Adaptive PID Control for Quadrotor UAVs Using Feedback Linearization and Deep Reinforcement Learning

1st Name of first author
Role, Department
University
City, Country
email

2nd Name of second author
Role, Department
University
City, country
email

3rdname of third author
Role, Department
University
City, Country
email

***Abstract*—Nonlinearity, underactuated configuration, and vulnerability to uncertainties and disturbances make the Quadrotor Unmanned Aerial Vehicles (UAVs) a complex control problem. Feedback linearization is a nonlinear control technique that leverages nonlinear transformations to obtain a linearized model of the original system. This transformation facilitates the application of linear control strategies. Fixed-gain PID controllers often suffer from performance degradation caused by parameter variations and external disturbances. This study proposes the an adaptive PID controller, leveraging the Deep Reinforcement Learning (DRL) framework for altitude and attitude control. The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is employed to tune the PID controller gains in an online, model-free environment. The TD3 agent learns control policies that dynamically modify PID gains in response to observed system behaviour through continuous interaction with the environment. The performance of the TD3-driven PID controller is assessed through extensive simulations performed on the feedback-linearized quadrotor model under varying altitude trajectories and external disturbances, followed by a comparative analysis between the proposed controller and a fixed-gain PID to evaluate improvements in robustness and tracking.**

***Index Terms*—Deep Reinforcement Learning, Quadrotor dynamics, PID Controller, Feedback linearization (FL), Actor-Critic Networks.**

I. Introduction

Advanced control strategies function as essential components to ensure stability and achieve optimal performance in autonomous systems which operate in various applications. Model Predictive control (MPC) and the Proportional-Integral-Derivative (PID) control algorithm are some of the widely adopted control techniques in the industry. PID controllers, known for their intuitive design, serve as the foundational layer for servo and regulatory control of a vast array of processes, including various applications in robotics and aerospace systems [1] [2]. The effectiveness of PID controllers is due to the tuning of their Proportional (K_p), Integral (K_i), and Derivative (K_d) gains to yield stable performance. Tuning methods proposed in literature, such as Ziegler-Nichols, Cohen-Coon require knowledge of system dynamics and responses and are limited to offline design, which renders them unable to adapt in real time to uncertainties and external disturbances. In order to address these challenges, several studies now focus on developing intelligent control algorithms, incorporating principles from Deep Reinforcement Learning (DRL), to incorporate adaptive behavior in baseline controllers.

Unlike conventional model-based techniques, DRL offers a powerful alternative by enabling control without the system model. in the design of modern control strategies. DRL is a model-free framework and, unlike classical control strategies that require system models, DRL does not require knowledge of the system dynamics. A DRL agent interacts with the system and learns optimal policies to maximize rewards over a period of time. This learning-based framework enables adaptation to parameter uncertainties and disturbances in real time. In DRL frameworks, Deep Neural Networks (DNNs) are effective function approximators that enable control in high-dimensional, continuous spaces that are characteristic of complex dynamical systems [3] [4].

Reinforcement Learning (RL) has been incorporated for the online tuning of PID controllers. In [5], a modified version of the Proximal Policy Optimisation (m-PPO) algorithm was developed for an adaptive PID controller to stabilise open-loop unstable systems. m-PPO demonstrated a reduced computational complexity, making it suitable for real-time control of a wide array of linear and nonlinear unstable processes. [6] proposed an actor-critic, model-free and off-policy algorithm known as the Deep-Deterministic Policy Gradient (DDPG) algorithm for achieving accurate trajectory tracking. In [7],a DDPG-based PID tuning method that uses phased action constraints guided by rewards to improve adaptability while maintaining closed-loop stability. [8] focused on improving the adaptive behavior of PID controllers in nonlinear systems by proposing a reinforcement learning-based control scheme, utilizing a single Radial Basis Function network to approximate actor and critic functions at the same time, with a novel temporal difference error to aid the learning process. In [9], an Advantage Actor-Critic (A2C) based deep reinforcement learning framework was proposed to tune PID gains for a robotic apple-harvest arm. A PID controller was used to model the trainable policy in the DRL framework, highlighting its

practical advantages such as compatibility with hardware [10]. Furthermore, in [11], an adaptive PID controller driven by DDPG-TD3 algorithm was designed for DC motor speed control, facilitating automatic gain tuning and demonstrating improved performance over conventional controllers under varying operating conditions.

Feedback linearization (FBL) is a nonlinear control method that performs linearization by cancelling system non-linearities through transformations [12], thereby facilitating the application of linear control laws (PID, LQR etc). In this study, FBL is implemented to the nonlinear altitude and attitude subsystems of a quadrotor, where it effectively cancels nonlinearities by remodeling the control input to formulate a linearized representation of the system [13]. This transformation facilitates subsequent design of standard linear controllers. This paper proposes the design of an adaptive Proportional-Integral-Derivative (PID) control scheme for quadrotor systems, integrating the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm . The adaptive control scheme is further enhanced through Feedback Linearization. By leveraging TD3's ability to handle continuous action spaces and reduce overestimation bias, the proposed method achieves improved robustness and adaptability when subjected to varying flight conditions and external disturbances. The performance is benchmarked against fixed-gain PID controller and evaluated using Integral Square Error (ISE), Root Mean Square Error (RMSE), and Integral of Absolute Error (IAE).

This study is presented in the following manner: Section II presents the nonlinear model of quadrotor UAV. In Section III, the nonlinear model is transformed using the Feedback Linearization (FBL) technique to obtain a linearized model suitable for PID controller design. Section IV details the real-time gain adjustment using the TD3 algorithm in response to varying operating inputs. Subsequent sections discuss the results obtained from simulations, including a comparative analysis of the proposed control strategy and conventional approaches, followed by conclusion.

II. Quadrotor Model

Accurate modeling of quadrotor dynamics is imperative for developing high-performance control algorithms for altitude and attitude tracking. The quadrotor mathematical model captures both translational and rotational dynamics, governed by six degrees of freedom (6-DOF). This research uses a nonlinear, 6-DOF quadrotor model that consists of four rotors arranged in a symmetric cross configuration, with pairs of rotors rotating in opposite directions to balance angular momentum. This arrangement enables the generation of control torques for pitch, roll, and yaw by varying the relative speeds of individual rotors. Two coordinate frames are typically used in system modelling: a moving frame that is connected to body (B-frame) and an inertial frame that is fixed to Earth (E-frame). The vector $\boldsymbol{\xi} = [x, y, z]^T$ represents quadrotor position in inertial frame and orientation is described by Euler angles $\boldsymbol{\eta} = [\phi, \theta, \psi]^T$. representing roll, pitch, and yaw, respectively, illustrated in Fig. 1. The transformation between

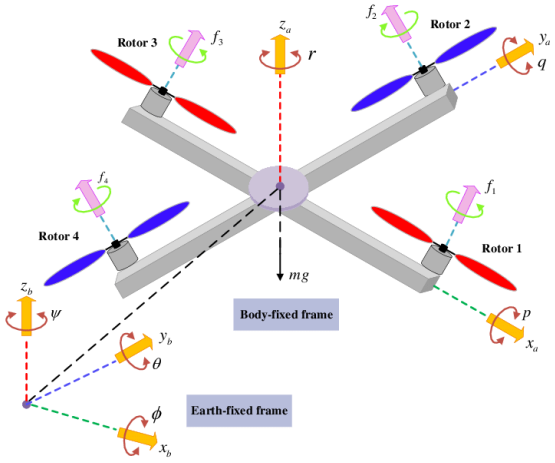


Fig. 1: Diagram of UAV

the body and inertial frames is captured using a rotation matrix and a transformation matrix, which relate linear and angular velocities between the two frames [13] [20].

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix}$$

$s(\cdot)$, $c(\cdot)$, and $t(\cdot)$ are abbreviations for $\sin(\cdot)$, $\cos(\cdot)$, and $\tan(\cdot)$, respectively. The Rotation matrix and the Transformation matrix is obtained through

$$\dot{\boldsymbol{\xi}} = R\boldsymbol{V}, \quad \dot{\boldsymbol{\eta}} = T\boldsymbol{\omega}$$

$\dot{\boldsymbol{\xi}}$ and $\boldsymbol{V}$ being linear velocity vectors and $\dot{\boldsymbol{\eta}}$ and $\boldsymbol{\omega}$ represent the angular velocity vectors. The equations describing the quadrotor dynamics are as follows [13] [17] [20]:

$$\ddot{x} = \frac{1}{m}\left(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi\right) u_4 \tag{1}$$

$$\ddot{y} = \frac{1}{m}\left(\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi\right) u_4 \tag{2}$$

$$\ddot{z} = -g + \frac{\cos\phi \cos\theta}{m} u_4 \tag{3}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} + \Omega_d \frac{J_r}{I_x} + \frac{l}{I_x} u_1 \tag{4}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \Omega_d \frac{J_r}{I_y} + \frac{l}{I_y} u_2 \tag{5}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \frac{l}{I_z} u_3 \tag{6}$$

u_1, u_2, u_3, u_4 represent control inputs into the system, which is given by,

$$u_1 = b(\Omega_4^2 - \Omega_2^2) \tag{7}$$

$$u_2 = b(\Omega_3^2 - \Omega_1^2) \tag{8}$$

$$u_3 = d(\Omega_4^2 + \Omega_2^2 - \Omega_3^2 - \Omega_1^2) \tag{9}$$

$$u_4 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{10}$$

while Ω_d representing the disturbance,

$$\Omega_d = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{11}$$

The parameters utilized for modeling the quadrotor dynamics are listed in Table I .

TABLE I: Quadrotor parameters [13]

Physical parameters	**Value**
Mass (m)	0.650 kg
inertia along x axis (I_x)	0.0075 kgm^2
inertia along y axis (I_y)	0.0075 kgm^2
inertia along z axis (I_z)	0.013 kgm^2
thrust coefficient (b)	3.13e-5 Ns^2
drag coefficient (d)	7.5e-7 Nms^2
rotor inertia (J_r)	6e-5 kgm^2
arm length (l)	0.23 m

III. Control Techniques

A. Feedback Linearization and PID Controller Design

This section presents the application of the feedback linearization technique for eliminating nonlinearities in quadrotor UAV. Feedback linearization is employed to transform the inherently nonlinear dynamics into an equivalent linear representation, thereby simplifying controller design [12] [13] [17] [18]. By canceling the system nonlinearities through input transformations, the method enables the deployment of linear control strategies for the resulting system. The proposed controller is subsequently designed and implemented on the linearized model. This is elaborated in the subsequent sections.

Consider the following non-linear system:

$$\dot{x} = f(x) + g(x)u \tag{12}$$

Here, the states are represented by x and nonlinear dynamics are captured by $f(x)$ and control input is given by u. The control law is formulated as follows:

$$u = G^{-1}(-f(x) + v) \tag{13}$$

Equation (13) substituted in (12) results in

$$\dot{x} = v \tag{14}$$

Taking the altitude and attitude subsystem of equations into consideration, the new control inputs can be defined as given below [13]

$$u_1 = I_x \left(-\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \dot{\theta}\Omega_d\frac{J_r}{I_x} + v_1 \right) \tag{15}$$

$$u_2 = I_y \left(-\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} - \dot{\phi}\Omega_d\frac{J_r}{I_y} + v_2 \right) \tag{16}$$

$$u_3 = I_z \left(-\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + v_3 \right) \tag{17}$$

$$u_4 = \frac{m}{\cos\phi\cos\theta}\left(g + v_4\right) \tag{18}$$

Substituting the new control inputs into the altitude and attitude subsystems yields,

$$\ddot{\phi} = v_1, \quad \ddot{\theta} = v_2, \quad \ddot{\psi} = v_3, \quad \ddot{z} = v_4 \tag{19}$$

Equation (19) demonstrates the transformation of nonlinear altitude and attitude equations into a linear form as shown. The linearized dynamics of the quadrotor are expressed in state-space form as shown in equation (20) [13]:

$$y = C(x), \quad \dot{x} = Ax + Bv \tag{20}$$

Four independent PID controllers are designed for each of the four decoupled control axes: (ϕ, θ, ψ, z) [13]. The compensated control law for each linearized channel is formulated as:

$$v_i = K_{P_i}e_i + K_{I_i}\int e_i\,dt + K_{D_i}\frac{de_i}{dt}$$

where e_i are the error trajectories,

$$e_\phi = \phi_d - \phi$$
$$e_\theta = \theta_d - \theta$$
$$e_\psi = \psi_d - \psi$$
$$e_z = z_d - z$$

IV. The Deep Reinforcement Learning Approach

A. Proposed Actor-Critic Architecture

Actor–Critic networks offer a natural framework for solving high-dimensional, model-free control problems [15] [16]. The proposed network consists of an actor network that learns a deterministic or stochastic control policy, mapping states to actions, and a critic that estimates a Q-value function to guide and evaluate the actor's decisions. Presence of dual networks is particularly useful in adaptive control problems to ensure real-time gain adjustment. This paper adopts the TD3 algorithm for the design of the adaptive PID control architecture. TD3 addresses two major challenges in actor-critic methods: function approximation errors and overestimation bias of Q-values [14].

1. Clipped Double Q-learning: Two critic networks Q_{θ_1} ,Q_{θ_2} are trained in parallel, and the Q-value with the least estimate is selected for policy evaluation. Adaptive PID gain updates are stabilized and the controller is safeguarded from converging to unsafe actions:

$$y = r + \gamma \min_{i\in\{1,2\}} Q_{\theta_i'}\left(s', \pi_{\phi'}(s') + \epsilon\right), \quad \epsilon \sim \text{clip}\left(\mathcal{N}(0,\sigma), -c, c\right) \tag{21}$$

2. Critic Update: minimizes the temporal difference (TD) error. The critic accurately evaluates the long-term performance of different PID gain combinations, enabling more reliable controller gain tuning.

$$\mathcal{L}(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_{\theta_i}(s,a) - y\right)^2\right], \quad i \in \{1,2\} \tag{22}$$

3. Delayed Actor Update: The actor policy π_ϕ makes less frequent updates to the PID policy parameters during training to guarantee stable and strong adaptation:

$$\nabla_\phi J(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q_{\theta_1}(s,a)\Big|_{a=\pi_\phi(s)} \cdot \nabla_\phi \pi_\phi(s)\right] \tag{23}$$

4. Soft Target Updates: prevents sudden shifts in the learned PID gains and improves convergence during continuous adaptation to changing system dynamics:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i', \quad \phi' \leftarrow \tau\phi + (1-\tau)\phi' \tag{24}$$

The actor-critic architectures are summarised in Tables II and III.

TABLE II: Actor Network architecture

Layer	Nodes	Activation Function
Input Layer	16	N/A
First Hidden Layer	256	ReLU + LayerNorm
Second Hidden Layer	256	ReLU + LayerNorm
Output Layer	12	Tanh + Scaling

TABLE III: Critic Network architecture (Per Q-Function)

Layer	Nodes	Activation Function
State Input Layer	16	N/A
Action Input Layer	12	N/A
Hidden Layer (State)	128	ReLU + LayerNorm
Hidden Layer (Action)	128	ReLU + LayerNorm
Merged Layer	128	ReLU
Output Layer	1	Identity

B. Observation and Action spaces

The environment consists of a 16-dimensional observation space and a 12-dimensional action space. The observation space includes the elements required for PID adaptation: the tracking error $e(t)$, its derivative $\frac{de(t)}{dt}$, its integral $\int e(t)\,dt$ and and the corresponding output states for each of the four axes (ϕ, θ, ψ, z). These observations are modeled into a vector of size 16×1, with no upper or lower bounds to ensure the RL agent explores the entire observation space. The actions are formulated as a continuous vector of dimension 12×1, corresponding to the three PID gains for each of the control axes. Each gain value is bounded within the range to ensure practical feasibility. The TD3 agent receives the states defined as per the observation space. The actor network maps the states to actions, facilitating continuous online tuning. Table IV lists the parameters used for the training of the proposed actor-critic network.

TABLE IV: Training Parameters

Hyperparameter	Value
Total Number of Episodes	1000
Maximum steps per episode	100
Actor Learning Rate	5×10^{-4}
Critic Learning Rate	1×10^{-3}
Gradient Threshold	1
Sample Time	0.1 s
Mini-Batch Size	256
Experience Buffer Length	2×10^{6}
Exploration Noise Mean	0
Exploration Noise Standard Deviation	0.15
Standard Deviation Decay Rate	3×10^{-3}

C. Reward Function Design

This work adopts an LQG-inspired cost formulation to define the reward signal used for training the RL agent. Linear Quadratic Gaussian (LQG) is an evaluation metric responsible for balancing tracking precision and control effort. The TD3 agent is tasked with adaptively adjusting PID gains to track attitude and altitude of a quadrotor UAV. In this research, the reward function is formulated as the negative of the LQG cost, which encourages the agent to select actions that minimize both tracking error and control effort, and maximise the negative of the cost [11] [15].

$$R = -\sum_{k=0}^{T} \left(w_{\text{error}} \cdot e_k^2 + w_{\text{action}} \cdot u_{k-1}^2 \right)$$

w_{error} and w_{action} are the weights assigned to the tracking errors and control inputs. A higher weight results in a greater penalty added to the cost, which prompts the agent to minimize the corresponding tracking error more aggressively during training. The block diagram illustrated in Fig. 2 represents a closed-loop control system designed for altitude and attitude tracking. This system employs four PID controllers for each of the four state variables to attain the desired flight performance. The TD3 agent learns from tracking error variations caused by output disturbances and stochastic reference signals in order to increase robustness. States are mapped to actions by the actor network, facilitating online tuning. The PID gains are implemented onto the PID controllers to attain the desired performance.

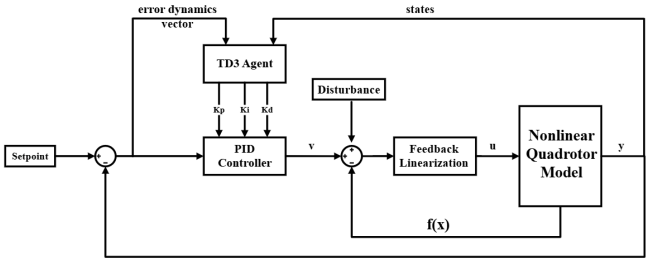


Fig. 2: Configuration of the proposed Altitude and Attitude Controller

D. Early Stopping Criteria and Reward Bonus

During the exploration phase, uncontrolled state trajectories can result in excessive accumulation of integral error, leading to high-variance gradients and instability in the process of convergence. An early stopping criterion during the training process is introduced to tackle the problem of divergence, which is also used in [5]. This mechanism defines a process-dependent constraint $y \in [y_{min}, y_{max}]$ on the system output y. During the training process, if the system outputs explode or go out of bounds, the episode is terminated immediately, and a penalty is assigned to the cost, which encourages the agent not to pursue actions that result in unsafe behavior. The

early stopping criterion ensures that the TD3 agent is trained in the absence of unstable state trajectories and eventually learns a control policy that caters to the system's operational boundaries. In addition, a reward bonus is provided upon the successful completion of the episode and a low Integral Square Error (ISE) for all the state variables.

V. Results and Discussions

The results obtained from training and simulations are discussed in this section. The RL-driven PID control scheme for a feedback-linearized Quadrotor UAV model, which consists of the nonlinear quadrotor model discussed in section II, was simulated. The control approach presented in Sections III and IV is used to execute the simulation. The proposed control architecture facilitates precise altitude and attitude tracking of the UAV, ensuring optimal performance through performance metrics such as Root Mean Square Error (RMSE), Integral of Squared Error (ISE), and Integral of Absolute Error (IAE) along with time-domain specifications. The proposed controller design introduces an adaptive characteristic to the PID controller by adjusting the PID gains upon being subjected to varying input trajectories. We arrive at optimal tracking by maximising the reward function as discussed in Section IV. The nonlinear UAV model, equations for Feedback Linearization and the PID controller architecture were developed using the MATLAB and Simulink software. The training environment and the proposed actor-critic architecture was designed on MATLAB's Reinforcement Learning Designer App. The training parameters discussed in Section IV.B were used to train the TD3 Agent. The training progress of the TD3 agent is shown in Fig.3 .

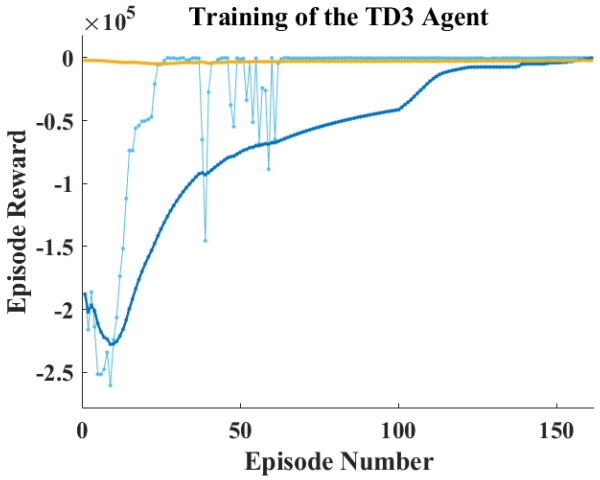
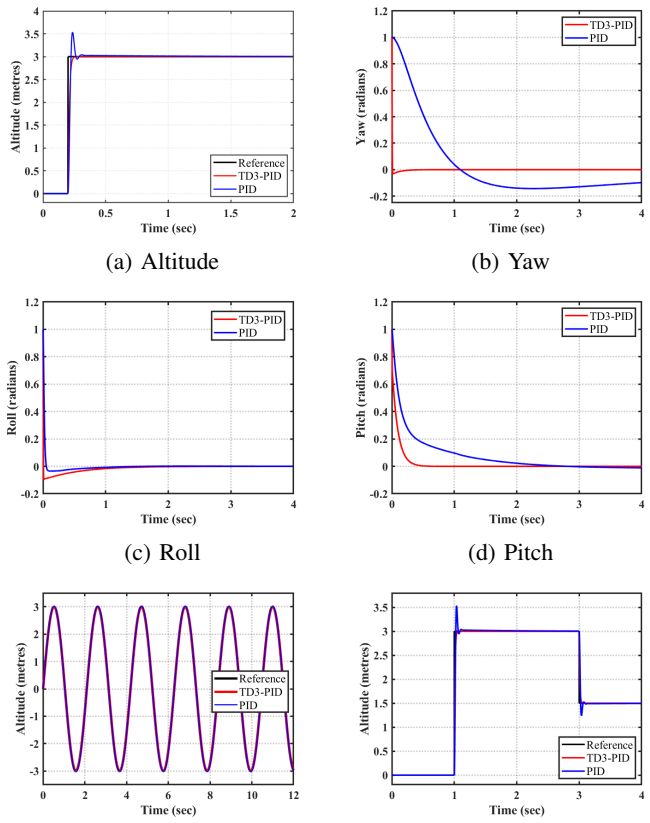


Fig. 3: Training of the TD3 Agent

The training process incorporated a threshold to stop training. The training is stopped when the average reward over a finite number of episodes exceeds a particular threshold. The performance of the adaptive PID controller is compared with the control scheme used in [13]. The feedback-linearized quadrotor UAV was made to track an altitude step change of 3 metres for altitude tracking and the remaining attitude variables were meant to drive to zero radians. In addition, the system was simulated to track a sine and a square wave input to validate the robustness of the altitude controller. The responses of the simulated system is documented in Fig.4. The time-domain metrics of both the controllers is listed in Table VI



(a) Altitude (b) Yaw

(c) Roll (d) Pitch

(e) sine wave tracking (f) square wave tracking

Fig. 4: Control performance of feedback-linearized UAV with adaptive PID controller: (a) Altitude; (b) Yaw; (c) Roll; (d) Pitch; (e) Sine wave tracking; (f) Square Wave tracking.

Table V indicates that a significant improvement in performance metrics across all control axes has been achieved. Reduced RMSE, IAE, and ISE values demonstrate improved tracking capabilities of the suggested controller. It also achieves faster response and lower overshoot compared to the baseline fixed-gain PID strategy. The controller ensures that stability and robustness are maintained across a range of operating conditions and external disturbances. This dynamic gain adjustment capability significantly enhances transient response without compromising steady-state accuracy. Overall, the proposed framework demonstrates superior adaptability and reliability in demanding UAV flight scenarios.

VI. Conclusion

This paper presents an adaptive control framework for achieving accurate attitude and altitude tracking in a quadrotor

TABLE V: Tracking Error Metrics for TD3-PID and fixed-gain PID Controllers

Error Metric	Altitude		Pitch		Yaw		Roll	
	TD3-PID	PID	TD3-PID	PID	TD3-PID	PID	TD3-PID	PID
RMSE	1.979×10^{-10}	2.867×10^{-6}	2.834×10^{-9}	2.015×10^{-4}	3.157×10^{-8}	7.735×10^{-5}	9.13×10^{-12}	3.038×10^{-7}
IAE	0.03003	0.08148	0.07034	0.3972	0.0083	1.039	0.05691	0.0429
ISE	0.05067	0.0905	0.02746	0.1059	0.0028	0.3703	0.005015	0.0127

TABLE VI: Comparative analysis of the time-domain specifications for the two control techniques

Time Domain Parameters	TD3-PID	PID
Settling Time (sec)	0.036	0.0583
Rise Time (sec)	0.0198	0.0161
Overshoot (%)	0.0	17.81
Peak Time (sec)	4.00	0.036
Peak (m)	3.00	3.535
Settling Min (m)	2.7062	2.708
Settling Max (m)	3.00	3.535

UAV by conceptualizing and deploying an adaptive PID controller trained using a TD3 agent and enhanced using feedback linearization. The inherent nonlinearities of the quadrotor are linearized through methodological transformations into a linear dynamic model using feedback linearization, enabling the application of classical control strategies. To overcome the degradation in performance of fixed-gain PID tuning under varying flight conditions, we devise a deep reinforcement learning (DRL) approach, where the TD3 agent adaptively tunes the PID gains in real time based on feedback given from the environment. The proposed TD3-driven controller is simulated under dynamic flight scenarios and external disturbances. Performance is evaluated and compared with the fixed-gain PID approach using standard time-domain specifications and error metrics such as RMSE, IAE, and ISE. The TD3-based adaptive PID control scheme demonstrates superior tracking accuracy, and enhanced disturbance rejection capabilities compared to conventional PID controllers with static gains. This study establishes the effectiveness of actor-critic reinforcement learning algorithms in adaptive control of underactuated, nonlinear aerial systems. The presented methodology offers high adaptability for dynamic and complex systems such as UAV and can be generalized to broader robotic and aerospace applications. Its modular architecture also enables seamless integration with various control schemes, making it suitable for real-world deployment in uncertain and nonlinear environments.

REFERENCES

[1] Alexandru BÂRSAN, "Position Control of a Mobile Robot through a PID controller", ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES Vol. 71 2019

[2] Hans Oersted, Yudong Ma Review of PID Controller Applications for UAVs, https://arxiv.org/abs/2311.06809

[3] Baha Zarrouki, Verena Kl¨ os, Nikolas Heppner, Simon Schwan, Robert Ritschel and Rick Voßwinkel "Weights-varying MPC for Autonomous Vehicle Guidance: a Deep Reinforcement Learning Approach", 2021 European Control Conference (ECC)

[4] K.-S. Hwang, S.-W. Tan, M.-C. Tsai, Reinforcement learning to adaptive control of nonlinear systems, IEEE Trans. Syst. Man Cybern. B 33 (3) (2003) 514–521, http://dx.doi.org/10.1109/TSMCB.2003.811112.

[5] T. Shuprajhaa, Shiva Kanth Sujit, K. Srinivasan "Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes " Applied Soft Computing, Volume 128, 2022, 109450, ISSN 1568-4946,

[6] Bartomeu Rub´i and Bernardo Morcego and Ramon Perez, "A Deep Reinforcement Learning Approach for Path Following on a Quadrotor" in 2020 European Control Conference (ECC), Saint Petersburg, Russia

[7] Ding, Y.; Ren, X.; Zhang, X.; Liu, X.; Wang, X. Multi-Phase Focused PID Adaptive Tuning with Reinforcement Learning. Electronics 2023, 12, 3925.

[8] Z. Guan, T. Yamamoto, Design of a reinforcement learning PID controller, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–6,

[9] V. van Veldhuizen, "Autotuning PID control using Actor-Critic Deep Reinforcement Learning," B.Sc. thesis, Univ. of Amsterdam, Amsterdam, The Netherlands, 2020.

[10] Nathan P. Lawrence, Michael G. Forbes, Philip D. Loewen, Daniel G. McClement, Johan U. Backström, R. Bhushan Gopaluni, Deep reinforcement learning with shallow controllers: An experimental application to PID tuning, Control Engineering Practice, Volume 121, 2022, 105046, ISSN 0967-0661,

[11] U. Alejandro-Sanjines, A. Maisincho-Jivaja, V. Asanza, L. L. Lorente-Leyva, and D. H. Peluffo-Ordóñez, "Adaptive PI Controller Based on a Reinforcement Learning Algorithm for Speed Control of a DC Motor," Biomimetics, vol. 8, no. 5, p. 434, Sep. 2023

[12] Subbareddy Chitta and Ramakalyan Ayyagari, "On the Nonlinear Control of a Class of Cruise Missiles," International Journal of Control, Automation, and Systems, vol. 23, no. 3, pp. 788-797, 2025.

[13] A. E. Taha, M. F. Rahmat, J. Mu'azu, and M. Musa, "Feedback linearization and sliding mode control design for quadrotor's attitude and altitude," 2019 IEEE 1st International Conference on Mechatronics, Automation and Cyber-Physical Computer System, Owerri, Nigeria, Apr. 2019, pp. 205–210.

[14] Fujimoto, S., van Hoof, H., & Meger, D. (2018). "Addressing Function Approximation Error in Actor-Critic Methods." Proceedings of the 35th International Conference on Machine Learning, 2018.

[15] Gheorghe Bujgoi and Dorin Sendrescu, "Tuning of PID Controllers using Reinforcement Learning for Nonlinear Systems Control" (2024), doi: 10.20944/preprints202403.0914.v1

[16] Chowdhury, M. A., & Lu, Q. (2022). "A Novel Entropy-Maximizing TD3-Based Reinforcement Learning for Automatic PID Tuning." IEEE Transactions on Systems, Man, and Cybernetics: Systems. https://doi.org/10.1109/TSMC.2022.3179646

[17] R. Olfati-Saber, "Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles", Ph.D. dissertation, Dept. of Electrical Eng. and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[18] P. Mukherjee and S. L. Waslander, "Direct adaptive feedback linearization for quadrotor control," in Proc. AIAA Guidance, Navigation and Control Conf., Minneapolis, MN, USA, Aug. 2012, pp. 1–10, doi:10.2514/6.2012-4917.

[19] V. P. Tran, M. A. Mabrok, S. G. Anavatti, I. R. Petersen, and M. A. Garratt, "Robust fuzzy Q-learning-based strictly negative imaginary tracking controllers for the uncertain quadrotor systems," *IEEE Transactions on Cybernetics*, vol. 53, no. 8, pp. 5306–5317, Aug. 2023.

[20] Zachary Dydek, Anuradha Annaswamy and Eugene Lavretsky in "Combined Composite Adaptive Control of a Quadrotor UAV in the Presence of Actuator Uncertainty" AIAA Guidance, Navigation, and Control Conference, August 2010, Toronto, Ontario, Canada, https://doi.org/10.2514/6.2010-7575