DOI: https://doi.org/10.62985/j.huit_ojs.vol26.no2E.407

OPTIMIZING UAV PID FLIGHT CONTROLLERS WITH REINFORCEMENT LEARNING: A TD3 ALGORITHM APPROACH

Minh Tu Nguyen*, Van Truong Hoang

Faculty of Missile and Gunship, Naval Academy, Nha Trang, Viet Nam

*Email: *minhtu1709@gmail.com*

Received: 19 December 2025; Revised: 18 March 2026; Accepted: 11 April 2026

ABSTRACT

This paper proposes a reinforcement learning-based approach to optimize PID control parameters for a quadcopter unmanned aerial vehicle (UAV), employing the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. The model of the UAV is derived considering nonlinearity, exterior disturbances, and strong coupling. The learning agent adaptively updates the PID gains using a reward function that optimizes tracking deviations and excessive control effort. Simulation and comparisons with other methods indicate that the TD3-tuned scheme delivers fast trajectory tracking while enhancing stability, adaptability, and control robustness.

Keywords: Reinforcement learning, UAV, PID controller, learning-based control.

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) have become increasingly important in both civilian and military applications, including surveillance, inspection, mapping, and autonomous transportation [1, 2]. Despite their growing utility, quadcopters present significant control challenges due to their underactuated nature, nonlinear dynamics, and sensitivity to external disturbances. The inherent coupling of motion and susceptibility to aerodynamic frictions demand sophisticated control mechanisms to ensure stability and precise maneuvering. Given these complexities and the variability of system parameters, there is a compelling need for robust, adaptive control frameworks such as optimized PID systems to maintain reliable operation in dynamic environments.

In recent years, various advanced control strategies have been explored to address the inherent complexities of UAV flight. Nonlinear methods such as Sliding Mode Control (SMC) and Backstepping have demonstrated significant robustness against model uncertainties and external perturbations [3, 4]. However, these techniques often suffer from the 'chattering' phenomenon or require intricate mathematical derivations that complicate real-time implementation. Model Predictive Control (MPC) has emerged as a powerful tool for high-precision trajectory tracking, yet its application is frequently hindered by a heavy computational burden that exceeds the capacity of standard onboard processors [5]. Furthermore, while Robust Control (H-infinity) strategies effectively mitigate disturbances, their performance remains highly sensitive to the accuracy of the linearized system model [6]. Despite these developments, the Proportional-Integral-Derivative (PID) controller continues to be the industry benchmark due to its structural simplicity and computational efficiency. Nevertheless, the primary limitation of PID control persists in the difficulty of optimal parameter tuning, as static gains typically fail to adapt to the highly dynamic and unpredictable operating environments of modern UAVs [7].

In fact, conventional PID tuning methods such as Ziegler–Nichols or trial-and-error typically depend on linearized or time-invariant system models, a condition rarely satisfied in

UAV flight where strong coupling effects, nonlinear aerodynamics, and time-varying operating conditions are prevalent [8, 9]. As a result, manually tuned PID gains often fail to provide optimal performance across diverse flight regimes, leading to slow convergence, overshoot, or instability, particularly when the UAV tracks agile trajectories or encounters unexpected disturbances [10, 11].

In recent years, reinforcement learning (RL) has emerged as a powerful framework for optimizing control policies in complex, nonlinear, and uncertain environments [12]. By enabling agents to learn optimal policies through interaction with the environment, RL offers advantages over traditional model-based control, especially in scenarios involving model uncertainties or high-dimensional state spaces [13]. To overcome those limitations of conventional methods for PID gains tuning, contemporary research has pivoted toward Deep Reinforcement Learning (DRL) for autonomous parameter optimization. Building on the work of Bujgoi (2025), who utilized the TD3 algorithm to optimize PID parameters for bacterial growth bioprocess control [14], this paper extends the scope to the simultaneous multi-loop tuning of a quadrotor UAV. Shifting from the relatively stable dynamics of a bioprocess to a full 6-DOF flight model introduces unique complexities, specifically the intense coupling between translational and rotational motions. The novelty of this research lies in adapting the TD3 agent to navigate a much larger and more volatile state space, ensuring stability across multiple axes (altitude, roll, pitch, and yaw) concurrently.

In this paper, we propose a TD3-based automatic PID tuning strategy designed for the closed-loop control of a 6-DOF quadrotor UAV. Unlike traditional methods, this approach leverages a deep reinforcement learning environment to autonomously refine PID coefficients for simultaneous altitude and orientation tracking. Systematic evaluations against classical PID benchmarks reveal that the proposed method not only reduces overshoot and settling time but also enhances the system's resilience to dynamic uncertainties, offering a more robust solution for complex UAV maneuvers.

The paper is organized as follows. Section 2 briefly describes the dynamic model of the quadcopter. Section 3 describes presents the design of the proposed Reinforcement learning-based pid parameter optimization for UAV flight control using the TD3 algorithm. Simulation and comparison with experimental data are introduced in Section 4. The paper ends with a conclusion and recommendation for future work.

2. SYTEM MODEL

The quadrotor UAV is modeled as a rigid body with six degrees of freedom (6-DOF) operating in three-dimensional space. Its motion is described using two primary coordinate systems: the Earth-centered inertial frame I and the body-fixed frame B, which is attached to the vehicle's center of mass.

2.1. Translational Dynamics

The total thrust T generated by the four rotors is proportional to the sum of the squares of the individual angular velocities [8]:

$$T = b\sum_{i=1}^{4}\omega_i^2 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (1)$$

where $b > 0$ denotes the thrust coefficient and ω_i represents the angular velocity of the i^{th} motor.

According to Newton's second law of motion, the translational dynamics in the inertial frame I are governed by [15]:

$$m\dot{\mathbf{v}} = m\mathbf{g} + \mathbf{R}(\phi,\theta,\psi)\begin{bmatrix}0\\0\\-T\end{bmatrix} + \mathbf{F}_{wind} \tag{2}$$

In this formulation:

* m is the total mass of the UAV.
* $\mathbf{g} = [0,0,g]^T$ is the gravitational acceleration vector.
* $\mathbf{v} = [u,v,w]^T$ is the velocity vector expressed in I .
* $\mathbf{R}(\phi,\theta,\psi) \in SO(3)$ is the rotation matrix representing the transformation from the body-fixed frame B to the inertial frame I , parameterized by the Euler angles (roll ϕ , pitch θ , and yaw ψ).
* $\mathbf{F}_{wind}$ represents the external aerodynamic disturbances (wind) expressed in inertial coordinates.

2.2. Rotational Dynamics

Assuming the quadrotor structure is symmetric along its principal axes, the inertia matrix $\mathbf{I}$ is defined as a diagonal matrix [10]:

$$\mathbf{I} = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz}) = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3}$$

The rotational motion is derived from Euler's equations for rigid body dynamics in the body-fixed frame [16]:

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \tag{4}$$

where $\boldsymbol{\omega} = [p,q,r]^T$ is the angular velocity vector in the body frame and $\boldsymbol{\tau} = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ represents the vector of control torques.

For a quadrotor in the "+" configuration, the control torques (roll, pitch, and yaw) are generated by the differential thrust and drag moments as follows [8, 10, 15-17]:

$$\begin{cases} \tau_\phi = bL(\omega_4^2 - \omega_2^2) \\ \tau_\theta = bL(\omega_1^2 - \omega_3^2) \\ \tau_\psi = d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{cases} \tag{5}$$

where:

* L denotes the distance from the center of mass to the rotor axis (arm length).
* d is the aerodynamic drag (moment) coefficient.
* The terms $\tau_\phi, \tau_\theta, \tau_\psi$ correspond to the roll, pitch, and yaw moments, respectively.

3. TWIN DELAYED DEEP DETERMINISTIC POLICY GRADIENT (TD3) ALGORITHM

This section details the theoretical framework of the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm employed for the autonomous tuning of the quadrotor's PID

controller. TD3 is an advanced actor-critic methodology designed to overcome the stability issues and overestimation biases inherent in standard reinforcement learning approaches when applied to continuous, nonlinear control systems. To establish a comprehensive understanding of the proposed tuner, we first discuss the fundamental principles of Reinforcement Learning (RL) and its mathematical formulation.

3.1. Reinforcement Learning (RL)

RL is the process of finding a mapping between situations and actions using a numerical reward signal. Generally, an agent interacts with an environment without being told which actions to take or what their effect would be on the environment. Instead, the agent has to learn a behavior by trial and error [12]. The process of RL is depicted in Figure 1.

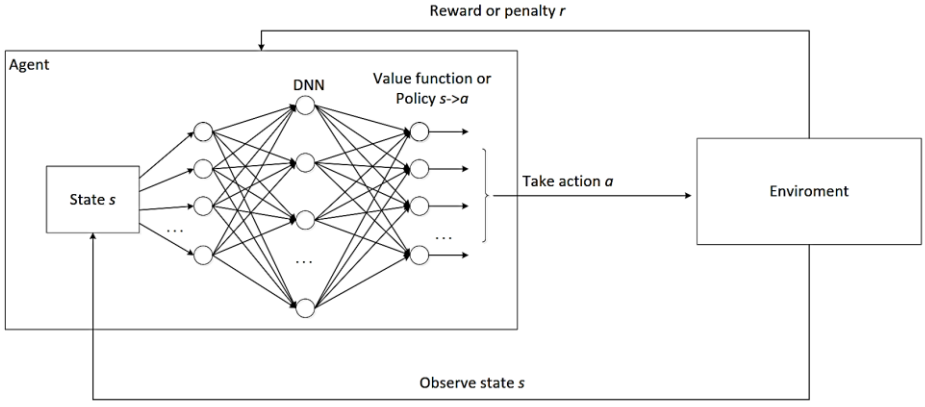

Fig. 1. The basis scheme of RL

The RL algorithms for continuous state spaces has undergone significant evolution, transitioning from classical tabular methods to sophisticated deep architectures. Early RL research primarily focused on discrete state and action spaces using dynamic programming and Bellman-based algorithms; however, these approaches faced the "curse of dimensionality" when applied to continuous environments [12]. To mitigate this, researchers initially utilized linear function approximation techniques, such as tile coding and coarse coding, to discretize or represent high-dimensional spaces [12, 18].

A paradigm shift occurred with the emergence of policy gradient methods, which directly optimize a parameterized policy rather than estimating value functions. Key milestones include the reinforce algorithm [19] and the development of actor-critic frameworks [20], both of which proved highly effective for continuous action spaces. The integration of deep learning further catalyzed this progress; while Deep Q-Networks (DQN) revolutionized high-dimensional discrete tasks [21], subsequent algorithms such as Deep Deterministic Policy Gradient (DDPG) [22] and Trust Region Policy Optimization (TRPO) [23] provided robust solutions for continuous control by leveraging neural networks as universal function approximators.

3.2. Twin Delayed Deep Deterministic Policy Gradient (TD3)

The TD3 is an advanced off-policy actor-critic algorithm that addresses the inherent overestimation bias in DDPG [24]. By employing three key mechanisms - Clipped Double Q-learning, Delayed Policy Updates, and Target Policy Smoothing - TD3 significantly enhances the robustness and convergence speed of PID parameter tuning for complex nonlinear systems such as quadrotor UAVs [14, 24].

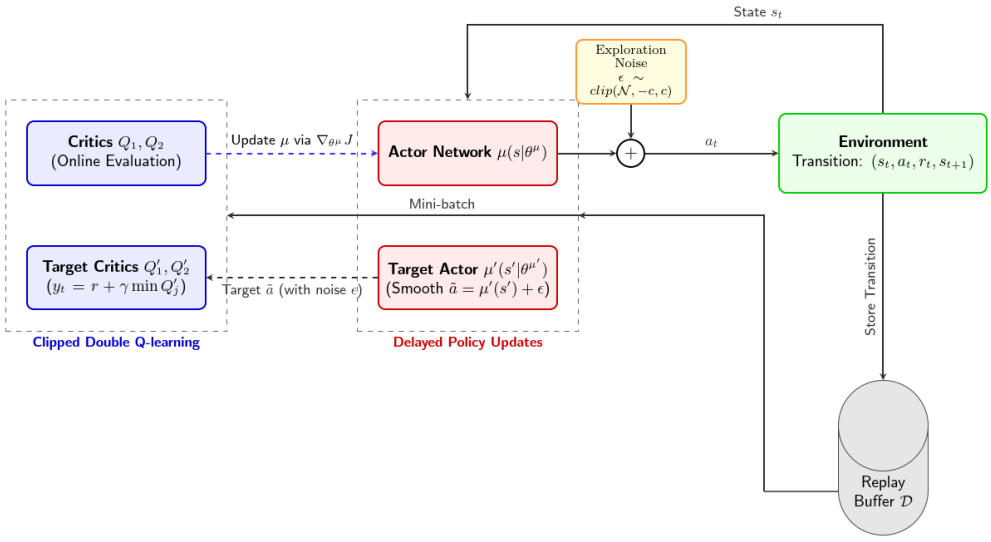

Fig. 2. Structure of TD3 Algorithm

To enhance the stability of PID parameter optimization in highly nonlinear UAV dynamics, TD3 utilizes the following mechanisms (Figure 2) [24]:

Clipped Double Q-learning: To prevent the overestimation of the value function, TD3 maintains two independent critic networks (Q_1, Q_2). The target value y_t is computed using the minimum of the two target critics:

$$y_t = r_t + \gamma \min_{j=1,2} Q_j'(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) + \grave{o} \mid \theta^{Q_j'}) \tag{6}$$

where $\grave{o} \sim \text{clip}(\text{N}(0, \sigma), -c, c)$ is the added noise for target policy smoothing.

Delayed Policy Updates: The actor network μ and target networks are updated less frequently than the critic networks (e.g., once every two critic updates), ensuring that the value function converges more reliably before guiding the policy improvement.

Target Policy Smoothing: A small amount of noise ϵ is added to the target actions to smooth out Q-value changes across similar actions, making the policy more resistant to function approximation errors.

Following the methodology established in [14], the PID tuning task is formulated as a Markov Decision Process (MDP) defined by the tuple $(\text{S}, \text{A}, \text{R}, \gamma)$:

Observation Vector (S - state space): The state s_t is defined by the error dynamics of the quadrotor (e.g., attitude or position errors):

$$s_t = [e(t), \int e(t)dt, \dot{e}(t)]^T \tag{7}$$

where $e(t)$ represents the difference between the reference trajectory and the current state.

Action Space (A): The actor network outputs the continuous gains for the PID controller:

$$a_t = [K_p, K_i, K_d]^T \tag{8}$$

These gains are dynamically applied to the low-level control loop of the UAV.

Reward Function (R): To ensure fast convergence and minimize oscillations, the reward function r_t is designed to penalize tracking errors and control effort:

$$r_t = -\left[a.e(t)^2 + b.\Delta u(t)^2 \right] \quad (9)$$

where *a,b* are weighting coefficients determined through empirical testing [14].

3.3. TD3-based PID tuning approach for UAV controller

A primary advantage of TD3 in UAV control is the reduction of overestimation bias, a common defect in other reinforcement learning algorithms that can lead to erratic motor commands or system failure [12]. By introducing a delay in policy updates, TD3 mitigates the risk of high-frequency oscillations and instability during the learning process. The synergy of reduced overestimation bias, delayed policy updates, and controlled exploration results in a stable learning convergence. This stability is vital in UAV PID tuning to ensure that the algorithm identifies a suitable solution without inducing physical fluctuations or inconsistencies in the flight control parameters.

In this paper, we consider a standard PID control architecture with the following input–output relation, as described in the proposed methodology [14]:

$$u = \left[e \quad \int e dt \quad \frac{de}{dt} \right] \times \left[K_p \quad K_i \quad K_d \right]^T \quad (10)$$

Where:

u represents the output of the actor neural network;

$K_p, K_i,$ and K_d are the PID controller parameters to be optimized;

$e(t) = v(t) - y(t)$ is the tracking error, where $y(t)$ denotes the actual state of the UAV (e.g., altitude or attitude) and $v(t)$ is the desired reference signal.

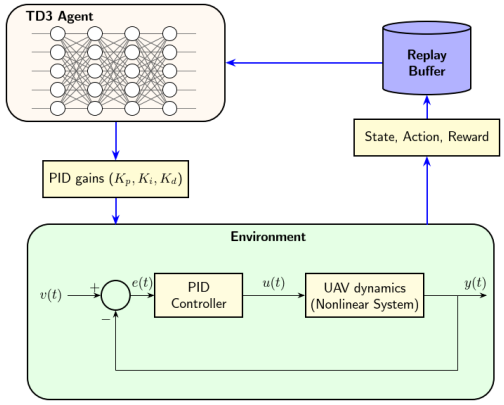

Fig. 3. TD3-based PID tuning approach for UAV control system.

The weights of the TD3 actor network are effectively the gains of the PID controller. Thus, the PID controller is modeled as a neural network with one fully connected layer, where the inputs are the system error, error integral, and error derivative. The proposed scheme operates on two distinct time scales: one for adapting the PID parameters (the weights of the actor) and another for analyzing the UAV's response to step inputs or environmental disturbances. As shown in the general structure of the TD3-based approach, the agent iteratively interacts with the flight environment to maximize the reward function, which is typically based on minimizing the settling time and overshoot while ensuring flight stability.

4. SIMULATION RESULTS

Extensive simulations and comparative analyses were conducted to evaluate the performance of the proposed controller, utilizing the quadcopter model and parameters specified in Table 1.

Table 1. Parameters of the quadcopter model

Name	Parameter	Value	Unit
Mass	m	1.1	kg
Arm Length	l	0.25	m
Gravity	g	9.81	m/s^2
Moment of Inertia about the X-axis	I_{xx}	$8{,}95.10^{-3}$	$kg.m^2$
Moment of Inertia about the Y-axis	I_{yy}	$8{,}95.10^{-3}$	$kg.m^2$
Moment of Inertia about the Z-axis	I_{zz}	0,0165	$kg.m^2$
Thrust Coefficient	b	$9{,}6.10^{-6}$	$N.s^2/rad^2$

4.1. Scenario 1: Control under High wind 10m/s (no obstacles)

In the first scenario, the primary objective is to assess the disturbance rejection capability of each controller. Both systems were tasked with following a square trajectory of $20\text{m} \times 20\text{m}$

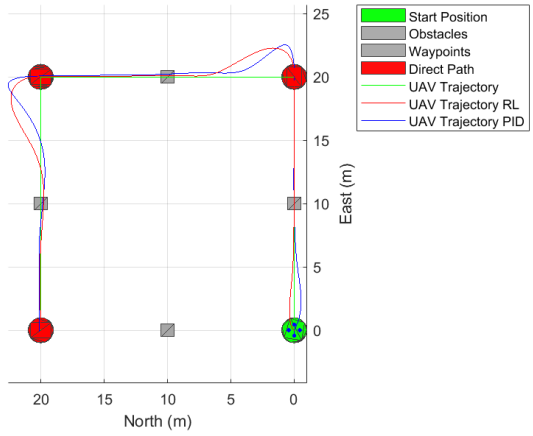

Fig. 4. UAV Trajectories in Scenario 1

As shown in Figure 4, the RL controller exhibits superior wind-rejection characteristics. At the first waypoint $(0,\ 20)$, the PID controller suffered a maximum spatial overshoot of 2.518 m, whereas the RL agent constrained this deviation to 2.261 m, representing a 10.2% reduction in peak error. While the PID system demonstrates significant lateral drift and 'bowing' effects on the straight legs due to the linear lag in error correction, the RL controller effectively learns a compensatory 'crabbing' behavior. This proactive adjustment maintains a tighter alignment with the reference path, achieving a negligible steady-state cross-track error compared to the $0.5\text{m} - 0.8\text{m}$ offset observed in the PID response.

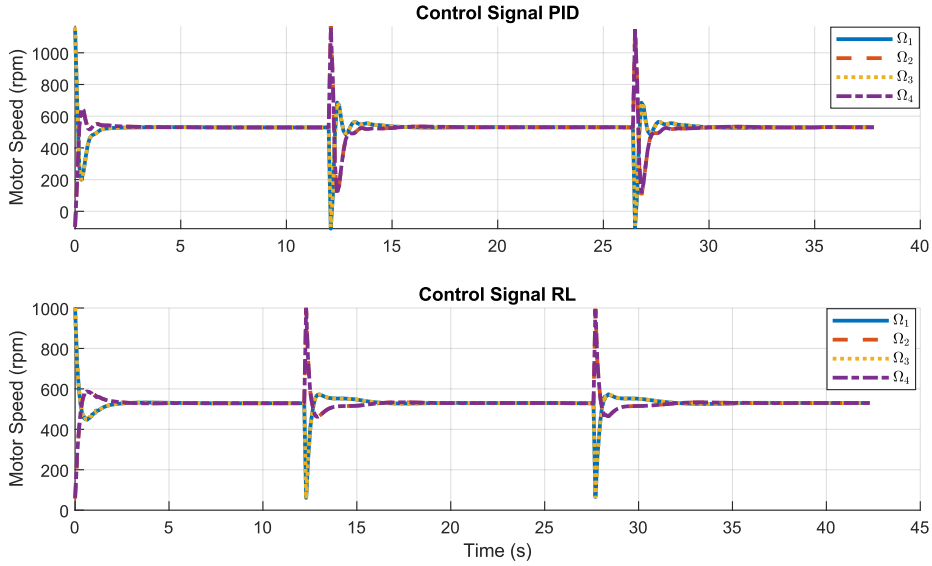

Fig. 5. UAV control signals in Scenario 1

A comparative analysis of the motor speed in Figure 5 profiles reveals that while both controllers converge to a nominal hovering speed of approximately 535rpm, the RL controller demonstrates significantly superior transient stability and damping characteristics compared to the PID baseline. Specifically, during critical state transitions at $t \approx 12\text{s}$ and $t \approx 27\text{s}$, the PID controller exhibits severe underdamped oscillations with drastic speed fluctuations ranging from 0 to over 1100rpm, whereas the RL controller executes smoother, more monotonic transitions with minimal overshoot. Furthermore, the RL agent achieves a reduction in settling time of approximately 50%, effectively eliminating the persistent chattering observed in the PID response; this indicates a more optimized control policy that minimizes mechanical stress on the motor actuators while providing the rapid, decisive thrust modulation required to maintain trajectory fidelity under high-wind conditions.

4.2. Case 2: Environment with obstabcles

The second scenario introduces static obstacles to test the maneuverability and safety margins of the UAV. The integration of high-velocity wind vectors and obstacle avoidance maneuvers reveals a fundamental performance disparity

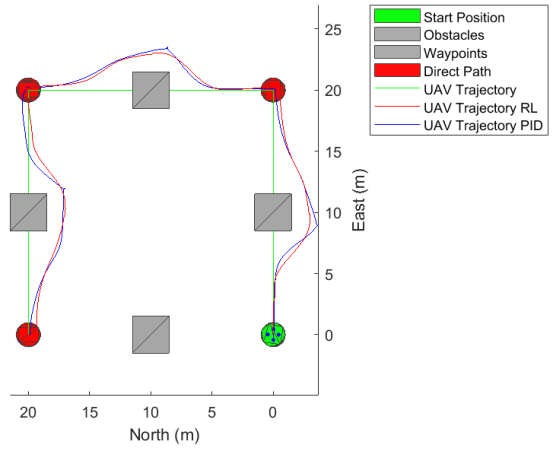

Fig. 6. UAV Trajectories in Scenario 2

In Figure 6, the RL controller demonstrates superior performance through its advanced capacity for non-linear optimization and dynamic adaptation, effectively surpassing the linear constraints of the PID baseline. Unlike the PID controller, which exhibits passive, underdamped responses characterized by wide arcing and significant lateral drift, the RL-based approach facilitates agile dynamic maneuvering with decisive thrust allocation, allowing the UAV to execute tight turns while maintaining rigorous stability. Furthermore, the RL agent's superior disturbance rejection and rapid convergence rate enable a significantly faster recovery to the reference trajectory following avoidance maneuvers, highlighting a fundamental shift from the reactive nature of PID to a proactive, optimized control architecture that thrives in high-uncertainty environments.

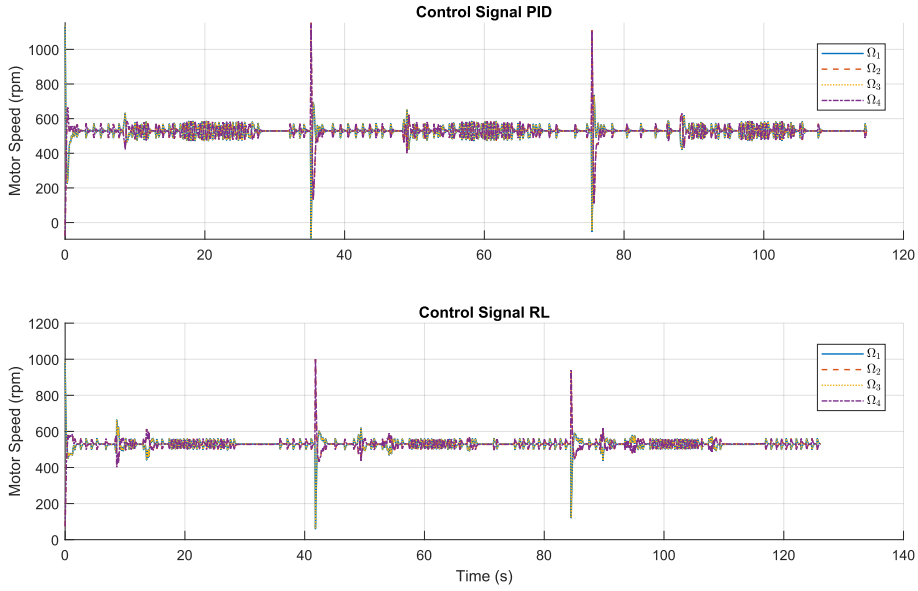

Fig. 7. UAV control signals in Scenario 2

Analysis of the motor speed data in Figure 7 under 10 m/s wind conditions reveals that the RL controller significantly outperforms the PID baseline in terms of stability and efficiency. Unlike the PID controller, which exhibits severe chattering and spikes above 1,149 rpm, the RL agent produces a smoother output with peaks limited to 1,000 rpm, eliminating the aggressive switching behavior. These findings underscore the RL controller's ability to provide robust disturbance rejection while promoting energy efficiency and mechanical durability.

5. CONCLUSION

In conclusion, this research successfully implemented an adaptive PID control framework tuned by a TD3 agent, proving its superior robustness over fixed-gain PID in high-windspeed and obstacle-cluttered environments. By dynamically optimizing the $K_p, K_i,$ and K_d gains in real-time, the TD3 agent effectively compensated for the non-linear aerodynamic disturbances and inertial effects, resulting in more reduction in peak spatial overshoot and a more decrease in settling time. The simulation results confirm that this hybrid approach maintains the inherent stability of PID control while leveraging the deep learning capability of TD3 to achieve smoother motor speed modulation and higher trajectory fidelity under a 10 m/s wind load. Future work will focus on evaluating the real-time computational efficiency of the TD3 algorithm for hardware-in-the-loop testing and investigating the generalization of the gain-tuning policy across varying UAV payloads and diverse atmospheric turbulence models.

REFERENCES

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace sciences,* vol. 91, pp. 99-131, 2017, doi: https://doi.org/10.1016/j.paerosci.2017.04.003.

[2] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intelligent service robotics,* vol. 16, no. 1, pp. 109-137, 2023, doi: https://doi.org/10.1007/s11370-022-00452-4.

[3] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005: IEEE, pp. 2247-2252, doi: https://doi.org/10.1109/ROBOT.2005.1570447.

[4] V. Hoang, M. D. Phung, and Q. P. Ha, "Adaptive twisting sliding mode control for quadrotor unmanned aerial vehicles," in *2017 11th Asian control conference (ASCC)*, 2017: IEEE, pp. 671-676, doi: https://doi.org/10.1109/ASCC.2017.8287250.

[5] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory & Applications,* vol. 6, no. 12, pp. 1812-1827, 2012, doi: https://doi.org/10.1049/iet-cta.2011.0348.

[6] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter," *Automatica,* vol. 46, no. 1, pp. 29-39, 2010, doi: https://doi.org/10.1016/j.automatica.2009.10.018.

[7] B. Han, Y. Zhou, K. K. Deveerasetty, and C. Hu, "A review of control algorithms for quadrotor," in *2018 IEEE international conference on information and automation (ICIA)*, 2018: IEEE, pp. 951-956, doi: https://doi.org/10.1109/ICInfA.2018.8812437.

[8] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ international conference on intelligent robots and systems*, 2007: IEEE, pp. 153-158, doi: https://doi.org/10.1109/IROS.2007.4399042.

[9] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and control of mini-flying machines*. Springer, 2005, doi: https://doi.org/10.1007/1-84628-179-2.

[10] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE robotics & automation magazine,* vol. 19, no. 3, pp. 20-32, 2012, doi: https://doi.org/10.1109/MRA.2012.2206474.

[11] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Backstepping/nonlinear H∞ control for path tracking of a quadrotor unmanned aerial vehicle," in *2008 American Control Conference*, 2008: IEEE, pp. 3356-3361, doi: https://doi.org/10.1109/ACC.2008.4587010.

[12] A. G. Barto, "Reinforcement learning: An introduction. by richard's sutton," *SIAM Rev,* vol. 6, no. 2, p. 423, 2021, doi: https://doi.org/10.1016/S0893-6080(99)00098-2.

[13] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017, doi: https://doi.org/10.1201/9781439821091.

[14] G. Bujgoi and D. Sendrescu, "Tuning of PID controllers using reinforcement learning for nonlinear system control," *Processes,* vol. 13, no. 3, p. 735, 2025, doi: https://doi.org/10.3390/pr13030735.

[15] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012, doi: https://doi.org/10.2514/1.61067.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*, 2011: IEEE, pp. 2520-2525, doi: https://doi.org/10.1109/ICRA.2011.5980409.

[17] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo,* vol. 22, no. 22, pp. 1-24, 2011.

[18] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems,* vol. 12, 1999, doi: https://dl.acm.org/doi/10.5555/3009657.3009806.

[19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning,* vol. 8, no. 3, pp. 229-256, 1992, doi: https://doi.org/10.1007/BF00992696.

[20] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems,* vol. 12, 1999, doi: https://doi.org/10.1137/S0363012901385691.

[21] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015, doi: https://doi.org/10.1038/nature14236.

[22] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971,* 2015, doi: https://doi.org/10.48550/arXiv.1509.02971.

[23] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015: PMLR, pp. 1889-1897, doi: https://doi.org/10.48550/arXiv.1502.05477.

[24] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, 2018: PMLR, pp. 1587-1596, doi: https://doi.org/10.48550/arXiv.1802.09477.