidence should explicitly identify the conditions under which satisfactory behaviour has been observed. This provides a sound basis for positioning the present study as a contribution to dependable and practically assessable intelligent PID tuning.[26]

III. PROBLEM STATEMENT

Nonlinear dynamic systems present substantial control challenges because their behaviour changes with operating conditions, parameter variations, external disturbances, and actuator constraints. Although proportional–integral–derivative (PID) controllers offer a simple and practical control structure, their performance depends strongly on the selection of proportional, integral, and derivative gains. Parameters tuned for a particular operating point may become unsuitable when system dynamics change, resulting in excessive overshoot, prolonged settling time, persistent tracking errors, oscillations, or unnecessarily large control inputs. The problem is therefore to determine suitable PID gains that maintain satisfactory closed-loop performance across a defined range of nonlinear operating conditions.

Conventional tuning methods commonly rely on simplified process descriptions, empirical rules, or optimisation under predetermined conditions. These approaches can provide effective nominal settings, but maintaining performance as the plant changes may require repeated identification, manual intervention, or additional adaptation mechanisms. Deep reinforcement learning offers a means of learning gain-selection strategies from closed-loop interaction. However, its application introduces further challenges: poorly designed rewards can encourage aggressive control actions, exploratory gain changes can produce undesirable responses, and policies trained under limited conditions may fail to generalise. Consequently, successful training alone does not establish robustness, constraint satisfaction, or closed-loop stability.

The central problem addressed in this study is the development of a deep reinforcement learning framework that selects PID gains according to observed system behaviour while balancing tracking accuracy, transient response, and control effort. The formulation must define informative observations, an appropriate gain-adjustment action space, and a reward that reflects these competing requirements. It must also account for actuator limitations and restrict unacceptable tuning actions through suitable safeguards. Where gains are adjusted during operation, the effect of adaptation on the complete closed-loop response must be assessed explicitly.

The effectiveness of the resulting controller must be established through comparison with appropriately tuned conventional PID baselines under reference changes, disturbances, and plant-parameter variations. Evaluation should determine whether improvements remain consistent across repeated trials and operating conditions beyond those used for training. The intended outcome is a practically implementable tuning strategy that improves the adaptability of PID control for the nonlinear systems examined, with its performance benefits and stability limitations supported by clearly defined evidence.

IV. MATHEMATICAL MODELLING

The proposed framework combines a nonlinear plant, a PID controller, and a deep reinforcement learning agent that adjusts the controller gains. The PID controller generates the plant input, while the agent selects gain adjustments from measured closed-loop behaviour. The following formulation provides the essential mathematical structure without assuming a particular nonlinear plant.

4.1 Nonlinear Dynamic System

The controlled process is represented in discrete time as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k, \boldsymbol{\theta}_k) + \mathbf{w}_k, \qquad y_k = h(\mathbf{x}_k) + v_k. \tag{1}$$

Here, x_k is the plant state, u_k is the applied control input, and y_k is the measured output. The nonlinear functions f(.) and h(.) describe the state evolution and output relationship, respectively. The vector θ_k contains plant parameters that may vary with operating conditions, while w_k and v_k represent process disturbances and measurement noise. The sampling interval is denoted by T_s. Although the learning agent need not use an explicit plant model, a specified

model or physical process is required to generate its interaction data.[27]

4.2 PID Controller and Actuator Constraints

The tracking error is the difference between the reference r_k and measured output. A discrete PID controller with a filtered derivative and back-calculation anti-windup is expressed as

$$\begin{aligned} e_k &= r_k - y_k, \\ D_k &= \alpha D_{k-1} + (1-\alpha)\frac{e_k - e_{k-1}}{T_s}, \\ u_k^* &= K_{p,k}e_k + K_{i,k}I_k + K_{d,k}D_k, \\ u_k &= \mathrm{clip}(u_k^*, u_{\min}, u_{\max}), \\ I_{k+1} &= I_k + T_s[e_k + K_{\mathrm{aw}}(u_k - u_k^*)]. \end{aligned} \tag{2}$$

The quantities I_k and D_k represent the integral state and filtered error derivative. The coefficient $0 \leq \propto < 1$ determines derivative filtering, and $K_{aw} > 0$ controls anti-windup correction. The unsaturated command u_k^* is restricted to the actuator limits u_{min} and u_{max}. Unlike a fixed-gain PID controller, the gains $K_{p,k}$, $K_{i,k}$, and $K_{d,k}$ can change according to the learned tuning policy.[28]

4.3 Reinforcement Learning Observations and Actions

The agent receives an observation vector containing information relevant to tracking and controller behaviour:

$$\mathbf{o}_k = [r_k,\ y_k,\ e_k,\ I_k,\ D_k,\ u_{k-1},\ \mathbf{K}_k^{\mathsf{T}}]^{\mathsf{T}}, \qquad \mathbf{K}_k = [K_{p,k}, K_{i,k}, K_{d,k}]^{\mathsf{T}}. \tag{3}$$

These observations should be normalised before entering the neural network. For a plant whose internal dynamics cannot be inferred adequately from current measurements, a history of observations or a recurrent policy may be required. Thus, the observation vector is not automatically equivalent to the full Markov state.

The policy produces bounded gain increments:

$$\mathbf{a}_k = \pi_\phi(\mathbf{o}_k) \in [-1,1]^3, \qquad \mathbf{K}_{k+1} = \mathrm{clip}\,(\mathbf{K}_k + \mathbf{S}\mathbf{a}_k, \mathbf{K}_{\min}, \mathbf{K}_{\max}) \tag{4}$$

Here, $\pi_\emptyset$ is the policy parameterised by neural-network weights $\emptyset$. The diagonal matrix S specifies the maximum adjustment of each gain per tuning step, and the clipping operation enforces componentwise gain limits. The updated gains are used at the next control step. Gain bounds and limited increments restrict tuning actions but do not independently guarantee stability.[29]

4.4 Reward Function

The reward penalises tracking error, actuator demand, and abrupt changes in control input:

$$R_k = -\left[w_e\left(\frac{e_k}{e_s}\right)^2 + w_u\left(\frac{u_k}{u_s}\right)^2 + w_{\Delta u}\left(\frac{u_k - u_{k-1}}{\Delta u_s}\right)^2\right] - \lambda\, 1_{V_k}. \tag{5}$$

The non-negative weights w_e, w_u, and $w_{\Delta u}$ determine the relative importance of tracking accuracy, control effort, and input smoothness. The positive constants e_s, u_s, and Δu_s provide consistent scaling. The indicator 1_{V_k} equals one when a predefined operating-limit violation occurs, and lambda > 0 specifies its penalty. Physical constraint enforcement or supervisory intervention must operate separately from this reward penalty.[30]

4.5 Learning Objective

The agent seeks a policy that maximises the expected discounted reward over an episode:

$$\phi^* = \arg\max_{\phi} \mathbb{E}_{\pi_\phi}\left[\sum_{k=0}^{N-1}\gamma^k R_k\right], \qquad 0 < \gamma \leq 1. \tag{6}$$

Here, N is the episode length, and γ determines the importance assigned to future performance. The expectation accounts for variations in initial conditions, disturbances, plant parameters, and policy exploration. This objective defines the PID tuning task independently of the selected DRL algorithm. Maximising it establishes a performance-seeking policy; any claim of closed-loop stability requires additional analysis or explicitly stated validation conditions.[31]

V. MODEL IMPLEMENTATION

The proposed intelligent PID tuning framework is implemented in Python as shown in Figure 3 for execution on the Google Colab platform. It integrates a nonlinear simulation environment, a PID controller, and a twin delayed deep deterministic policy gradient (TD3) agent. The controller generates the plant input, while the learning agent adjusts the proportional, integral, and derivative gains according to observed system behaviour.

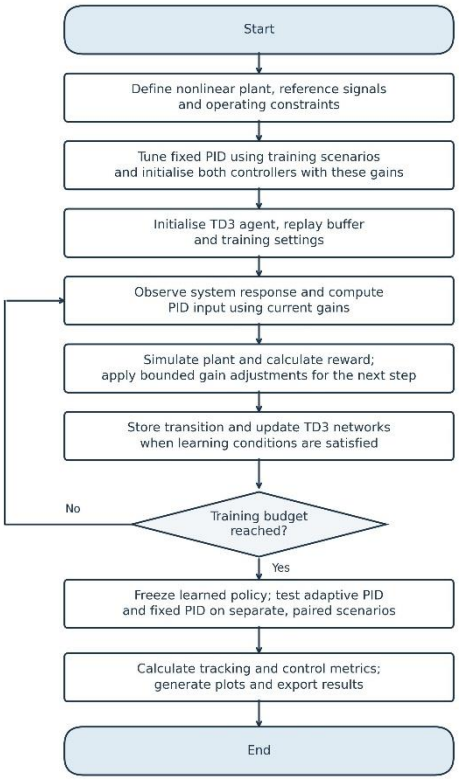


Figure 3: General implementation workflow of the TD3-based intelligent PID tuning framework, including nonlinear plant modelling, baseline tuning, agent training, independent testing, and performance evaluation.

Gymnasium provides the interaction environment, and Stable-Baselines3 supplies the TD3 implementation. The workflow comprises plant simulation, baseline tuning, agent training, independent testing, and performance assessment.

The controlled plant is represented by a nonlinear second-order oscillator incorporating damping and cubic stiffness. This structure provides a simulation setting in which the relationship between input and output changes with the system state. The plant response is calculated using fourth-order Runge–Kutta integration, with five integration sub steps within each control interval. A sampling interval of 0.05 seconds is adopted, and each episode contains 400 control steps, corresponding to 20 seconds of simulated operation. During training, plant parameters, initial conditions, reference levels, disturbances, and measurement noise are varied to expose the agent to a range of operating conditions.

A fixed-gain PID controller is established as the comparison baseline before reinforcement learning begins. Its gains are determined using bounded numerical optimisation over three designated training trajectories. Two initial gain combinations are considered to reduce dependence on a single optimisation starting point. The selected gains initialise both the fixed PID controller and the adaptive TD3–PID controller, providing a common starting condition. The PID implementation incorporates derivative filtering, actuator saturation, and back-calculation anti-windup. These features address measurement sensitivity, input limitations, and integral accumulation during saturation.[32]

The reinforcement learning agent receives a nine-component observation vector containing the reference, measured output, tracking error, integral state, filtered error derivative, previous control input, and current PID gains. Each component is scaled before being supplied to the neural network. The agent produces three continuous actions representing incremental changes to the controller gains. Both the gain increments and the resulting gain values are bounded to restrict abrupt or excessive parameter changes. The current gains generate the control input for the present interval, while the selected adjustments determine the gains used at the following interval.[33]

The TD3 agent uses an actor network to select gain adjustments and two critic networks to estimate their expected cumulative reward. The configured networks contain two hidden layers with 128 units each. Training uses a learning rate of 0.0003, a batch size of 128, a replay-buffer capacity of 100,000 transitions, and a discount factor of 0.995. Gaussian action noise supports exploration during training. Network updates begin after 1,500 interaction steps, and the default training budget is 30,000 steps. Experience replay, delayed actor updates, and gradual target-network updates support the learning process.[34]

The reward penalises tracking error, control effort, and changes in the control input. This combination encourages accurate reference following while discouraging unnecessarily large or rapidly varying actuator commands. An additional penalty is applied when predefined operating limits are violated, and the affected episode is terminated. The environment is reset at episode boundaries so that training can continue under a newly generated operating condition. These restrictions provide practical safeguards within the simulation; their presence alone does not establish closed-loop stability.[35]

After training, the learned network parameters are frozen, and the controller is evaluated on scenarios separate from those used for baseline tuning and learning. The tests cover nominal tracking, a sudden change in plant parameters, an external disturbance, increased measurement noise, and an unfamiliar reference trajectory. Although the policy weights remain fixed during testing, the policy continues to adjust PID gains in response to observations. Both controllers experience paired initial conditions and externally generated noise within each comparison, reducing differences unrelated to the control method.

Performance is assessed using root mean square error, integral absolute error, integral squared error, squared-input integral, control-input total variation, actuator saturation percentage, and episode completion. Three repetitions are configured for each test scenario, and the corresponding means and standard deviations are exported alongside individual trajectories. The implementation also generates training curves, tracking comparisons, control-input plots, and gain trajectories. The trained model, configuration, figures, and numerical results are packaged for download, supporting inspection and reproduction of the experiment. Performance improvements are determined from the resulting measurements rather than assumed in advance.[36]

VI. RESULTS AND DISCUSSION



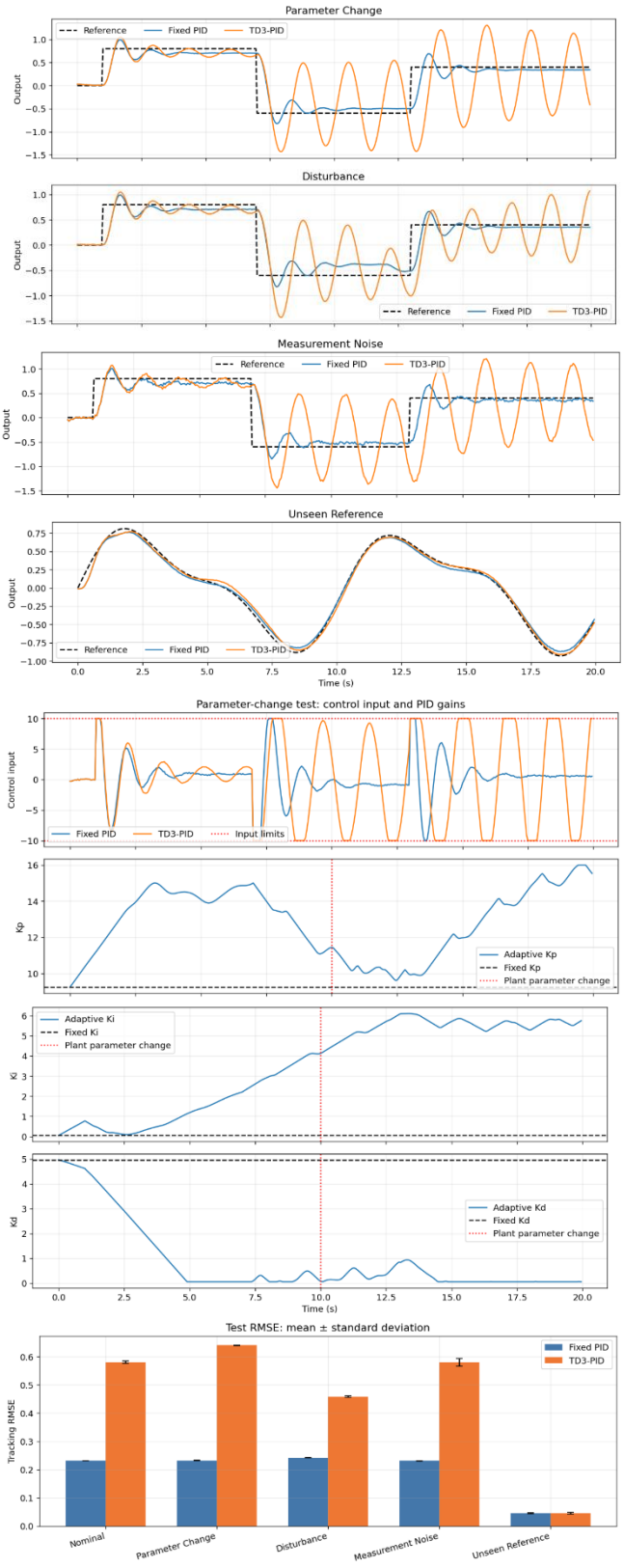


Figure 4: Training and independent evaluation of the TD3-based adaptive PID controller against a tuned fixed-gain PID controller: episode rewards and their moving average; tracking responses under nominal conditions, plant-parameter changes, external disturbance

The TD3-based adaptive PID controller was evaluated against a numerically tuned fixed-gain PID controller under five operating scenarios: nominal conditions, plant-parameter changes, external disturbances, measurement noise, and an unseen reference trajectory as presented in Figure 4. The results reveal a clear dependence on the tracking task. Although the learned policy actively adjusts the PID gains, the fixed PID controller achieves lower tracking error and better damping in all four step-reference scenarios. Both controllers provide comparable tracking accuracy for the smoothly varying unseen reference. Thus, the present implementation demonstrates autonomous gain adjustment, but its performance does not establish a consistent advantage over the tuned baseline.

The training history exhibits pronounced fluctuations between approximately 6,000 and 10,000 interaction steps, during which several episode rewards fall below −400. The ten-episode moving average reaches approximately −250 before recovering towards zero by around 14,000 steps. Thereafter, the average remains comparatively high, with a smaller deterioration towards the later stages of training. This recovery indicates improved performance within the sampled training episodes. However, the discrepancy between the training returns and the independent tracking responses shows that reward improvement alone is insufficient evidence of reliable generalisation. The curve does not establish convergence to an optimal controller or guarantee satisfactory closed-loop behaviour.

Under nominal conditions, both controllers initially exhibit overshoot following the positive reference transition. The fixed PID response subsequently becomes well damped, although a small tracking offset remains. The TD3–PID response retains oscillations, which increase substantially after the reference changes to a negative value. These oscillations persist after the final positive transition, preventing the output from settling near the desired reference. The nominal test therefore identifies a limitation of the learned policy even in the absence of an imposed parameter change or major external disturbance.

The parameter-change experiment produces a similar contrast. The fixed PID controller maintains a comparatively damped response after the plant parameters change at 10 seconds. The adaptive controller continues to exhibit large oscillations, with repeated excursions far above and below the reference. Because oscillatory behaviour is already present before the parameter change, the degradation cannot be attributed exclusively to the altered plant dynamics. Instead, the results suggest that the learned gain-adjustment policy provides insufficient damping for the step-tracking conditions encountered during testing.

During the disturbance experiment, the fixed PID response departs from its preceding tracking level but subsequently returns towards the reference region. The adaptive response remains strongly oscillatory, with no sustained settling over the affected intervals. Although its oscillation amplitude varies during and after the disturbance, this variation does not demonstrate improved disturbance rejection. Under measurement noise, small fluctuations appear in both measured outputs. Nevertheless, the dominant feature of the adaptive response remains its large, persistent oscillations. Their presence in both the nominal and noisy tests indicates that measurement noise alone does not explain the poorer TD3–PID performance.

The control-input histories provide further evidence of overly aggressive adaptive operation. In the parameter-change test, the TD3–PID controller repeatedly reaches the actuator limits of −10 and +10, including extended saturation intervals. The fixed PID controller also reaches these limits during abrupt reference transitions, but its input subsequently returns to smaller values. Repeated saturation in the adaptive case coincides with sustained output oscillations and indicates that bounded controller gains do not necessarily produce a well-damped response. Actuator limiting prevents excessive input magnitude, but it does not ensure satisfactory tracking.

The gain trajectories offer a plausible explanation for this behaviour. The adaptive proportional gain increases from approximately 9.3 to values approaching 16, while the integral gain rises from near its lower bound to approximately 6. The derivative gain decreases from about 5 to near its lower bound within the first five seconds and remains small for much of the experiment. Increasing proportional and integral action while substantially reducing derivative action is consistent with the observed oscillatory response. However, these plots establish an association rather than the individual causal contribution of each gain. Separate tests that restrict or disable particular gain adjustments would be needed to verify the underlying mechanism.

For the unseen smoothly varying reference, both controllers follow the desired trajectory closely, with modest phase lag and amplitude differences. The TD3–PID output does not exhibit the sustained large oscillations observed during step tracking. This result indicates that the learned policy can support satisfactory operation for the particular smooth trajectory examined. Its effectiveness is therefore task dependent: acceptable performance for a continuously varying reference does not imply reliable behaviour following abrupt setpoint changes.

The RMSE comparison supports these observations. Approximate readings from the plotted bars give fixed-PID RMSE values of 0.23 under nominal conditions, 0.23 under parameter changes, 0.24 under disturbance, and 0.23 under measurement noise. The corresponding TD3–PID values are approximately 0.58, 0.64, 0.46, and 0.58. For the unseen reference, both values are close to 0.05. These estimates show a substantial tracking disadvantage for the adaptive controller in the step-reference tests and comparable performance for smooth-reference tracking. Exact numerical values should be taken from the exported metric files rather than inferred from the figure. The relatively small standard-deviation bars indicate limited variation across the displayed test repetitions; they do not establish statistical significance or quantify variability across independently trained agents.

The contribution of this study lies in its integrated evaluation of TD3-based online PID gain adaptation for a nonlinear dynamic system, with particular emphasis on the relationship between learned tuning decisions and practical closed-loop behaviour. The framework combines bounded gain increments, derivative filtering, anti-windup compensation, and actuator constraints, and compares the adaptive controller with a numerically tuned fixed PID under paired test conditions. Joint examination of training rewards, tracking responses, gain trajectories, and actuator saturation reveals that improved learning returns can coexist with poorer tracking performance. This evidence-based assessment distinguishes successful implementation of autonomous tuning from demonstrated control improvement and identifies gain drift, reduced damping, and repeated saturation as specific limitations requiring further investigation.

The principal finding is that active gain adaptation must be assessed separately from effective control. In this experiment, improved training returns coexist with sustained test-time oscillations, repeated actuator saturation, and higher tracking RMSE. Further development should therefore investigate whether penalties on gain movement and saturation, slower adaptation, broader training trajectories, or supervisory restrictions can improve performance. Each modification requires independent testing. Repeated training with different random seeds is also necessary to determine whether the observed behaviour is characteristic of the formulation or specific to the trained policy presented here.

VII. CONCLUSION

This study developed and evaluated a deep reinforcement learning framework for adaptive PID tuning in a nonlinear dynamic system. A TD3 agent adjusted the proportional, integral, and derivative gains using closed-loop observations, while the PID controller retained responsibility for generating the control input. The implementation incorporated derivative filtering, anti-

windup compensation, bounded gain adjustments, and actuator saturation. Comparison with a numerically tuned fixed PID controller across five test scenarios provided a systematic assessment of tracking performance and adaptive behaviour.

The results demonstrate that autonomous gain adjustment does not necessarily yield improved control. Although training rewards recovered substantially during learning, the TD3–PID controller exhibited persistent oscillations, repeated actuator saturation, and higher tracking RMSE under nominal conditions, parameter changes, external disturbances, and measurement noise. Both controllers achieved comparable accuracy for the unseen smoothly varying reference, indicating that the learned policy's effectiveness depended strongly on the tracking task. These findings do not support a claim of general superiority or guaranteed closed-loop stability for the present formulation.

The principal contribution is an integrated simulation and evaluation framework that exposes the relationship between learned gain adaptation, actuator demand, and tracking quality. The experiments identify excessive integral action, reduced derivative action, and substantial gain variation as behaviours warranting further investigation. Future work should examine constrained adaptation, saturation-aware rewards, supervisory fallback control, and training across broader operating conditions. Validation using multiple independent training seeds and physical experiments would further establish whether these refinements produce dependable improvements over conventional PID tuning.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, no. 8, pp. 759–768, 1942.

[2] G. H. Cohen and G. A. Coon, "Theoretical consideration of retarded control," *Transactions of the ASME*, vol. 75, no. 5, pp. 827–834, 1953.

[3] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[4] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.

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

[6] Y. Li, K. H. Ang, and G. C. Y. Chong, "PID control system analysis and design: Problems, remedies, and future directions," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 32–41, 2006.

[7] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC, USA: ISA, 2006.

[8] A. Visioli, *Practical PID Control*. London, U.K.: Springer, 2006.

[9] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd ed. London, U.K.: Imperial College Press, 2009.

[10] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[11] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[12] K. J. Åström and L. Rundqwist, "Integrator windup and how to avoid it," in *Proceedings of the American Control Conference*, 1989, pp. 1693–1698.

[13] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[14] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[16] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[17] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[18] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[19] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, pp. 387–395, 2014.

[20] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proceedings of the International Conference on Learning Representations*, 2016.

[21] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1587–1596, 2018.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1861–1870, 2018.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889–1897, 2015.

[25] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.

[27] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable reinforcement learning

implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[28] G. Brockman et al., "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[29] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 3207–3214, 2018.

[30] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015.

[31] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 908–918, 2017.

[32] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, "Deep reinforcement learning with shallow controllers: An experimental application to PID tuning," *Control Engineering Practice*, vol. 121, Art. no. 105046, 2022.

[33] O. Dogru et al., "Reinforcement learning approach to autonomous PID tuning," *Computers & Chemical Engineering*, vol. 161, Art. no. 107760, 2022.

[34] A. I. Lakhani, M. A. Chowdhury, and Q. Lu, "Stability-preserving automatic tuning of PID control with reinforcement learning," *Complex Engineering Systems*, vol. 2, Art. no. 3, 2022.

[35] G. Bujgoi and D. Sendrescu, "Tuning of PID controllers using reinforcement learning for nonlinear system control," *Processes*, vol. 13, no. 3, Art. no. 735, 2025.

[36] M. A. Chowdhury, S. S. S. Al-Wahaibi, and Q. Lu, "Entropy-maximizing TD3-based reinforcement learning for adaptive PID control of dynamical systems," *Computers & Chemical Engineering*, vol. 178, Art. no. 108393, 2023.