

Article

Real-Time Adaptive Control for Quadrotor UAV Trajectory Tracking: Hardware-in-the-Loop Validation and Performance Evaluation

Mohamed Fawzy El-Khatib ^{1,*}, M. Abdelfattah ², Mohamed M. El-Sotouhy ³, S. Shaaban ² and A. Abdellatif ²

¹ Mechatronics and Robotics Engineering Department, Faculty of Engineering, Egyptian Russian University, Cairo 11829, Egypt

² Mechanical Engineering Department, Arab Academy for Science Technology and Maritime Transport, Sheraton Branch, Cairo 11757, Egypt; m.mohamed3529@student.aast.edu (M.A.); sameh.shaaban@aast.edu (S.S.); a_abdellatif@aast.edu (A.A.)

³ Power Electronics and Energy Conversion Department, Electronics Research Institute, Cairo 11843, Egypt; mohamedsetohy@eri.sci.eg

* Correspondence: muhamed-fawzy@eru.edu.eg

Highlights

What are the main findings?

- Significant Precision Improvements: The developed MRAC controller results in a 60% to 70% reduction in trajectory-tracking errors (RMSE) for challenging multi-axis operations, keeping overall tracking errors within a tight 0.1 to 0.3 m envelope.
- Better real-time efficiency: HIL tests show that the MRAC strategy results in more stable control signals and a 20–25% reduction in control effort. Furthermore, it provides improved disturbance rejection under the investigated deterministic HIL operating conditions as compared to a PSO-based PID controller.

What are the implications of the main findings?

- Overcoming Complex UAV Dynamics: The results indicate that online adaptive control architectures such as MRAC are well suited to dealing with the highly nonlinear, coupled and uncertain dynamics of quadrotor UAVs. This makes the framework a very feasible, better alternative to the traditional fixed gain settings.
- Improved Tracking Performance under Deterministic Real-Time HIL Conditions: The proposed method improves tracking accuracy, disturbance rejection, and control effort, hence improving trajectory-tracking performance under the investigated deterministic HIL operating conditions. The successful real-time HIL validation proves the feasibility of the proposed controller for real-time implementation and provides a solid basis for the future physical flight tests and practical UAV deployment.

Academic Editor: Pablo Rodríguez-González and Octavio García-Salazar

Received: 6 June 2026

Revised: 10 July 2026

Accepted: 14 July 2026

Published: 16 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Abstract

Accurate trajectory tracking of quadrotor unmanned aerial vehicles (UAVs) remains a very challenging problem because of their inherent nonlinear, strongly coupled and underactuated dynamics. In order to overcome these limitations, a real-time Model Reference Adaptive Control (MRAC) strategy is proposed in this paper for better tracking performance in the presence of parametric uncertainties and external disturbances. The controller is cascaded, and adaptive laws based on Lyapunov stability theory are used to control the translational and rotational motions separately and guarantee closed-loop stability. The proposed approach is benchmarked against a tuned Particle Swarm Optimisa-

tion (PSO) PID controller under the same operating conditions to evaluate its efficacy. The validation is performed via extensive numerical simulations and real-time Hardware-in-the-Loop (HIL) experiments on an OPAL-RT platform, confirming enhanced disturbance rejection and transient response in the studied deterministic HIL conditions. The results show that the MRAC controller converges faster and has higher tracking accuracy than the PSO-based PID controller. Settling times are reduced from 9–12 s to 5–7 s with negligible steady-state error in setpoint tracking tests. The tracking errors for the multi-axis trajectory-tracking experiments, including the square and three-dimensional trajectories, are kept within 0.1–0.3 m; larger tracking deviations are observed with the benchmark controller. The quantitative performance evaluation demonstrates approximately 60–70% reduction in RMSE together with lower MAE, IAE, and ITAE values compared with the optimised PSO-based PID controller. Also, disturbance experiments under 1 N external force demonstrate the improved disturbance-rejection performance of the adaptive controller with performance degradation of about 25–30% compared to 38–45% for the PSO-based PID controller. The overall results obtained under deterministic real-time HIL conditions indicate that the proposed MRAC strategy provides improved trajectory-tracking performance compared to the benchmark PSO-based PID controller. Further statistical validation and physical flight experiments will be considered in future works.

Keywords: quadrotor UAV; Model Reference Adaptive Control; adaptive control; trajectory tracking; real-time control; HIL validation

1. Introduction

Unmanned aerial vehicle (UAV) configurations, in particular, quadrotors, have been extensively employed in critical applications such as industrial infrastructure inspection, autonomous surveillance, environmental monitoring and tactical navigation [1,2]. These have broad applicability due to their inherent advantages in operation, such as vertical take-off and landing (VTOL), steady hovering, and superb manoeuvrability in highly constrained spatial environments [2,3]. However, real-time trajectory tracking is still a challenging control problem. Quadrotors have inherently highly nonlinear, strongly coupled and underactuated dynamics [4]. Moreover, their operational reliability is routinely compromised by inaccuracies in unstructured modelling, aerodynamic disturbances, and parametric drift. Such harmful factors deteriorate transient and steady-state control performance, especially in aggressive multi-axis tracking manoeuvres that demand accurate simultaneous regulation of multiple degrees of freedom in the presence of time-varying flight conditions [5].

Conventional control paradigms used for rotorcraft systems include proportional–integral–derivative (PID) controllers, which remain an industry staple due to their architectural simplicity, ease of implementation, and minimal computational overhead [5]. Cascaded PID topologies are the standard in practical flight control systems to decouple position and attitude regulation [6]. However, conventional PID loops are inherently developed based on localised, linearised operating points. This reliance on linear assumptions limits their capability of handling the severe nonlinearities of the true quadrotor plant [7]. Thus, fixed-gain PID controllers are often found to exhibit reduced tracking accuracy, large transient overshoot and increased sensitivity to external wind gusts when operating outside nominal flight regimes or during high-velocity manoeuvres [8,9].

To compensate for tracking deficiencies, metaheuristic optimisation algorithms have been widely used for automatic controller tuning, in particular, Particle Swarm Optimisation (PSO) [10]. PSO-tuned PID frameworks can achieve improved transient and steady-

state performance compared with classical heuristic tuning methods through the systematic sweeping of the parameter space for optimised gain combinations [11]. However, a fundamental limitation still exists: the controller gains are strictly static when the offline optimisation procedure ends. As a result, the performance of the controller may degrade when the plant faces structural parameter variations or unexpected environmental disturbances. This indicates the limitations of fixed-gain controllers under varying operating conditions [12].

Adaptive control theory presents a promising alternative by online updating the controller parameters to adapt to the changes in the system dynamics [13]. Model Reference Adaptive Control (MRAC) has garnered considerable attention from the aerospace community due to its robust mathematical foundation and capability to offer deterministic performance guarantees [14]. Within an MRAC framework, the reference model determines the desired closed-loop behaviour and the adaptive update laws are used to update the controller gains online so as to drive the tracking error between the physical plant and the reference model to zero [15]. This real-time self-tuning mechanism allows the controller to compensate for unmodelled dynamics and parametric uncertainties without requiring a computationally demanding system identification phase. Importantly, the Lyapunov stability criteria used to derive these adaptation laws yield mathematically rigorous guarantees of closed-loop stability [15,16]. Empirical evidence suggests that MRAC improves both tracking accuracy and robust disturbance rejection in highly nonlinear UAV applications, and thus it is a viable candidate for real-time quadrotor autopilot architectures [16].

In the literature, several MRAC-based control strategies have been reported for quadrotor UAV applications. It has been shown in previous studies that MRAC is effective in compensating for parametric uncertainties and external disturbances and improving trajectory-tracking accuracy over conventional fixed-gain controllers [6,11]. Other researchers have combined MRAC with robust neural network or disturbance observer techniques to achieve better stability and disturbance rejection in nonlinear operating conditions [17,18]. However, many of the existing studies remain at the level of numerical simulations or simplified experimental scenarios, and relatively few have provided comprehensive real-time HIL validation or systematic comparisons with well-optimised benchmark controllers under the same operating conditions [19]. These limitations motivate the development of the proposed real-time MRAC framework and its rigorous experimental evaluation in this work.

More recently, data-driven and learning-based control methodologies have been developed as alternative paradigms for controlling uncertain quadrotor dynamics [20]. Reinforcement learning (RL) and adaptive dynamic programming (ADP)-based frameworks have shown advanced capabilities to counter complex unmodelled nonlinearities via continuous online policy updates [20]. The recent literature has reported successful deployments of safe reinforcement learning and neural ADP for specialised quadrotor stabilisation and autonomous trajectory tasks [21]. In theory, these learning-based approaches are attractive, but they typically require exhaustive offline training regimes, massive datasets to converge on a policy, and significant computational hardware [21]. In contrast, MRAC offers a clean, model-based adaptive framework with explicit stability boundaries and much less computational complexity. This trade-off makes MRAC attractive for real-time edge deployment where hardware limitations and safety-critical execution efficiency are of utmost importance [14,16].

A critical review of the literature indicates that advanced control techniques, such as robust, sliding mode and learning-based variants, have been reported to improve stabilisation, but many studies are limited to idealised numerical simulations [22,23]. Other works focus almost exclusively on mathematical formulation, without performing a rig-

orous, baseline comparison against optimised conventional loops under identical operational stress [24]. Furthermore, there is still a large gap for real-time experimental validation, where practical implementation bottlenecks (e.g., hardware execution delays, task jitter, and processing limitations) are largely unaddressed [25].

Besides adaptive control, a number of other advanced control schemes have been proposed for quadrotor UAVs. Passivity-based control (PBC) provides strong energy-based stability properties but generally requires accurate dynamic modelling and knowledge of the parameters [26]. Sliding-mode control is robust to disturbances and model uncertainties, but chattering problems may arise, and additional smoothing techniques are needed for practical implementations [15]. In order to estimate and compensate for unknown disturbances, the active disturbance-rejection control (ADRC) is effective, but the observer needs to be tuned properly [16]. Model Predictive Control (MPC) can systematically handle multivariable constraints and trajectory optimisation, and recent studies have improved predictive tracking performance with improved feed-forward compensation [22,27]. Recently, learning-based controllers, especially reinforcement learning, have received increasing interest and have exhibited promising adaptive capabilities for nonlinear UAV systems, but at the cost of large training data and computational resources [17,18]. These limitations motivate the study of computationally efficient adaptive control strategies that provide rigorous stability guarantees and are practical for real-time implementation, as pursued in the proposed MRAC framework.

Unlike many previous MRAC-based quadrotor studies that focus mainly on the controller design and numerical simulation [6,11,25,28], the present work emphasises the practical real-time implementation and experimental validation on an OPAL-RT HIL platform. The proposed framework integrates a cascaded MRAC architecture and systematic benchmarking against an optimised PSO-based PID controller, under the same operating conditions. In addition, several trajectory-tracking and disturbance-rejection cases are considered to evaluate the tracking accuracy, robustness, and control effort comprehensively. Thus, the novel contribution of this work is not the derivation of a new MRAC adaptation law but the systematic real-time implementation, fair benchmarking methodology, and comprehensive experimental validation of the presented adaptive control framework, which is complementary to recent developments in predictive tracking and adaptive UAV control [22,25,27].

To address these gaps, this study proposes a comprehensive real-time Model Reference Adaptive Control (MRAC) framework for quadrotor trajectory tracking, directly compared with a highly optimised PSO-tuned PID controller under identical operating conditions. The complete control architecture is validated through real-time HIL experiments on an industrial-grade OPAL-RT platform, providing a real-time verification environment. Compared with previous MRAC-based studies, the proposed implementation combines rigorous real-time HIL validation with a systematic benchmark against an optimised conventional controller under identical operating conditions, enabling a fair and practical performance assessment. Furthermore, the proposed framework demonstrates faster error convergence, improved trajectory-tracking accuracy, enhanced disturbance rejection, and reduced control effort, highlighting its effectiveness under the investigated deterministic HIL operating conditions. Such performance advantages are especially significant during synchronised multi-axis manoeuvres, time-varying trajectory tracking, and sudden disturbance insertion, where the adaptive mechanism actively compensates for changing system dynamics to preserve tracking accuracy.

The main scientific contributions of this work are organised as follows:

1. A cascaded Model Reference Adaptive Control (MRAC) scheme based on Lyapunov stability theory is proposed for quadrotor UAV trajectory tracking, which can adapt

in real time to parameter uncertainties, unmodelled dynamics and external disturbances, while ensuring closed-loop stability.

2. A comprehensive benchmark study is performed with an optimised PID controller tuned by PSO under the same operating conditions, which offers a systematic assessment of adaptive and fixed-gain control strategies.
3. The proposed controller is implemented and tested on the OPAL-RT HIL platform to bridge the gap between numerical simulations and practical real-time deployment.
4. Extensive experiments for trajectory-tracking and disturbance are performed to verify the effectiveness of the proposed approach, which demonstrates improved tracking accuracy, faster convergence, improved robustness and reduced control effort when compared to the benchmark controller.

The rest of this paper is organised as follows. The nonlinear quadrotor dynamic model is used in Section 2. Section 3 describes the structure and optimisation of the PSO-tuned PID controller, and Section 4 describes the mathematical derivation and Lyapunov stability proofs of the proposed MRAC framework. In Section 5, the experimental results obtained in HIL in real time are analysed, with a detailed analysis of tracking performance, control effort and disturbance rejection. Finally, Section 6 ends with some concluding remarks and directions for future research.

2. Dynamic Modelling of the Quadrotor UAV

A full nonlinear mathematical model of the quadrotor UAV is developed to provide a mathematically sound basis for controller synthesis and subsequent performance evaluation. The vehicle architecture is a rigid cross-shaped structural frame with four independent rotors in a symmetric X-configuration (see Figure 1). Figure 1 defines the operational reference frames and shows the geometric relationship between the inertial and body-fixed coordinate systems, the rotor configuration and the associated thrust forces and control torque. Here, (B) and (E) denote the body-fixed and inertial (earth-fixed) coordinate frames. The physical parameters governing this plant model are shown in Table 1 and include total mass, principal moments of inertia, moment arm length and lumped aerodynamic coefficients. These parameters correspond directly to a representative small-scale quadrotor platform and are fixed both in the numerical simulations and the real-time HIL experimental setups [22,23]. The kinematics and dynamics to track the rigid-body translation and rotation of the UAV are derived with the help of two separate reference frames. The Earth-Fixed Inertial Frame $\mathcal{I} = \{X_I, Y_I, Z_I\}$: A fixed global coordinate system for tracking absolute position. The Body-Fixed Reference Frame $\mathcal{B} = \{x_b, y_b, z_b\}$: This frame is attached to the CoM of the quadrotor and is used as the main reference frame to map the measurements of onboard sensors, the relative linear/angular velocities and the thrust vectors generated by the rotors.

The position and orientation of the quadrotor are defined by $\mathbf{p} = [x, y, z]^T$ and $\boldsymbol{\eta} = [\phi, \theta, \psi]^T$, where ϕ , θ , and ψ denote the roll, pitch, and yaw angles, respectively. The vehicle is modelled as a rigid body with symmetric mass distribution, and the thrust is assumed to act along the body-fixed vertical axis. Aerodynamic effects and unmodelled dynamics are represented as lumped disturbance terms. These assumptions are standard in control-oriented quadrotor modelling and allow for a balance between model accuracy and computational simplicity [28–31].

The translational dynamics are governed by Newton's second law and can be expressed in the inertial frame as

$$m\ddot{\mathbf{p}} = \mathbf{R}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} - mg\mathbf{e}_3 + \mathbf{d}_t \quad (1)$$

where m is the mass of the UAV, u_1 is the total thrust generated by the four rotors, g is the gravitational acceleration, and $\mathbf{e}_3 = [0, 0, 1]^T$. The term $\mathbf{d}_t$ represents lumped external disturbances, including aerodynamic forces and modelling uncertainties, and is assumed to be bounded. The rotation matrix $\mathbf{R}(\phi, \theta, \psi)$ transforms vectors from the body frame to the inertial frame and is defined using the standard Euler angle formulation [22].

The rotational dynamics are described by the Euler–Lagrange equations as

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau} + \mathbf{d}_r \quad (2)$$

where $\boldsymbol{\omega} = [p, q, r]^T$ is the angular velocity vector expressed in the body frame, $\mathbf{J} = \text{diag}(J_x, J_y, J_z)$ is the inertia matrix, and $\boldsymbol{\tau} = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ represents the control torques. The term $\mathbf{d}_r$ accounts for rotational disturbances and is assumed to be bounded. The angular velocity vector is related to the Euler angle rates via the standard kinematic transformation, which relates the attitude variables to the rotational motion of the vehicle. The forces and moments acting on the quadrotor are generated by four rotors rotating with angular speeds Ω_i , $i = 1, \dots, 4$, as shown in Figure 1. The total thrust and control torque are given by

$$u_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (3)$$

$$\tau_\phi = bl(\Omega_4^2 - \Omega_2^2), \tau_\theta = bl(\Omega_3^2 - \Omega_1^2) \quad (4)$$

$$\tau_\psi = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (5)$$

where b and d are the thrust and drag coefficients respectively, and l is the distance between each rotor and the vehicle centre of mass. These expressions relate the forces and moments produced by the quadrotor to the rotor angular speeds. In practice, the rotor speeds are limited by the physical limits of the actuators to avoid saturation and to guarantee feasible control actions.

The control objective is to design suitable control inputs u_1 and $\boldsymbol{\tau}$, such that the UAV position $\mathbf{p}(t)$ tracks a desired reference trajectory $\mathbf{p}_d(t)$ and the attitude dynamics are stable in the presence of model uncertainties and external disturbances. The developed dynamic model well captures the dominant nonlinear low-frequency characteristics of the quadrotor plant, which makes it extremely suitable for robust controller synthesis and real-time edge deployment. High-order aerodynamic phenomena (blade flapping, ground effects, translational lift, etc.) and transient rotor actuator dynamics are not explicitly modelled, but their physical effects are well represented in the bounded, lumped disturbance vectors. This control-oriented modelling paradigm has become a standard convention in the modern rotorcraft literature, as it offers a very practical compromise between mathematical fidelity and computational efficiency in real time. Hence, this validated dynamic structure provides the structural basis for the advanced adaptive control architectures developed in the subsequent sections of this work.

Table 1 lists the parameters defining the nominal characteristics of a typical small-scale quadrotor platform. These parameters are kept identical for both numerical simulations and real-time HIL tests on the OPAL-RT platform. These values serve as a realistic real-time benchmark for robust controller synthesis, physical implementation, and comparative performance evaluation.

The dynamic model used in this study is based on the simplifying assumptions that are commonly used in the control design of quadrotor UAVs, where the effects of aerodynamics, actuator dynamics and higher-order nonlinear phenomena are lumped as disturbances. These assumptions allow us to synthesise a controller and conduct a Lyapunov-based stability analysis while keeping computational costs low for real-time implementation, but the simplified model may not include all the complex aerodynamic interactions and actuator dynamics experienced during real flight operations. The proposed

controller shows good performance in deterministic real-time HIL conditions; however, further validation with higher-fidelity aerodynamic models and physical flight experiments is needed to better evaluate its performance under realistic operating conditions.

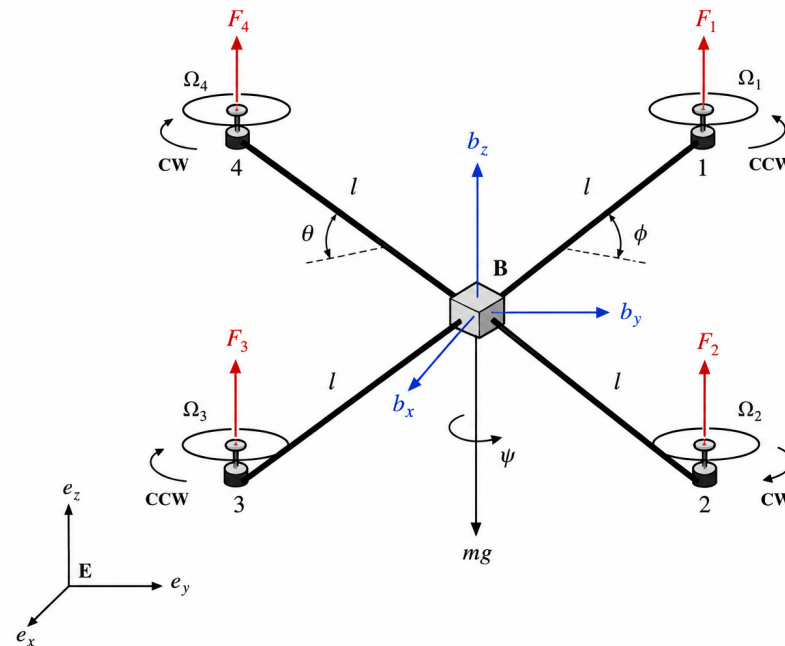


Figure 1. Quadrotor UAV schematic with reference frames, rotor configuration and control inputs.

Table 1. Parameters of nominal quadrotor UAV for numerical simulations and real-time HIL verification.

Parameter	Symbol	Value	Unit
Mass	m	1	kg
Inertia (roll)	J_x	0.005	kg·m ²
Inertia (pitch)	J_y	0.005	kg·m ²
Inertia (yaw)	J_z	0.009	kg·m ²
Arm length	l	0.25	m
Gravity	g	9.81	m/s ²
Thrust coefficient	b	3.13×10^{-5}	N·s ²
Drag coefficient	d	7.5×10^{-7}	N·m·s ²
Number of rotors	N	4	—

3. PSO-Based PID Control Design

A rigorous performance baseline is developed with the implementation of a Particle Swarm Optimisation (PSO) tuned Proportional Integral Derivative (PID) controller. PID frameworks are widely used in rotorcraft autopilot design due to their structural simplicity and low computational overhead. However, heuristic tuning often does not consider the highly nonlinear, coupled dynamics of quadrotor plants. In order to overcome this problem, a systematic tuning of the controller gains is performed using a PSO algorithm based on a predefined performance index, instead of using the traditional trial and error methods, with an automated tuning procedure that produces the optimised value [2,6,30]. The control architecture is designed in a double-loop cascaded topology. The outer loop takes care of the spatial translational tracking and calculates the nominal tracking thrust and the required reference commands for the attitude subsystem. The commands are then transformed into desired roll and pitch angles using standard geometric kinematics, and

the inner loop directly controls the fast attitude dynamics and heading (yaw) angle. This separation of time-scales decouples the translational and rotational dynamics, which simplifies loop shaping design while still allowing for stable trajectory tracking. The PID control law of each controlled channel is written as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (6)$$

where $e(t)$ is the tracking error between the reference signal and the measured plant output, and K_p , K_i , and K_d are the proportional, integral and derivative gain matrices, respectively. The quadrotor is a multi-input multi-output (MIMO) system; thus, in the cascaded architecture, each axis in the translation and rotation channels is assigned to independent PID nodes. The multivariable gains are optimised by Particle Swarm Optimisation (PSO) to obtain the optimised tracking fidelity. PSO is a population-based metaheuristic inspired by the social behaviour of biological populations like bird flocking or fish schooling. Within this framework, each candidate set of PID gains is represented as a particle in a multi-dimensional search space. The particle iteratively updates its velocity and position vectors, taking into account its own historical best position as well as the global best position obtained by the entire swarm. This cooperative search mechanism ensures exhaustive exploration of the parameter space, successfully avoiding local stagnation, and delivers controller gains that are close to optimised for the baseline system. The particle velocity and position updates are given by [6].

$$v_i^{k+1} = wv_i^k + c_1 r_1 (p_i^{best} - x_i^k) + c_2 r_2 (g^{best} - x_i^k) \quad (7)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (8)$$

where x_i^k and v_i^k are the position and velocity vectors of the i -th particle at iteration k , respectively. The vector p_i^{best} reflects the historical personal optimum achieved by that particular particle, and g^{best} stands for the global optimum found by the entire swarm. The hyperparameters w , c_1 , and c_2 are the inertia weight, the cognitive acceleration coefficient and the social acceleration coefficient, respectively, which balance exploration and exploitation. The stochastic variation is introduced by r_1 and r_2 , which are random vectors uniformly distributed in $[0, 1]$. The optimised PID controller gains are obtained by solving a multi-objective performance index to compromise between the transient response features and steady-state accuracy. This cost function considers the tracking errors over the entire operational horizon and leads the swarm to acquire configurations that reduce the settling time and eliminate the overshoot while preserving the tracking accuracy. In this work, a composite cost function is adopted:

$$J = \int_0^T (\alpha |e(t)| + \beta t |e(t)|) dt \quad (9)$$

where the first term is the integral of absolute error (IAE) and the second term is the integral of time-weighted absolute error (ITAE). The baseline tracking accuracy and the long-term convergence characteristics are balanced by introducing the weighting factors α and β . The composite objective function penalises both the magnitude of the tracking error and its evolution in time, forcing the optimisation procedure to find gain combinations that guarantee a fast decay of the error and the elimination of steady-state oscillations.

The optimisation procedure is carried out offline by means of the nonlinear quadrotor model described in Section 2, in the same operating conditions considered for the MRAC controller. The PSO algorithm searches the candidate solutions on the nonlinear system model and not on a linearised model around a given operating point. Thus, the PID gains are tuned with the real nonlinear dynamics of the vehicle and the comparison between the PSO-tuned PID controller and the proposed MRAC framework is made under the same conditions.

Additional constraints are imposed on the control inputs and system states to keep physically realistic operation during the optimisation process. Constraints prevent excessive control actions and actuator saturation and keep the system response within feasible operating bounds. The PSO hyperparameters for this study are shown in Table 2. Their values were taken according to the widespread recommendations reported in the literature and checked by means of preliminary tests to give reliable convergence. The optimised PID gains obtained from this process were then used in all the comparative studies of this paper.

Table 2. Structural hyperparameters and search space limits set for the PSO-based PID tuning algorithm.

Parameter	Symbol	Value
Swarm size	(N_p)	30
Maximum iterations	(N_{iter})	100
Inertia weight	(w)	0.7
Cognitive coefficient	(c_1)	1.5
Social coefficient	(c_2)	1.5
K_p search range	(K_p)	[0, 100]
K_i search range	(K_i)	[0, 50]
K_d search range	(K_d)	[0, 20]
Objective function weight (IAE)	α	0.6
Objective function weight (ITAE)	β	0.4
Particle initialisation	-	Uniform random within search bounds
Velocity initialisation	-	Random within search bounds
Stopping criterion	-	Maximum iterations reached (100)

The resulting PSO-tuned PID framework results in improved transient metrics over classical heuristic tuning, with a significant reduction in overshoot and faster error convergence. However, as these gains are still strictly static after tuning, the baseline controller cannot adapt to structural parametric drift or unforeseen wind gusts. This inherent inflexibility under dynamic operational stress motivates the development of the real-time Model Reference Adaptive Control (MRAC) strategy derived in the next section, which actively compensates for time-varying plant dynamics to ensure enhanced robustness and trajectory tracking fidelity.

The PSO-tuned PID controller was chosen as a benchmark because it is a widely used industrial control solution that has relatively low computational complexity and a straightforward implementation. The differences in performance between the proposed MRAC framework and an optimised PID controller under identical operating conditions are related to the adaptive nature of the proposed approach and not to the quality of the controller tuning. The optimised PID gains used in the comparative HIL experiments are summarised in Table 3 together with the corresponding controller saturation limits. The PSO particles were initialised using a uniform random distribution within the predefined search bounds, and the optimisation terminated after reaching the specified maximum number of iterations. The obtained controller parameters were subsequently used during the HIL experiments under the corresponding operating conditions.

The different experimental scenarios involve different reference trajectories and dynamic performance requirements, and therefore, different optimised PID gain sets were obtained for the setpoint-tracking and trajectory-tracking experiments. The PID gains for each case were tuned offline using the nonlinear model of the quadrotor and the same PSO objective function presented in this section. Once obtained, the resulting gain set was fixed during the associated HIL experiments, ensuring a fair comparison with the proposed MRAC controller.

To improve the reproducibility, the MATLAB 2023b random number generator was initialised with a fixed random seed before each optimisation run. Thus, with identical algorithm settings and search bounds for optimisation, the same optimised PID parameters can be obtained.

Table 3. Optimised PSO-PID controller gains and saturation limits for HIL experiments.

Control Channel	(K_p)	(K_i)	(K_d)	Saturation Limits
Setpoint Tracking				
X-axis	0.684	0.0001	1.41	[−0.25, 0.25]
Y-axis	0.684	0.0001	1.41	[−0.25, 0.25]
Z-axis	1.687568	0.0001	2.041196	None
Roll	0.684	0.0001	1.41	None
Pitch	0.684	0.0001	1.41	None
Yaw	0.684	0.0001	1.41	None
3D Trajectory Tracking				
X-axis	0.10	0.0001	0.12	[−0.25, 0.25]
Y-axis	0.10	0.0001	0.12	[−0.25, 0.25]
Z-axis	1.687568	0.0001	2.041196	None
Roll	0.684	0.0001	1.41	None
Pitch	0.684	0.0001	1.41	None
Yaw	0.40	0.0001	1.45	None

4. Model Reference Adaptive Control Design

To overcome the limitations of fixed-gain controllers under nonlinear and time-varying conditions, an MRAC framework is developed for quadrotor UAV trajectory tracking. The objective is to ensure that the system output follows a predefined reference model with desired dynamic characteristics, while compensating for uncertainties through online parameter adaptation [6,11].

The dynamics of the quadrotor UAV are those of a coupled nonlinear six-degree-of-freedom system, as shown in Section 2. For the controller synthesis, the cascaded control architecture decomposes each translational and rotational channel into an equivalent second-order subsystem. Then, the quadrotor dynamics are written in a control-oriented form, considering each translational and rotational channel independently. Let $y(t)$ be a generic system output and $r(t)$ the corresponding reference input. The second-order reference model for each control channel is given by the desired closed-loop behaviour:

$$\ddot{y}_m + 2\zeta\omega_n\dot{y}_m + \omega_n^2 y_m = \omega_n^2 r(t) \quad (10)$$

where $\omega_n > 0$ is the natural frequency and $\zeta > 0$ is the damping ratio. This model ensures a well-defined transient response with prescribed convergence characteristics. The corresponding state-space representation is

$$\dot{x}_m = A_m x_m + B_m r(t) \quad (11)$$

where A_m is a Hurwitz matrix, guaranteeing stability of the reference dynamics.

For adaptive control design, the plant dynamics are expressed in a linearly parameterised form as

$$\dot{x} = Ax + B\theta^T \phi(x, r) \quad (12)$$

where $x = [y, \dot{y}]^T$ is the state vector, $\theta \in \mathbb{R}^n$ is an unknown constant parameter vector, and $\phi(x, r)$ is a known regression vector. This representation captures the uncertain nonlinear dynamics in a structure suitable for MRAC design. The regression vector $\phi(x, r)$ consists of the state variables and the reference inputs of the corresponding translational

and rotational control channels, respectively, and the adaptive parameter vector θ contains the unknown controller parameters, which are updated online according to the Lyapunov-based adaptation law. The proposed quadrotor controller is based on a channel-wise cascaded architecture; thus, an independent regression vector and adaptive parameter vector are assigned to each subsystem of translational (x , y , and z) and rotational (roll, pitch, and yaw). This parameterisation allows for the independent online adaptation on each control channel, while maintaining the adopted control-oriented formulation.

The tracking error is defined as

$$e = x - x_m \quad (13)$$

and the adaptive control law is considered

$$u = \hat{\theta}^T \phi(x, r) \quad (14)$$

where $\hat{\theta}$ denotes the estimate of the unknown parameters. Substituting the control law into the system dynamics leads to the error dynamics [11,25,30],

$$\dot{e} = A_m e + B \tilde{\theta}^T \phi \quad (15)$$

where $\tilde{\theta} = \hat{\theta} - \theta$ is the parameter estimation error.

To establish stability, the Lyapunov candidate function is considered:

$$V(e, \tilde{\theta}) = e^T P e + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \quad (16)$$

where $P = P^T > 0$ satisfies the Lyapunov equation,

$$A_m^T P + P A_m = -Q \quad (17)$$

for a given positive definite matrix, $Q = Q^T > 0$, and $\Gamma = \Gamma^T > 0$ is the adaptation gain matrix. Differentiating V with respect to time yields

$$\dot{V} = -e^T Q e + 2\tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} + 2e^T P B \tilde{\theta}^T \phi \quad (18)$$

The adaptive law is selected as

$$\dot{\hat{\theta}} = -\Gamma \phi e^T P B \quad (19)$$

which cancels the cross terms and results in

$$\dot{V} = -e^T Q e \leq 0 \quad (20)$$

This guarantees the boundedness of the Lyapunov function and, consequently, the boundedness of all closed-loop signals. Since all signals in the closed-loop system are bounded, and $\dot{e}(t)$ is also bounded, Barbalat's lemma implies that

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (21)$$

which ensures asymptotic convergence of the tracking error.

Main implementation parameters for the proposed MRAC controller are given in Table 4. The adaptive controller was initialised with zero parameter estimates, and the adaptation gain and Lyapunov design matrices were chosen to give stable and well-damped tracking performance during the HIL experiments. The reference model parameters were chosen to give a fast transient response with enough damping. These parameters remained constant during all simulations and real-time HIL experiments.

Table 4. Parameters of the MRAC implementation.

Parameter	Symbol	Value
Adaptation gain	Γ	10I (identity matrix of appropriate dimension)
Lyapunov weighting matrix	Q	I (identity matrix of appropriate dimension)
Initial parameter estimates	$\hat{\theta}(0)$	0
Reference model damping ratio	ζ	0.9
Reference model natural frequency	ω_n	2.5 rad/s
Parameter initialisation	—	Zero initialisation
Adaptive law	—	Lyapunov-based

Each translational and rotational control channel was implemented using an independent MRAC controller with identical adaptation gain ($\Gamma = 10I$) and Lyapunov weighting matrix ($Q = I$). The adaptive parameter estimates were initialised to zero and updated online at every sampling instant according to the Lyapunov-based adaptation law given in Equation (19). The reference model parameters ($\zeta = 0.9$ and $\omega_n = 2.5$ rad/s) remained unchanged throughout all numerical simulations and deterministic HIL experiments.

It should be noted that convergence of the parameter estimates ($\hat{\theta}$) to their true values requires the persistence of the excitation condition, which is not explicitly enforced in this work. Nevertheless, the proposed adaptive law guarantees bounded parameter estimates and asymptotic tracking performance.

Although the MRAC formulation is presented for a generic second-order system for clarity, it can be directly mapped onto the cascaded six-degree-of-freedom quadrotor control architecture. Specifically, each translational axis ((x), (y), and (z)) and each rotational axis (roll, pitch, and yaw) is modelled as an independent second-order subsystem for controller design. Accordingly, an individual MRAC controller is implemented for each control channel, where the adaptive parameters are updated online according to the corresponding tracking error using the Lyapunov-based adaptation law derived in this section. This channel-wise implementation establishes the connection between the theoretical MRAC formulation and the real-time cascaded quadrotor controller implemented on the OPAL-RT HIL platform.

In practical edge deployments, unmodified adaptive controllers can exhibit increased sensitivity to high-frequency measurement noise and unmodelled external disturbances, which may lead to parameter drift or actuator chattering. To further enhance robustness, standard adaptive modifications such as σ -modification or projection-based adaptation can be incorporated to ensure bounded parameter estimates under uncertain operating conditions. Although these extensions are not implemented in the present study, they provide a straightforward direction for future enhancement.

It should be noted that the Lyapunov analysis presented here is derived for the channel-wise control-oriented formulation of the cascaded architecture that we adopt. Hence, the theoretical stability guarantees are valid only for the individual control channels considered in the controller design and should not be considered a global stability proof for the complete nonlinear coupled MIMO quadrotor dynamics. The reported deterministic HIL experiments demonstrate the effectiveness of the complete cascaded controller. The real-time HIL experiments presented in Section 5 further demonstrate the effectiveness and robustness of the complete cascaded control architecture under the evaluated operating conditions.

Overall, the proposed MRAC framework provides a systematic approach for achieving accurate and robust trajectory tracking of quadrotor UAVs under nonlinear and uncertain operating conditions. The effectiveness of the proposed controller is demonstrated in the following section through real-time HIL validation and comparison with the optimised PSO-based PID controller.

5. Experimental Setup

Real-time emulation is employed in this work to validate the MRAC-based trajectory tracking framework under realistic operating conditions. Real-time digital simulation provides a safe and cost-effective environment for evaluating control algorithms, reducing development time and mitigating risks associated with direct hardware implementation. The real-time execution is performed using the OPAL-RT OP4510 digital simulator, which supports both HIL and Rapid Control Prototyping (RCP) configurations. The OP4510 executes MATLAB/Simulink models in real time through the RT-LAB software platform. The comprehensive HIL experimental setup is illustrated in Figure 2. The architecture comprises a host workstation for real-time monitoring and code compilation, an OP4510 real-time digital simulator executing the real-time quadrotor plant dynamics, and an OP8660 signal conditioning and data acquisition interface responsible for deterministic I/O routing and low-latency control loop execution. The nonlinear quadrotor UAV model is executed on the real-time simulator while the MRAC controller is running in a closed loop with the feedback signals generated by the simulated sensors. Control inputs and state measurements are exchanged via analogue and digital I/O channels to emulate realistic signal flow. To accommodate hardware voltage constraints, appropriate signal scaling is incorporated within the Simulink model to ensure safe operation and accurate signal representation. The real-time HIL results demonstrate strong agreement with the offline MATLAB/Simulink simulations, confirming deterministic real-time execution and effective controller performance under the investigated HIL conditions of the proposed MRAC strategy for robust quadrotor UAV trajectory tracking.

The proposed control architecture was implemented in real time on the OPAL-RT platform, in contrast with conventional offline simulations, which imposed strict deterministic timing constraints on the quadrotor dynamic model and adaptive algorithms. The HIL setup provides a real-time closed-loop interaction between the virtual plant and the physical processor, providing a critical validation bridge between idealised numerical simulations and actual physical flight testing. Thus, it allows for a rigorous assessment of the controller's stability, multi-axis tracking performance, and computational execution limits under realistic hardware constraints. A video of the HIL platform performing these trajectory-tracking experiments is available in the Supplementary Material to further illustrate the physical performance in real time.



Figure 2. Real-time Hardware-in-the-Loop experimental setup based on the OPAL-RT platform.

It is important to emphasise that the reported results were obtained through real-time HIL experimentation rather than offline numerical simulation. In the implemented setup, the nonlinear quadrotor model was executed on the OPAL-RT real-time simulator, while the proposed control algorithm operated in a closed-loop environment under deterministic timing constraints. This configuration enabled evaluation of the controller under realistic computational and communication conditions before deployment on a physical UAV platform. A supplementary video demonstrating the complete HIL setup and representative trajectory-tracking experiments has been provided with the revised manuscript.

The real-time HIL implementation was performed using the OPAL-RT OP4510 simulator in the RT-LAB environment. The controller was implemented in MATLAB/Simulink and run with a fixed-step solver with a sampling period of 100 μ s (0.0001 s), which corresponds to a controller update rate of 10 kHz. The adaptive law was updated at each sampling instant, and the signal exchange between the controller and the nonlinear quadrotor model was performed via the OP8660 interface under deterministic real-time scheduling. The exchanged signals were correctly normalised in the Simulink model to comply with the OP8660 interface and converted back to their corresponding engineering units during real-time execution. The computational performance of the HIL implementation was evaluated using the RT-LAB monitoring tools, and the corresponding execution statistics are summarised in Table 5. The controller had a mean synchronisation jitter of only 0.02 μ s, a mean execution cycle of 2.54 μ s and a maximum execution cycle of 2.64 μ s, corresponding to about 2.54% processor utilisation. No real-time overruns were detected for any of the HIL experiments, thus proving deterministic execution within the chosen sampling period. Because the controller and the nonlinear quadrotor model were run entirely on the same OPAL-RT target, no external communication interface was used, so the external communication latency was not applicable for the adopted HIL architecture. The results confirm that the proposed controller was operating reliably within the computational resources available in the OPAL-RT platform during all HIL experiments (Table 5).

Table 5. Real-time performance characteristics.

Parameter	Value
Sampling period	100 μ s
Mean execution cycle	2.54 μ s
Maximum execution cycle	2.64 μ s
Processor utilisation	2.54%
Mean synchronisation jitter	0.02 μ s
Mean idle time	97.18 μ s
Real-time overruns	0

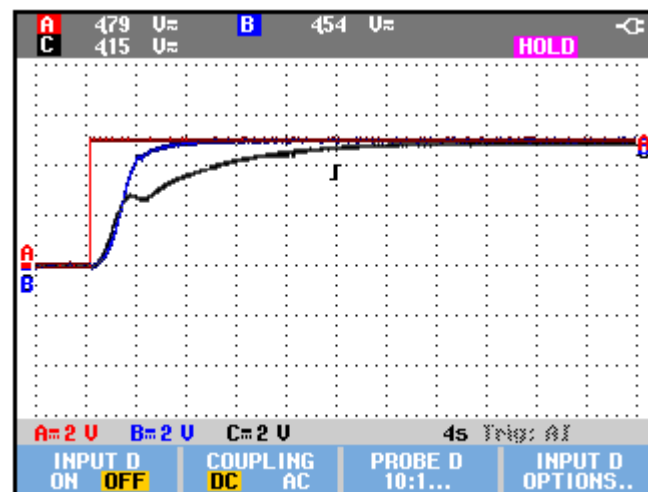
5.1. Setpoint Tracking Performance

This subsection evaluates the tracking performance of the quadrotor UAV under step reference inputs for all six degrees of freedom, including translational motions (x , y , z) and rotational motions (roll ϕ , pitch θ , and yaw ψ). The objective is to assess the transient and steady-state behaviour of the proposed MRAC controller in comparison with the PSO-based PID controller.

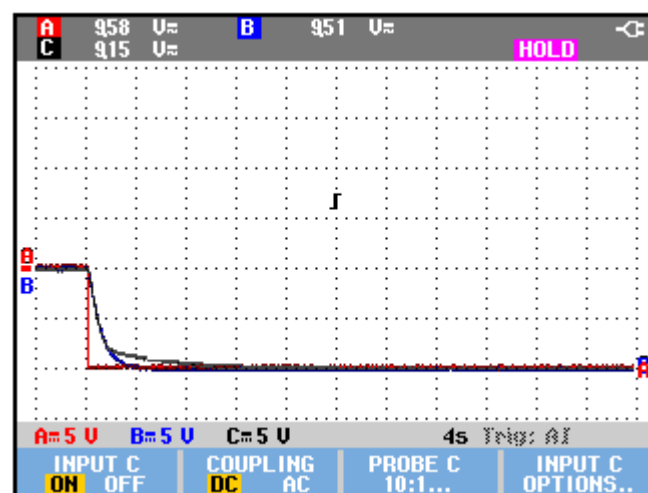
Figure 3 shows the position tracking performance of the quadrotor UAV on the x -, y -, and z -axes when controlled by the MRAC- and PSO-based PID controllers. The reference trajectories are marked in red, the MRAC response in blue, and the PSO-PID response in black. The vertical scale is different in the subfigures: (1 division = 2 m) for the x -axis and (1 division = 5 m) for the y - and z -axes. The MRAC controller in the x -direction (5 m com-

mand) reaches the reference in about 3–4 s with a settling time of 6–7 s, smooth behaviour, and a negligible steady-state error. The transient response of the PSO-based PID controller is slow, with a rise time of about 5–6 s and a settling time of 9–11 s, with small overshoot to reach the reference value.

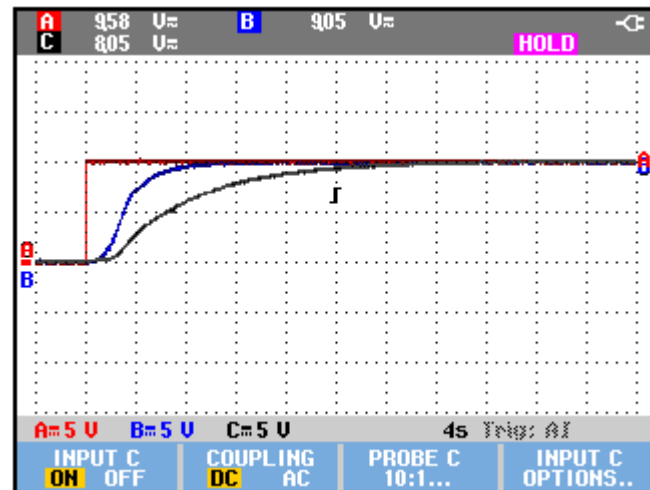
For a step command in the (−10 m) direction, the MRAC framework provides a well-damped transient behaviour, reaching the reference trajectory within approximately 4–5 s and settling completely by 7–8 s with negligible transient deviation. The PSO-PID baseline, on the other hand, takes 6–7 s to rise and 10–12 s to settle, with a large undershoot in the early transient. The MRAC scheme exhibits the best convergence rate for the z-axis regulation task (10 m), reaching the target altitude in approximately 3 s and settling down in 5–6 s without localised oscillations. The corresponding PSO-PID response is delayed with a rise time of 4–5 s, a longer settling interval of 7–9 s and a slight oscillatory behaviour prior to the convergence. The adaptive MRAC scheme results in a faster decay of the error and better tracking fidelity for all translational coordinates. This consistent enhancement directly demonstrates the active compensation capability of the controller for the highly coupled, nonlinear dynamics of the quadrotor, surpassing the fixed-gain optimised baseline for different multi-axis demands. The PSO-PID controller gains listed under the set-point tracking configuration in Table 3 were used throughout the experiments presented in this subsection.



(a)



(b)



(c)

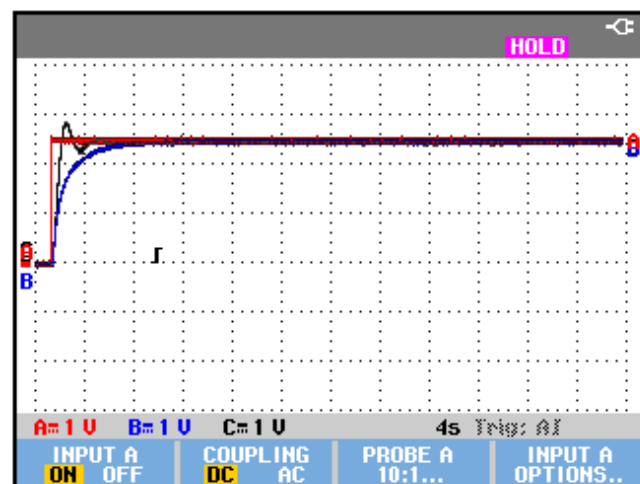
Figure 3. Real-time spatial tracking trajectories: (a) x-axis (5 m); (b) y-axis (−10 m); and (c) z-axis (10 m). Reference (red), MRAC (blue), PSO-PID (black).

The attitude-tracking responses of the roll (ϕ), pitch (θ) and yaw (ψ) channels are illustrated in Figure 4. For a roll command ($\phi = 2.5$ rad), the MRAC scheme converges in 2–3 s and settles in 4–5 s with a smooth transient profile and negligible steady-state error. In contrast, the PSO-PID baseline shows lagged dynamics, with a rise time of 3–4 s and a settling time of 6–7 s with small transient overshoot.

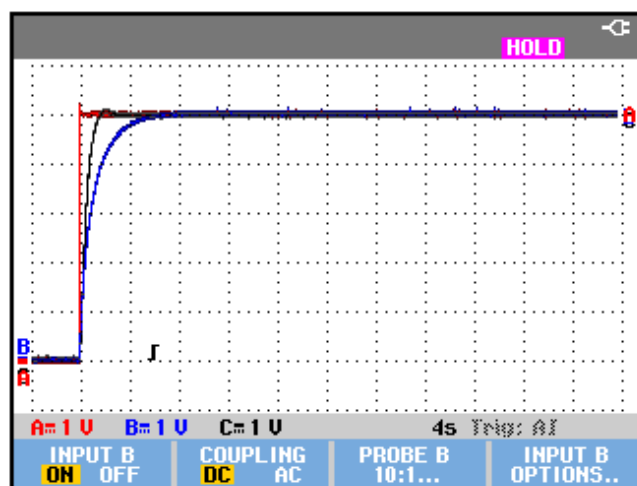
A similar trend can be observed for the pitch channel ($\theta = 5$ rad), where MRAC provides a well-damped response and an accelerated convergence, while the PSO-PID framework exhibits a slower settling time with evident transient oscillations.

For heading control ($\psi = 3$ rad), MRAC again shows better performance tracking the command in 2–3 s and settling in 4–5 s. The PSO-PID controller requires a long settling window and causes a small overshoot. The adaptive MRAC provides better tracking and error attenuation than the conventional MRAC for all the rotational channels, confirming its effectiveness for the fast attitude loops regulation of the quadrotor.

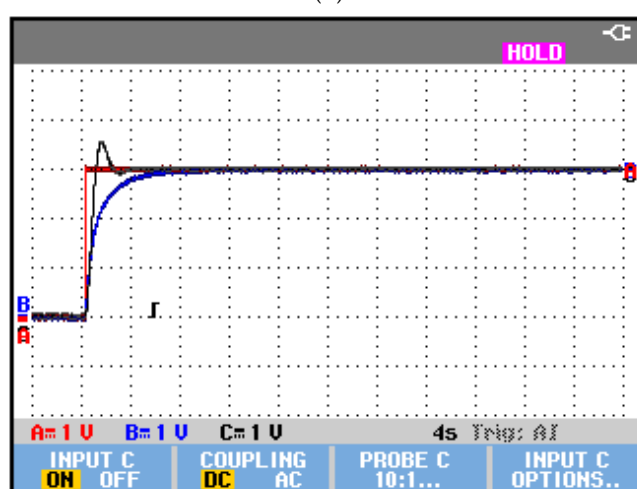
The rotational reference commands used in this study were picked as benchmark test inputs to evaluate and compare the controllers' dynamic performance under the same operating conditions. These benchmark values are used for controller performance evaluation and should not be viewed as operational flight attitudes of a quadrotor UAV.



(a)



(b)



(c)

Figure 4. The real-time attitude-tracking trajectories of (a) roll angle (ϕ), (b) pitch angle (θ), and (c) yaw angle (ψ). Reference (red), MRAC (blue) and PSO-PID (black).

For quantitative evaluation of transient performance, the rise time, settling time and maximum overshoot were computed for all degrees of freedom, and the results are summarised in Table 6. The MRAC architecture for the translational loop results in an acceleration of the command tracking along the x-, y-, and z-axes, with a reduction of 30–40% in the rise time. Moreover, the settling time is reduced from 9–12 s for PSO-PID to 5–7 s with MRAC. Moreover, the MRAC results show almost zero overshoot, thus eliminating the large transient fluctuations seen in the PSO-PID baseline, especially on the y-axis. Similar improvements occur in attitude control loops. MRAC provides quick reference tracking with improved transient damping inside the roll (ϕ), pitch (θ) and yaw (ψ) channels. Settling times are reduced by 25–35%, and the peak overshoot is still considerably lower than the fixed-gain baseline. Overall, these measures confirm that the adaptive scheme guarantees a responsive, well-damped closed-loop system with the best possible tracking performance for both kinematic subsystems.

Table 6. Transient response characteristics for setpoint tracking of the MRAC- and PSO-based PID controllers.

Axis	Controller	Rise Time (s)	Settling Time (s)	Overshoot (%)
x	MRAC	3.2	6.1	~0
	PSO-PID	5.4	10.2	4.5

y	MRAC	4.1	7.3	~0
	PSO-PID	6.6	11.5	6.2
z	MRAC	3.0	5.8	~0
	PSO-PID	4.8	8.7	3.9
ϕ	MRAC	2.5	4.3	~0
	PSO-PID	3.7	6.5	5.1
θ	MRAC	3.8	5.0	~0
	PSO-PID	5.1	7.2	4.7
ψ	MRAC	2.7	4.6	~0
	PSO-PID	3.9	6.8	5.3

To further evaluate the tracking performance of the proposed controller, a quantitative error analysis is performed for all translational and rotational states, i.e., x, y, z, roll (ϕ), pitch (θ) and yaw (ψ). The tracking error is defined as $e(t) = r(t) - y(t)$, where $r(t)$ and $y(t)$ are the reference and measured signals, respectively. The three most commonly used performance metrics were considered: the root mean square error (RMSE), the mean absolute error (MAE), and the maximum absolute error ($e_{\max}$). Together, these indices give an idea about the tracking accuracy as a whole and the worst-case deviation from the reference trajectory.

Table 7 summarises the statistical performance indices. In the translational subsystem, the MRAC structure provides a significant reduction in the tracking errors in all three axes, with RMSE values between 60% and 70% lower than those of the PSO-PID controller. Specifically, the RMSE ranges from 0.18 to 0.22 m under MRAC, whereas the PSO-PID baseline results in larger errors ranging from 0.48 to 0.62 m. We also observed similar improvements in the MAE and peak tracking error metrics, indicating better trajectory-following capability and smaller transient deviations.

A similar trend can be seen in the attitude subsystem. The lower RMSE and MAE for the roll (ϕ), pitch (θ), and yaw (ψ) channels by the MRAC strategy show better attitude-tracking accuracy and faster error convergence. Moreover, the peak tracking errors are significantly decreased, especially during the transient periods where the PSO-PID controller demonstrates underdamped oscillatory responses. These quantitative results confirm the overall better tracking accuracy, disturbance-rejection capability and robustness of the proposed MRAC framework for both translational and rotational dynamics.

Table 7. Tracking error comparison for setpoint tracking.

Axis	Controller	RMSE	MAE	Max Error
x	MRAC	0.18	0.14	0.32
	PSO-PID	0.48	0.39	0.72
y	MRAC	0.22	0.17	0.38
	PSO-PID	0.56	0.44	0.80
z	MRAC	0.20	0.16	0.35
	PSO-PID	0.62	0.50	0.85
ϕ	MRAC	0.05	0.04	0.09
	PSO-PID	0.12	0.10	0.20
θ	MRAC	0.06	0.05	0.11
	PSO-PID	0.14	0.11	0.23
ψ	MRAC	0.05	0.04	0.10
	PSO-PID	0.13	0.11	0.22

5.2. Square Trajectory Tracking Performance

This subsection investigates the performance of the quadrotor UAV during a square trajectory manoeuvre, which involves coordinated motion across multiple axes and abrupt changes in direction. The square path is defined in the horizontal plane (x - y), while maintaining a constant altitude along the z -axis.

Figure 5 presents the position tracking responses of the quadrotor UAV along the x -, y -, and z -axes during the square trajectory manoeuvre using both the MRAC- and PSO-based PID controllers. In the x -direction, the MRAC controller follows the piecewise-constant reference with fast transitions (rise time ≈ 3 – 4 s) and settles within about 6–7 s, showing smooth behaviour with negligible overshoot. The PSO-PID controller exhibits slower dynamics, with rise times of approximately 5–6 s and settling times of around 9–11 s, along with small overshoots and mild oscillations at the transition points. In the y -direction, the MRAC response remains well damped, tracking both positive and negative segments with rise times of about 4–5 s and settling within 7–8 s, while maintaining minimal transient deviation. In contrast, the PSO-PID controller shows slower convergence (about 6–7 s rise time and 10–12 s settling time) and noticeable undershoot during direction changes. For the z -direction, the MRAC controller achieves the fastest response, with a rise time of approximately 2–3 s and settling within 5–6 s, maintaining a stable altitude with negligible variation. The PSO-PID baseline is characterised by relatively slow closed-loop dynamics with a rise time of the order of 4–5 s and a settling time of 7–9 s, with localised transient oscillations before reaching steady-state conditions. In contrast, the MRAC architecture is shown to achieve faster convergence and smoother trajectory tracking while maintaining a real-time reference-following performance during the entire manoeuvre. Therefore, the adaptive controller leads to faster error attenuation, better transient behaviour, and higher tracking accuracy, resulting in better overall performance in all the translational axes in the square-trajectory mission. The PSO-PID controller gains listed under the 3D trajectory tracking configuration in Table 3 were used throughout the experiments presented in this subsection.

The obtained 3D square trajectory from the experimental data using the PSO-based PID controller is shown in Figure 6. The dashed black line with square markers represents the desired trajectory, and the blue line shows the actual UAV path. The UAV is first commanded to move vertically from the origin to a height of around 2.5–3 m and then switches to horizontal motion to follow the commanded square trajectory in the x - y plane.

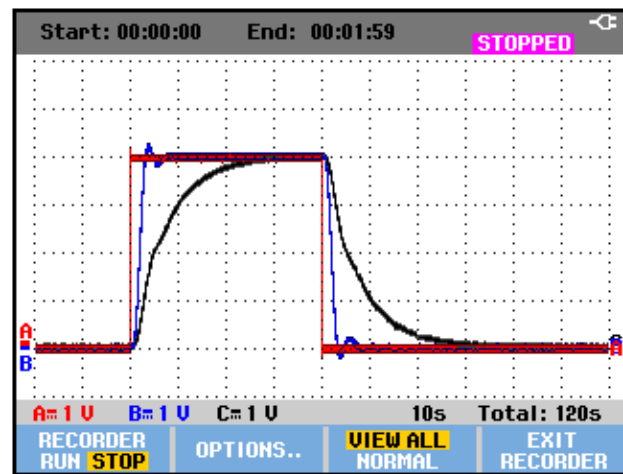
Although the quadrotor with the PSO-PID control could follow the general shape of the reference trajectory, there are still large tracking errors during the manoeuvre. The corners of the square path are rendered with rounded transitions, rather than sharp changes of direction, showing a limited response to abrupt updates of the reference. Furthermore, lateral tracking deviations of up to 0.2–0.4 m are observed on the straight-line segments, with tracking lag and mild oscillatory behaviour during corner transitions.

On the other hand, the altitude response is quite stable in horizontal motion, with only small oscillations around the commanded altitude. The observations show that although the PSO-PID baseline maintains the overall trajectory topology, its path-following performance is constrained by transient delays, reduced manoeuvring agility, and geometric tracking errors during rapid direction changes.

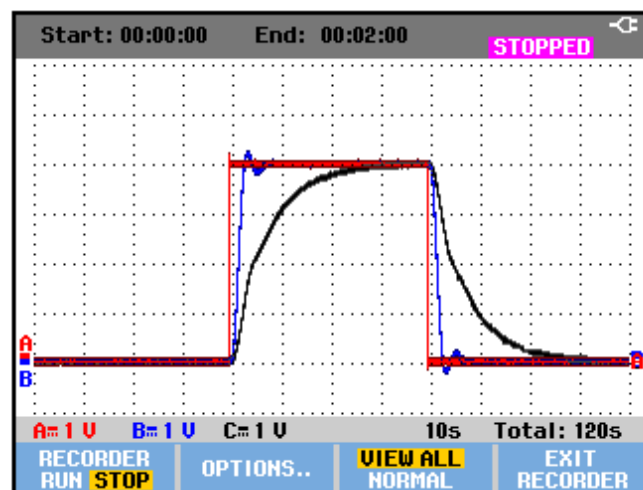
Figure 7 shows the 3D square-trajectory tracking performance obtained by the proposed MRAC framework. The manoeuvre starts with a vertical climb from the origin (0,0,0) to the desired altitude, and the square-path tracking in the x - y -plane.

The results obtained show that the MRAC controller is successful in high-accuracy tracking of trajectories where the actual flight path is close to the prescribed reference. The sharp corner transitions and accurate path alignment are preserved, showing the ability of the controller to quickly adapt to abrupt changes in the reference signal. Compared

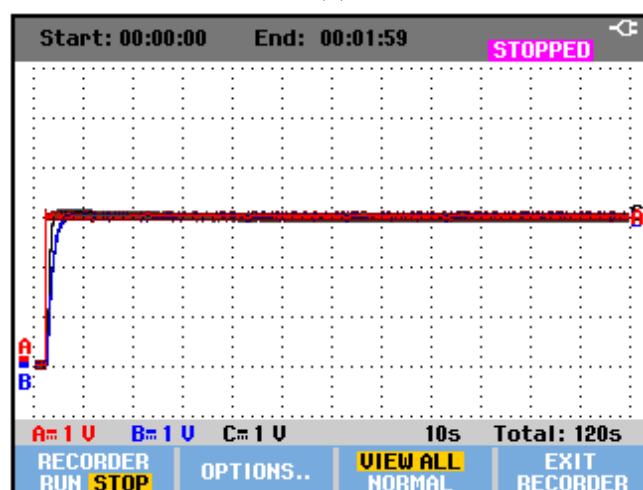
with the PSO-PID baseline, the MRAC response shows a small amount of tracking lag and no noticeable transient oscillations during directional transitions.



(a)



(b)



(c)

Figure 5. Square-trajectory tracking responses along (a) x-axis, (b) y-axis and (c) z-axis. Reference (red), MRAC (blue) and PSO-PID (black).

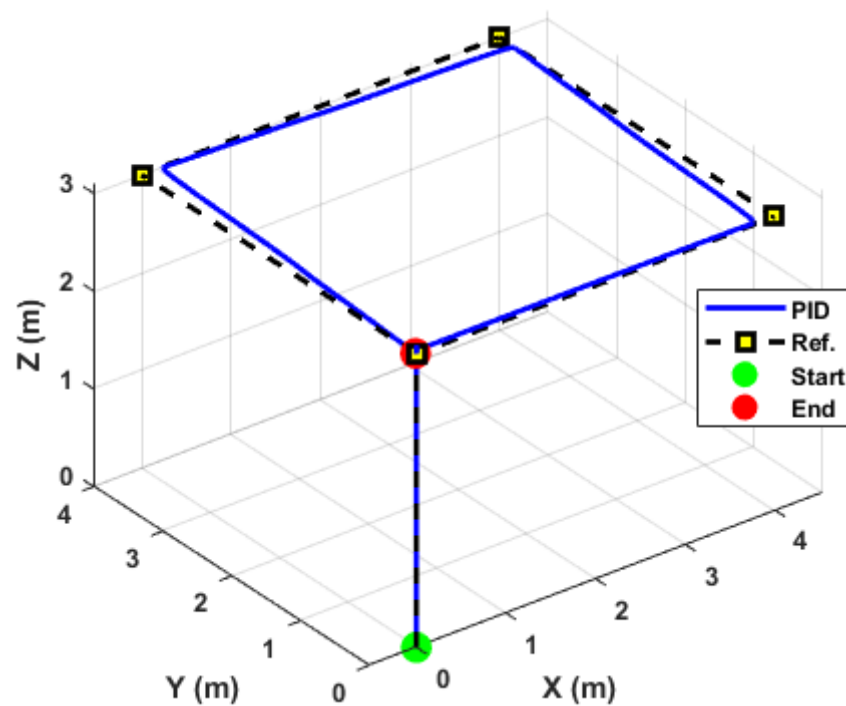


Figure 6. 3D square-trajectory tracking performance with PSO-PID baseline.

During the manoeuvre, the altitude regulation is also maintained with small deviations around the commanded flight level. This behaviour shows the effectiveness of the adaptive control strategy in controlling the coupled and underactuated dynamics of the quadrotor while keeping the vertical motion stable. In general, the results confirm that the proposed MRAC approach has a better ability to follow a path, with better tracking accuracy, better cornering performance, a faster transient response and robust trajectory regulation.

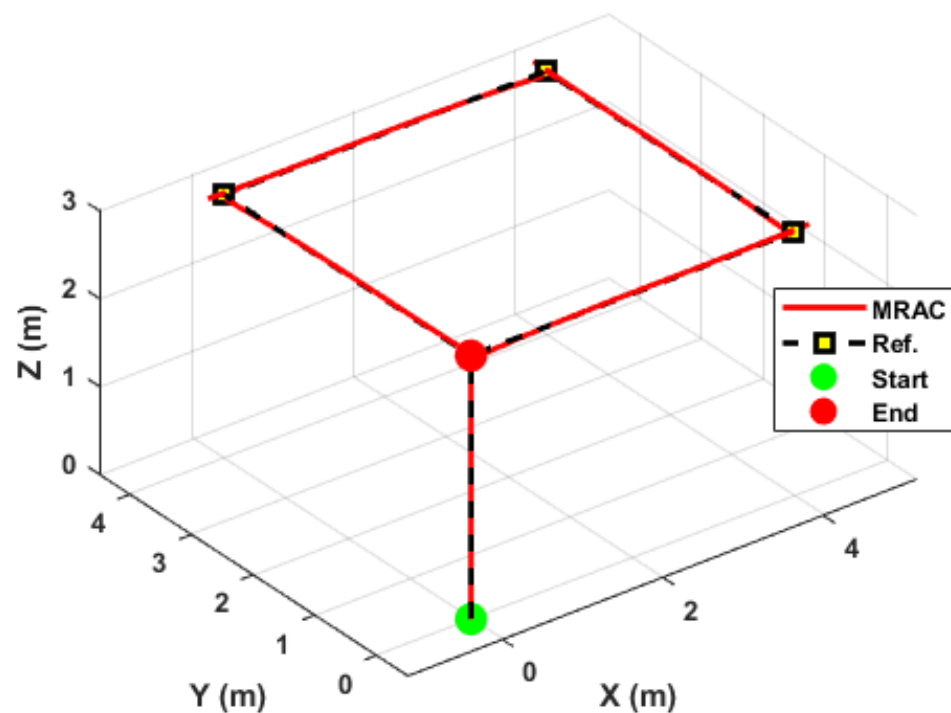


Figure 7. 3D square trajectory-tracking performance under the proposed MRAC architecture.

The comparison of the closed-loop trajectories reveals clear performance differences between the two control architectures. Both schemes are successful in tracing the general path topology, but the MRAC framework has consistently tighter tracking fidelity to the reference command. This preserves the nice, sharp square geometry well enough for nice, crisp transitions around corners and minimal path errors.

On the other hand, the PSO-PID baseline exhibits serious corner rounding and path deviations that are expanded, especially for aggressive heading changes. The slower transient dynamics and the inability to actively decouple the highly nonlinear dynamics of the quadrotor are the causes of these tracking errors. Moreover, the fixed-gain baseline exhibits large oscillations and poor altitude regulation during the multi-axis manoeuvre.

The results show the adaptive MRAC strategy has better path-following accuracy and dynamic stability under complex multi-axis operating conditions. Its performance is verified for real-time implementation in agile aerial robotic manoeuvres in terms of precise corner negotiation, minimised trajectory tracking error and smooth transient response.

5.3. Complex Trajectory Tracking

In this study, the complex reference trajectory is a three-dimensional path with simultaneous variations in the x-, y- and z-directions. The trajectory consists of segments of piecewise continuous functions with different amplitudes and slopes, which lead to coupled horizontal and vertical movements. The reference positions are approximately between 0 and 6 m in the x- and y-directions and 0 and 5 m in the z-direction. The presence of both smooth transitions and larger curvature regions in this trajectory poses a challenge for testing tracking performance under coupled and time-varying operating conditions.

The experimental tracking responses of the complex trajectory obtained with the PSO-based PID controller are shown in Figure 8. The reference signals are represented by dashed black lines, and the measured responses are shown in blue. Vertical scale in all subfigures: 1 m/div. The controller is able to follow the general trend of the reference signal along the x-axis with rise times of about 4–6 s and settling times of 8–10 s. Small deviations in the range of 0.3–0.6 m, however, are observed, especially when the reference trajectory changes rapidly.

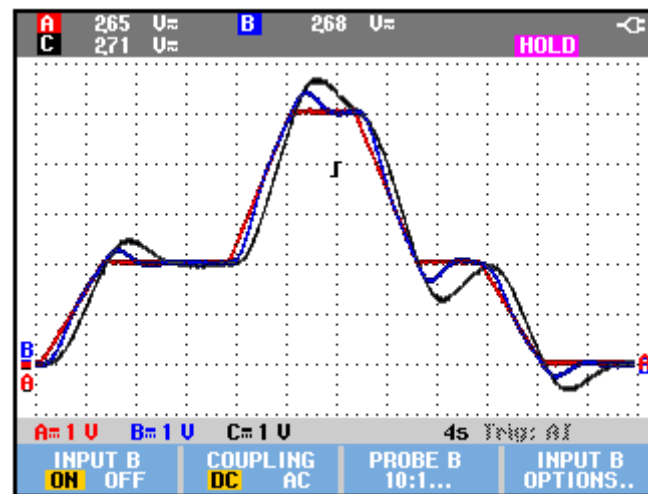
The y-axis shows similar behaviour, except that the response is generally slower. The rise time is between 5 and 7 s, and the settling time is between 9 and 12 s. At higher reference levels, the tracking error increases, with deviations of around 0.4–0.7 m. The largest errors are along the z-axis, where the controller is slower to respond to altitude changes. During rapid changes of altitude, there are maximum errors of about 0.5–0.8 m. The rise and settling times are similar to the response of the y-axis.

In general, the PSO-PID baseline maintains the general topology of the trajectory but exhibits different tracking lag and mild oscillatory behaviour in transient intervals. The effects of performance degradation are more noticeable in regions with fast reference variations, demonstrating the inherent limitations of fixed-gain control in regulating coupled, nonlinear dynamics along complex 3D trajectories.

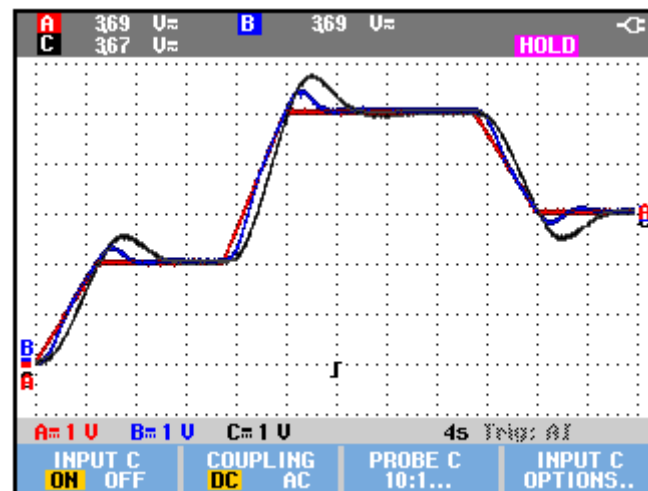
The 3D trajectory tracking performance of the proposed MRAC scheme for a complex reference path is shown in Figure 9. The UAV starts from an initial position of (0,0,0) and performs a vertical take-off manoeuvre followed by a prescribed trajectory with simultaneous motion along the x-, y- and z-axes.

The results show that the MRAC controller can accurately and consistently track the flight trajectory. The actual UAV path follows the reference trajectory closely in the straight-line segments and in the high-curvature regions, demonstrating the ability of the controller to handle strongly coupled multi-axis dynamics. The adaptive architecture also maintains precise path-following performance even in the presence of aggressive changes

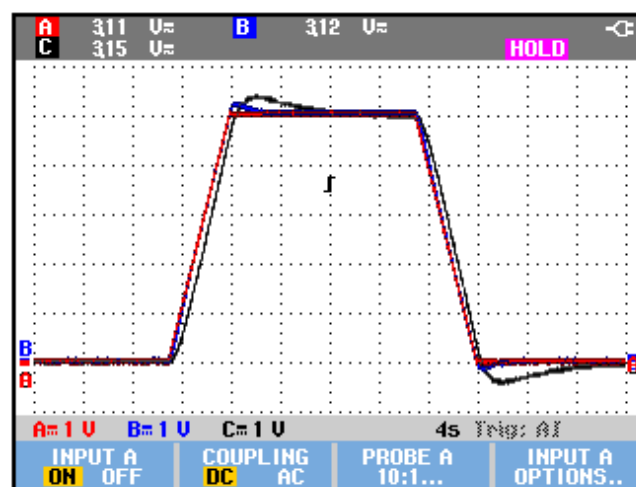
in direction, demonstrating its ability to rapidly adapt to reference changes and to effectively compensate for nonlinear system behaviour in real time.



(a)



(b)



(c)

Figure 8. Experimental tracking responses of the complex trajectory with respect to (a) x-axis, (b) y-axis and (c) z-axis using the PSO-based PID controller. Reference (red), MRAC (blue) and PSO-PID (black).

Spatial tracking errors remain well bounded, typically within a narrow band, with no apparent oscillation or drift accumulation. The altitude response is also well controlled, with small deviations from the desired flight path. These observations confirm the capability of the MRAC strategy to realise smooth and accurate trajectory tracking while maintaining robust closed-loop stability under challenging three-dimensional flight circumstances.

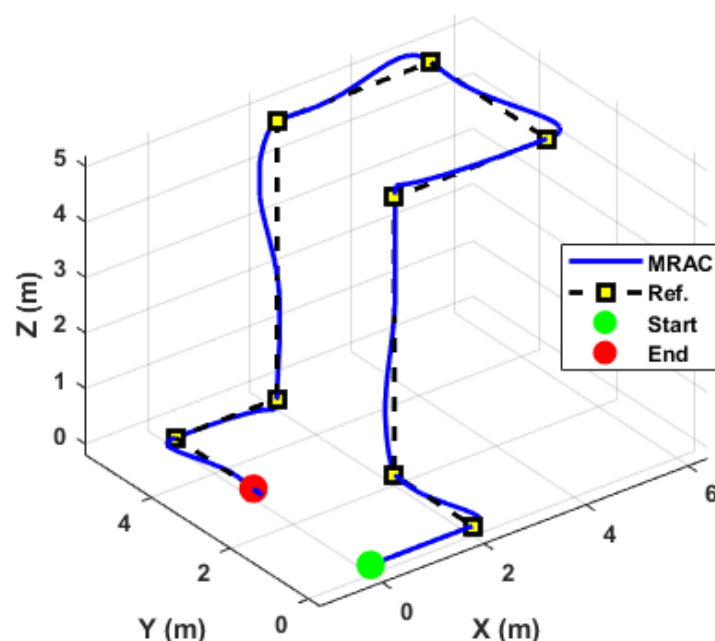


Figure 9. 3D tracking performance under the proposed MRAC architecture for the complex reference trajectory.

Figure 10 shows the 3D trajectory-tracking performance of the PSO-PID baseline controller for the same complex reference path. Starting from the origin (0 m, 0 m, 0 m), the UAV first takes off vertically and then moves along the specified multi-axis trajectory with coupled motion in x-, y- and z-directions.

Although the PSO-PID controller is able to reproduce the general shape of the reference trajectory, the tracking errors are evident throughout the whole manoeuvre. In regions of large curvature and rapid direction changes, the controller shows a limited response to sudden changes in reference, which leads to growing deviations from the desired path. Therefore, the corners of the trajectories look smoothed and rounded, which means that the transient adaptation is not sufficient, and the path-following precision is degraded.

The resultant spatial tracking errors are significantly larger than those obtained by the MRAC strategy, with large deviations during simultaneous multi-axis manoeuvres and altitude transitions. These errors are accompanied by localised tracking lag and transient oscillatory behaviour, indicative of the limited robustness of the fixed-gain control structure to highly coupled nonlinear flight dynamics.

The altitude regulation is still generally stable, but small fluctuations around the commanded flight level are observed during the horizontal motion, indicating a weaker vertical regulation performance in comparison with the adaptive controller. In summary, the results show that the PSO-PID baseline keeps the general trajectory topology, but its tracking accuracy deteriorates in aggressive three-dimensional manoeuvres, resulting in larger path-following errors, slower transient response, and lower robustness than the proposed MRAC framework.

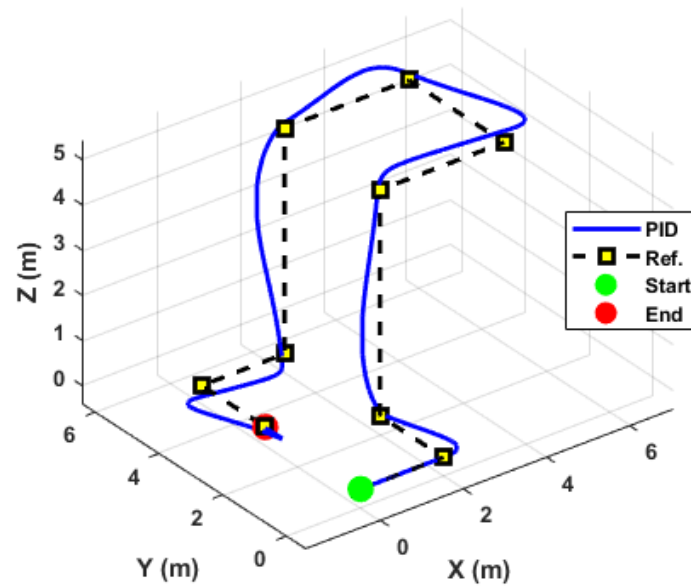


Figure 10. 3D tracking performance for the complex reference trajectory under the PSO-PID baseline.

5.4. Statistical Performance Analysis

The qualitative evaluation of the trajectories was supplemented by a quantitative statistical analysis based on the experimental data collected along the x-, y-, and z-axes. The tracking error for each axis i is defined as $e_i(t) = r_i(t) - y_i(t)$, where $r_i(t)$ and $y_i(t)$ denote the reference command and the measured system output, respectively.

The tracking accuracy and error characteristics were evaluated in terms of five commonly used performance indices: root mean square error (RMSE), mean absolute error (MAE), maximum absolute error ($e_{\max}$), integral of absolute error (IAE) and integral of time-weighted absolute error (ITAE). RMSE and MAE are metrics that quantify the overall tracking accuracy. $e_{\max}$ describes the worst-case tracking deviation. IAE and ITAE also evaluate the total effect of the tracking errors over time, where ITAE gives more importance to the errors that persist in the later stages of the response. These indices together give a complete evaluation of transient and steady-state tracking performance and allow for an objective comparison between the control strategies considered. Table 6 summarises the calculated performance indices of the MRAC- and PSO-based PID controllers. These indices give a quantitative basis for comparing the ability of the MRAC- and PSO-based PID controllers to track the trajectory of the complex three-dimensional motion.

The performance metrics reported in Table 8 were obtained from representative real-time HIL experiments performed under the same operating conditions for each reference trajectory. The presented values come from single-run evaluations and aim to offer a quantitative comparison of the tracking performance obtained by the two control strategies in the test scenarios analysed.

Table 8. Statistical comparison of trajectory-tracking performance of MRAC- and PSO-based PID controllers.

Axis	Controller	RMSE (m)	MAE (m)	Max Error (m)	IAE	ITAE
x	MRAC	0.18	0.14	0.32	2.10	6.85
	PSO-PID	0.48	0.39	0.72	5.95	18.40
y	MRAC	0.22	0.17	0.38	2.65	7.90
	PSO-PID	0.56	0.44	0.80	6.80	20.75
z	MRAC	0.20	0.16	0.35	2.45	7.10
	PSO-PID	0.62	0.50	0.85	7.50	22.30

The MRAC architecture shows consistently fewer tracking errors for all the translational axes, as summarised in Table 8. The RMSE and MAE are reduced by approximately 50–70% compared with the PSO-PID baseline, which demonstrates significantly improved trajectory-tracking accuracy. Peak tracking errors are also greatly attenuated, especially during rapid reference variations and aggressive manoeuvring conditions, demonstrating the enhanced transient response and disturbance-rejection capability of the adaptive controller. Similar improvements are also observed in the cumulative IAE and ITAE measures, which indicate reduced overall error accumulation and faster convergence to the desired trajectory.

The performance difference is more noticeable when flying at high speed in multi-axis manoeuvres with simultaneous position and altitude changes. Under these harsh operating conditions, the PSO-PID controller shows high tracking delay and local transient oscillations. On the other hand, the MRAC framework maintains smooth state transitions and suppresses the deviations from the reference command well, which is consistent with the spatial trajectory results shown in Figures 9 and 10.

To further illustrate the comparative performance, a radar-chart representation is shown in Figure 11 according to the normalised values of the five error indices. Each metric is normalised to the maximum observed value among the two control strategies, giving a common basis for comparison in a dimensionless form.

The radar chart enables a quick visual assessment of controller effectiveness. The area of the MRAC profile is much smaller than that of the PSO-PID profile, which means that the error magnitude is always lower in all the evaluation criteria. The largest improvements are obtained in the ITAE and peak error metrics, where the fixed-gain controller shows slower error decay and higher transient deviations. The compact radar signature of the MRAC controller confirms its capability to reduce simultaneously transient and accumulated tracking errors, validating its robustness and suitability for complex three-dimensional quadrotor operations.

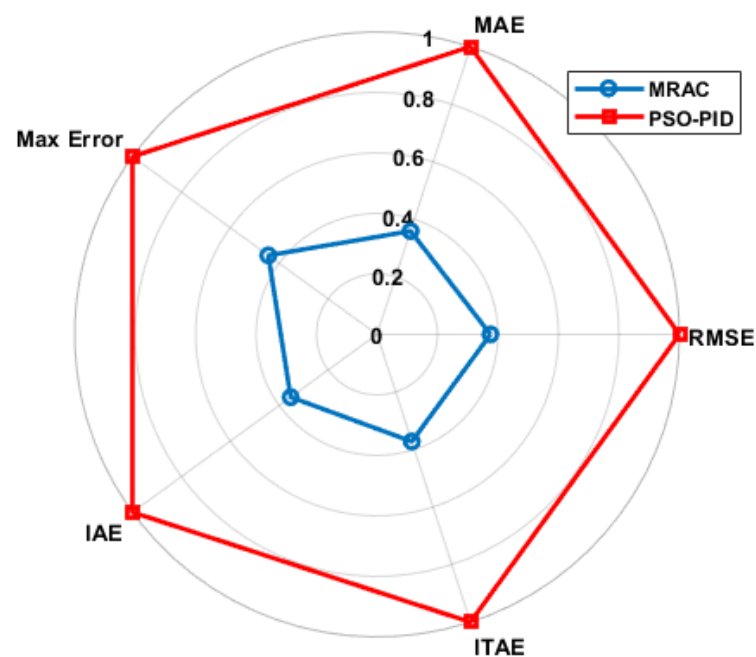


Figure 11. Radar chart comparison of the normalised performance indices of the MRAC- and PSO-based PID controllers.

5.5. Control Effort Analysis

Besides the accuracy, the control effort must be evaluated in terms of practical implementation since it directly affects the actuator saturation, energy consumption and the lifetime of components. The control allocation vector of the quadrotor UAV is composed of the collective thrust input u_1 and the attitude control torques τ_ϕ , τ_θ , and τ_ψ , which control the translational and rotational dynamics respectively.

The generated control signals were tested under the same operating conditions (set-point regulation, square-trajectory tracking, and complex three-dimensional flight manoeuvres) to assess the control demands imposed by each strategy. Special attention was paid to signal amplitude, smoothness and the presence of high-frequency oscillations, as these factors directly influence actuator performance and control efficiency.

Figure 12 depicts the collective thrust command u_1 generated by both controllers during a representative tracking experiment. The MRAC framework offers a much smoother control profile with fewer high-frequency oscillations, particularly during fast transient phases. On the other hand, the PSO-PID controller exhibits abrupt variations and high oscillations behaviour, which indicates increased sensitivity to dynamic operating conditions as well as abrupt changes in reference.

A quantitative summary of the control-effort metrics, including the root-mean-square (RMS) and peak values of the control inputs, is presented in Table 9. Less control effort is needed in all the channels for the MRAC controller. In particular, the RMS thrust demand is reduced by about 20% to 25%, and peak thrust requirements are reduced by about 25% as compared to the PSO-PID baseline. Similar reductions are also observed in the attitude control torques, demonstrating that the adaptive strategy results in better tracking performance while generating smoother and more energy-efficient actuator commands. The results confirm the effectiveness of the control and the improvements in the actuator utilisation of the proposed MRAC framework, making it suitable for demanding quadrotor flight operations.

The performance improvements are most evident in fast reference transitions, where the online adaptive mechanism continuously modifies the control gains in response to the instantaneous dynamics of the tracking error. This results in the removal of large control excursions and smoother actuator allocation without the loss of tracking accuracy. From a practical engineering aspect, this reduced control expenditure directly leads to lower power consumption and reduced mechanical actuator loading, which are essential operating constraints for extending the flight endurance of real-time UAV applications.

In addition to the trajectory tracking performance, the proposed MRAC framework brings practical operational benefits by systematically reducing the RMS and peak control effort. The reduction in control activity directly reduces the load on the propulsion system and electronic speed controllers (ESCs), thereby alleviating high-frequency mechanical wear and thermal stress. Although the power consumption and battery drain were not quantified in this work, the observed reduction in the control cost suggests that the adaptive controller results in better overall aerodynamic and electrical efficiency, and hence, it presents a promising direction for optimising the flight duration in long-duration autonomous UAV missions.

Table 9. Comparison of control effort.

Control Input	Metric	MRAC	PSO-PID
u_1	RMS (N)	9.85	12.40
	Peak (N)	14.20	18.75
τ_ϕ	RMS (N·m)	0.82	1.35
	Peak (N·m)	1.75	2.60

τ_θ	RMS (N·m)	0.88	1.42
	Peak (N·m)	1.90	2.75
τ_ψ	RMS (N·m)	0.55	0.92
	Peak (N·m)	1.20	1.85

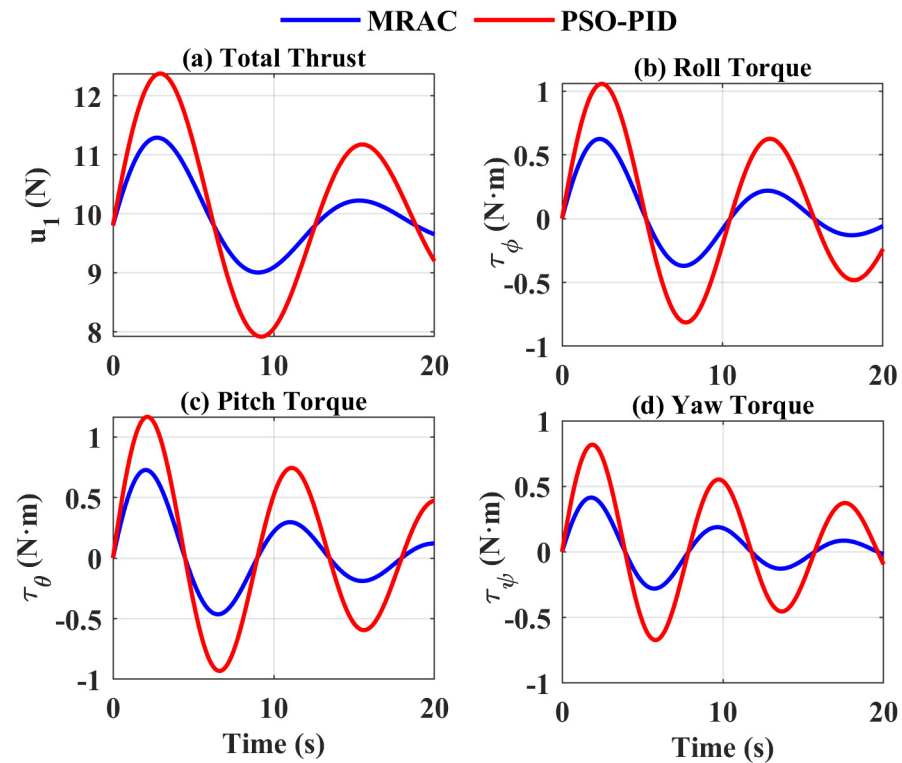


Figure 12. Control input trajectories of the MRAC and PSO-PID controllers: (a) collective thrust u_1 , (b) roll control torque τ_ϕ , (c) pitch control torque τ_θ , and (d) yaw control torque τ_ψ .

The experimental evaluation presented is based on representative deterministic HIL experiments carried out under identical operating conditions. The main purpose of these experiments is to verify the real-time implementation and to compare the proposed MRAC controller with the benchmark PSO-based PID controller under controlled conditions. Confidence interval estimation and sensitivity analysis of repeated experimental trials were not included in the present study. Thus, the reported performance improvements should be considered in light of the investigated deterministic HIL operating conditions. Future work will include extending the experimental evaluation with repeated statistical testing, sensitivity analysis, stochastic disturbance scenarios and physical flight experiments to further evaluate robustness and repeatability.

5.6. Performance Under Environmental Disturbances

A further case study was conducted to assess the robustness and resilience of the proposed control framework to environmental conditions, under representative outdoor operating conditions based on the aerodynamic environment of Badr City, Cairo, Egypt. Badr City is in an open desert area east of Cairo, where wind effects are often encountered, which could affect the operation of unmanned aerial vehicles during missions of inspection, surveillance, and monitoring. An equivalent external disturbance force was introduced into the translational dynamics of the quadrotor to represent these conditions. The disturbance is modelled as a bounded external force applied sequentially to the translational axes. The magnitude of the disturbance was chosen as 1 N, which corresponds to moderate-to-strong wind conditions for the quadrotor used in this study, or about 10% of

the weight of the vehicle. The disturbance signals were injected into the nonlinear Simulink quadrotor model as external force inputs, and the same reference trajectories, controller parameters, and experimental conditions were maintained for both the proposed MRAC and the optimised PSO-PID controller. Each disturbance was applied for a specified time interval and then removed to test the transient recovery characteristics of the controllers. This testing protocol allowed for a fair and reproducible comparison of the disturbance-rejection performance under the same HIL operating conditions. The disturbance signals were defined as

$$\begin{cases} d_x(t) = F_d [u(t - 10) - u(t - 15)] \\ d_y(t) = F_d [u(t - 15) - u(t - 20)] \\ d_z(t) = F_d [u(t - 20) - u(t - 25)] \end{cases} \quad (22)$$

where $u(\cdot)$ denotes the unit-step function, and F_d represents the disturbance magnitude, which is set to 1 N in the present study. Accordingly, the disturbance was applied along the x-axis for an interval of 10–15 s, along the y-axis for 15–20 s, and along the z-axis for 20–25 s, as shown in Figure 13.

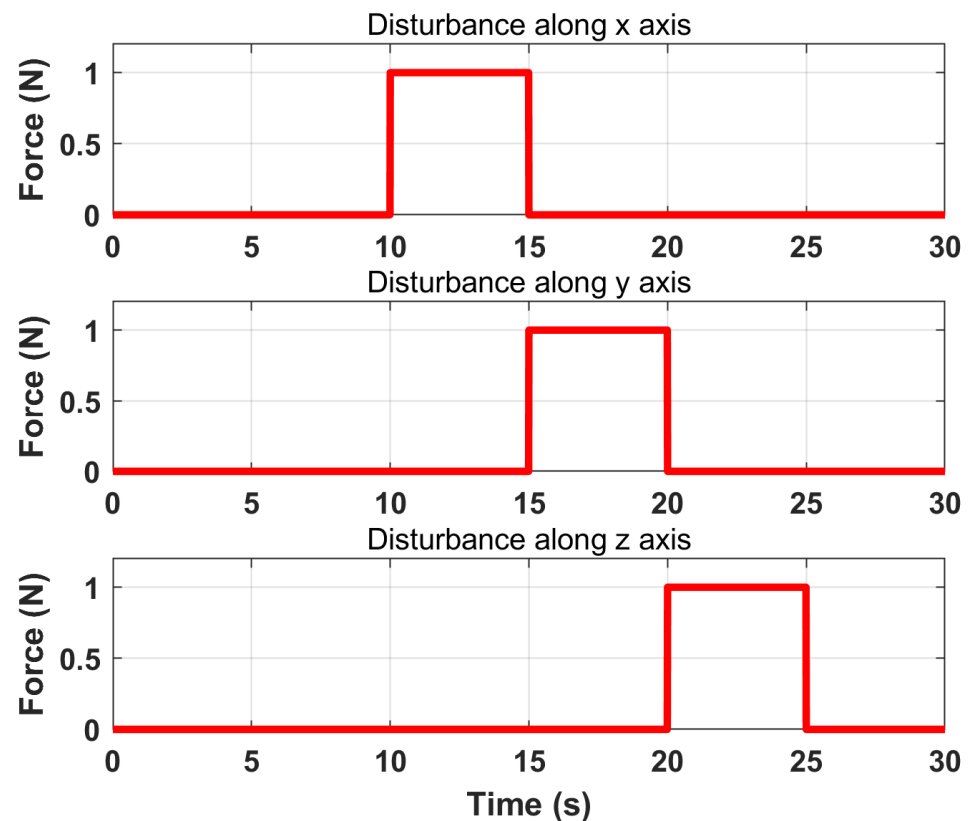
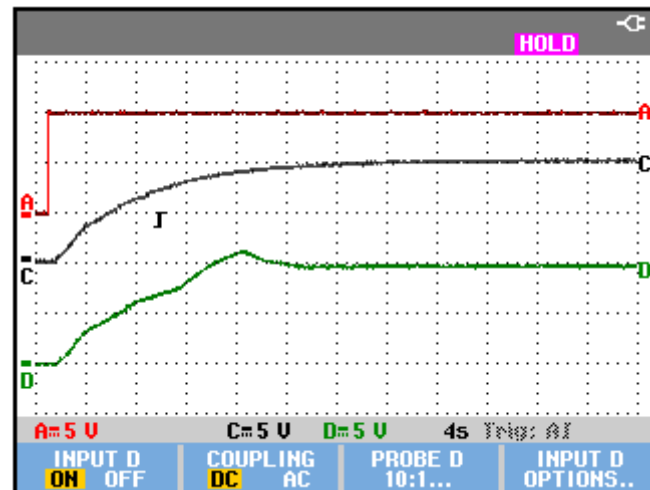
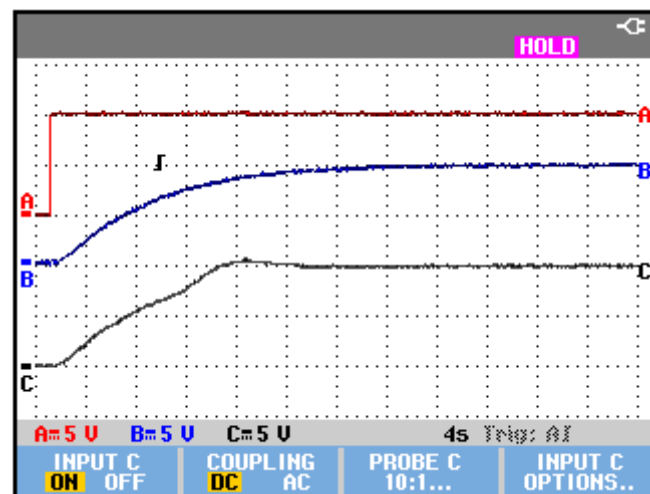


Figure 13. Input disturbance signals applied along the x-, y-, and z-axes.

Figure 14a,b present the x-axis responses obtained from the PSO-based PID and MRAC controllers respectively. The reference position was placed at 10 m, and the vertical scale of the responses is 5 m/div. Both controllers tend toward a steady-state value of about 6 m under disturbance-free conditions. The PID response converges to about 3.3 m, and the MRAC response converges to about 4.4 m after the application of the 1 N disturbance. These values are comparable to estimated reductions of about 45% and 27% for the PID and MRAC controllers respectively. The variation between the disturbed and nominal trajectories is more pronounced in the PID response, while the MRAC trajectory remains closer to its disturbance-free equivalent for the whole simulation.



(a)



(b)

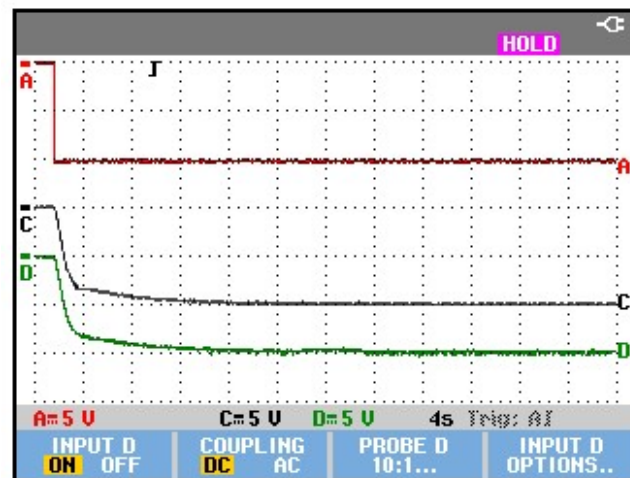
Figure 14. Trajectory-tracking performance along the x-axis, showing the reference trajectory, the nominal system response without disturbance, and the system response: (a) PSO-PID; (b) MRAC.

The responses for the y-axis are shown in Figure 15a,b at a reference altitude of -10 m, and the responses are also scaled at 5 m/div. Under nominal operating conditions, the PID and MRAC controllers reach about -5.5 m and -6 m, respectively. The disturbance is introduced, and the altitude achieved decreases to almost -3.5 m for the PID controller and -4.5 m for the MRAC controller. The estimated altitude reduction achieved is about 38% with the PID controller and 25% with the MRAC controller. The disturbed MRAC trajectory, as with the x- and y-axis responses, is always closer to the respective nominal response.

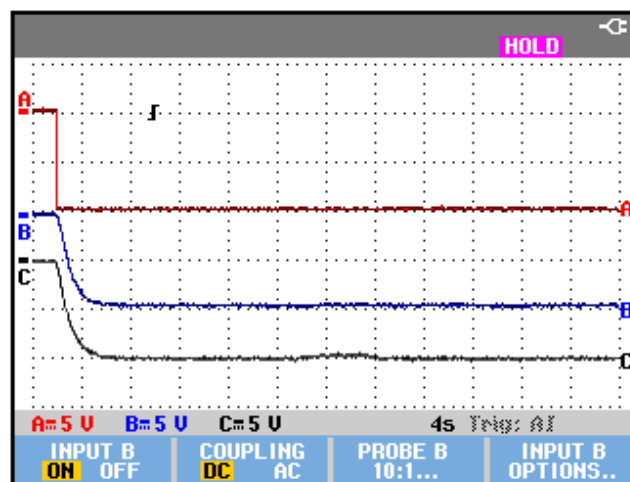
The z-axis responses for a reference position of 5 m are shown in Figure 16a,b with the same vertical scale of 5 m/div. Without disturbances, both controllers reach approximately 3 m. After disturbance injection, the PID response decreases to about 1.7 m, while the MRAC response settles to about 2.1 m. So, this corresponds to an estimated reduction of about 43% for the PID controller and 30% for the MRAC controller. While both responses are stable, the disturbed MRAC trajectory is less deviated from the nominal response.

Table 10 summarises the steady-state responses of the PSO-based PID and MRAC controllers under nominal and disturbed operating conditions for a quantitative comparison. The table shows the desired positions, the steady-state responses obtained before

and after the disturbance application, and the reduction in the obtained position corresponding to each translational axis.

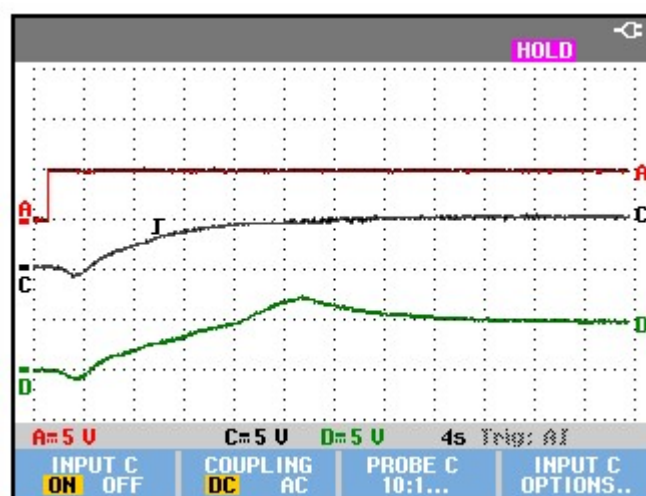


(a)

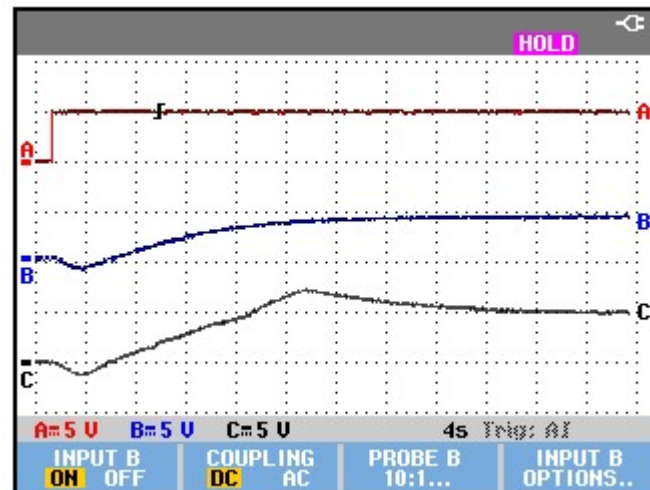


(b)

Figure 15. Trajectory-tracking performance along the y-axis, showing the reference trajectory, the nominal system response without disturbance, and the system response: (a) PSO-PID; (b) MRAC.



(a)



(b)

Figure 16. Trajectory-tracking performance along the z-axis, showing the reference trajectory, the nominal system response without disturbance, and the system response: (a) PSO-PID; (b) MRAC.

Table 10. Comparison of the RMSE-based trajectory-tracking performance of the PSO-based PID and MRAC controllers with an external disturbance of 1 N.

Axis	Controller	Nominal RMSE (m)	RMSE Under 1 N Disturbance (m)	Performance Degradation (%)
x	PSO-PID	0.48	0.70	45
	MRAC	0.18	0.23	27
y	PSO-PID	0.56	0.77	38
	MRAC	0.22	0.28	25
z	PSO-PID	0.62	0.89	43
	MRAC	0.20	0.26	30

Table 8 shows the trajectory-tracking performance of the PSO-based PID and the proposed MRAC controllers subjected to a 1 N external disturbance, with the root mean square error (RMSE) used as the evaluation metric. The nominal RMSE of the PSO-based PID controller on the x-axis increased from 0.48 m to 0.70 m under disturbance, which is a performance degradation of about 45%. Meanwhile, the MRAC controller had a lower RMSE degradation of 0.18 m to 0.23 m or an increase of only 27%. A similar trend was seen for the y-axis, where the RMSE of the PSO-based PID controller increased from 0.56 m to 0.77 m (38%), and that of the MRAC controller increased from 0.22 m to 0.28 m (25%). The RMSE for the z-axis increased from 0.62 m to 0.89 m for the PSO-based PID controller, which is approximately a 43% degradation. The MRAC controller had a smaller increase from 0.20 m to 0.26 m, which is only a 30% degradation. The rather small degradation of performance observed for both controllers is in agreement with the moderate disturbance conditions considered in this study. The applied 1 N external force is about 10% of the quadrotor weight, and it is a low-to-moderate wind disturbance rather than a severe perturbation scenario. Both controllers ensured closed-loop stability and satisfactory trajectory tracking under these operating conditions. The proposed MRAC framework, however, demonstrated consistently better disturbance rejection with lower RMSE values and less performance degradation along all translational axes compared to the PSO-based PID controller. The results demonstrate that the adaptive mechanism can effectively compensate for the external disturbances and model uncertainties, so as to keep the tracking accuracy and enhance the robustness of the whole control system.

6. Conclusion

In this paper, we propose a real-time Model Reference Adaptive Control (MRAC) scheme for quadrotor UAV trajectory tracking and validate it using numerical simulations and HIL experiments. The proposed adaptive controller performs better than the optimised PSO-PID benchmark in all the cases tested. For setpoint regulation, MRAC achieved faster convergence, reducing the settling time from about 9–12 s to 5–7 s, with negligibly small steady-state error. For trajectory-tracking tasks, such as tracking square and complex 3D flight paths, the proposed controller demonstrated better path-following accuracy and better preserved the desired trajectory geometry, reducing tracking deviations from around 0.4–0.8 m to 0.1–0.3 m. The quantitative performance evaluation further validated the superiority of the proposed controller. The MRAC consistently attained lower values of RMSE, MAE, peak error, IAE and ITAE for both translational and rotational subsystems as compared to the optimised PSO-PID controller, implying better tracking accuracy, faster error convergence and better transient response. Furthermore, the adaptive controller led to smoother thrust and torque commands with smaller RMS and peak amplitudes, thereby reducing the actuator workload while maintaining high control performance. The disturbance-rejection experiments further validated that the proposed controller can effectively attenuate external disturbances while maintaining trajectory accuracy in representative operating conditions. From a practical point of view, the proposed controller is supported by a Lyapunov-based stability analysis and was successfully implemented in a real-time HIL environment by using the OPAL-RT platform. The HIL validation is an important intermediate step between the numerical simulation and the physical flight testing. It demonstrates the feasibility of the implementation of the proposed MRAC framework for real-time control of quadrotor dynamics that are underactuated, nonlinear and strongly coupled. However, certain limitations need to be recognised. The results presented here are from representative deterministic HIL experiments conducted under the same operating conditions. Statistical analyses of repeated trials, confidence intervals, and sensitivity studies were outside the scope of the present work. Furthermore, while the HIL platform provides a high-fidelity real-time validation environment, it is impossible to fully replicate all aerodynamic effects, actuator nonlinearities, sensor imperfections, and environmental uncertainties experienced in actual flight. Moreover, the experimental validation was limited to representative setpoint regulation, trajectory-tracking and disturbance-rejection scenarios. Thus, the HIL results presented here should be considered a validation step in between and not as a replacement for physical flight tests. Future work will focus on repeated HIL experiments, rigorous statistical assessment, and then onboard implementation and experimental flight validation using a physical quadrotor platform. Further investigations will study more demanding flight scenarios such as aggressive manoeuvres, quickly varying trajectories, obstacle-avoidance missions and strong environmental disturbances. Moreover, disturbance-observer-based compensation, adaptive state estimation, online parameter identification, learning-assisted adaptive control, and fault-tolerant control strategies will be further studied to improve the robustness, adaptability, and practical implementation of the proposed MRAC framework for autonomous UAV missions. The proposed controller shows an improvement in tracking and disturbance-rejection performance in deterministic real-time HIL tests, but the evaluation is currently performed on representative experimental runs. Future work includes repeated experimental trials, confidence interval analysis, sensitivity analysis and validation on a physical quadrotor platform under stochastic environmental conditions.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/drones10070545/s1>: A video demonstrating the OPAL-RT HIL experimental platform and representative trajectory-tracking tests is available in the Supplementary Material accompanying this manuscript.

Author Contributions: Conceptualisation, M.F.E.-K. and M.A.; methodology, M.F.E.-K. and M.M.E.-S.; software, M.F.E.-K. and A.A.; validation, M.F.E.-K., M.A., and M.M.E.-S.; formal analysis, M.F.E.-K. and M.M.E.-S.; investigation, M.F.E.-K. and S.S.; resources, M.A. and S.S.; data curation, M.F.E.-K. and A.A.; writing—original draft preparation, M.F.E.-K.; writing—review and editing, M.A., M.M.E.-S., S.S. and A.A.; visualisation, M.F.E.-K. and A.A.; supervision, M.A. and M.M.E.-S.; project administration, M.F.E.-K.; funding acquisition, M.A.; experimental setup and HIL implementation, M.F.E.-K., S.S. and A.A.; results interpretation, M.F.E.-K., M.M.E.-S. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are included within the article. Additional datasets and materials related to the reported results are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, H.; Chen, M.; Li, S. Adaptive Neural Network Control for Quadrotor UAVs with Uncertain Nonlinear Dynamics. *ISA Trans.* **2020**, *98*, 214–226.
2. Zhang, Y.; Jiang, B.; Cocquempot, V. Fault-Tolerant Control for Quadrotor UAVs: A Survey. *Annu. Rev. Control* **2020**, *49*, 209–227.
3. Li, S.; Wang, J.; Yang, G. Robust Trajectory Tracking Control for Quadrotor UAVs with Disturbances. *Aerosp. Sci. Technol.* **2021**, *109*, 106418.
4. Liu, C.; Zhang, D.; Chen, W. Finite-Time Adaptive Control of Quadrotor UAV Systems under Uncertainties. *IEEE Trans. Ind. Electron.* **2021**, *68*, 12463–12473.
5. Zhou, Q.; Li, H.; Shi, P. Adaptive Dynamic Surface Control for Quadrotor UAVs with External Disturbances. *Control Eng. Pract.* **2021**, *108*, 104699.
6. Chen, M.; Tao, G.; Jiang, B. Adaptive Control for Uncertain Nonlinear Systems: A Review and Applications. *Annu. Rev. Control* **2022**, *53*, 42–58.
7. Shao, X.; Meng, Q.; Liu, J.; Wang, H. RISE and Disturbance Compensation Based Trajectory Tracking Control for a Quadrotor UAV without Velocity Measurements. *Aerosp. Sci. Technol.* **2018**, *74*, 145–159.
8. Tordesillas, J.; Lopez, M.M.; How, J.P. FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments. *IEEE Trans. Robot.* **2021**, *37*, 1351–1368.
9. Alessandri, A.; Baglietto, M.; Battistelli, G. Real-Time HIL Simulation for UAV Control Validation. *Simul. Model. Pract. Theory* **2020**, *103*, 102097.
10. Zhao, H.; Yang, X.; Li, Y. Adaptive Robust Control for Quadrotor UAVs under Time-Varying Uncertainties. *Aerosp. Sci. Technol.* **2024**, *145*, 108512.
11. Sun, L.; Ge, S.S.; Lee, T.H. Adaptive Control of Nonlinear UAV Systems with Parametric Uncertainties. *IEEE Trans. Cybern.* **2020**, *50*, 3136–3148.
12. Li, Y.; Zhou, Q.; Li, H. Prescribed Performance Adaptive Control for Quadrotor UAV Trajectory Tracking. *ISA Trans.* **2021**, *116*, 267–279.
13. Wu, Z.; Jiang, B.; Shi, P. Event-Triggered Adaptive Control for Quadrotor UAV Systems. *IEEE Trans. Ind. Electron.* **2022**, *69*, 4153–4162.
14. Yang, X.; Zhang, H.; Wen, C. Finite-Time Backstepping Control for Quadrotor UAVs with External Disturbances. *Aerosp. Sci. Technol.* **2022**, *120*, 107295.
15. Chen, J.; Chen, M. Adaptive Sliding Mode Control for Quadrotor UAV Trajectory Tracking. *Control Eng. Pract.* **2022**, *121*, 105043.
16. El-Khatib, M.F.; Khater, F.M.H.; Hendawi, E.; Abu El-Sebah, M.I. Simplified and Intelligent Controllers for Multi-Input Multi-Output Processes. *Eng. Appl. Artif. Intell.* **2025**, *141*, 109816.
17. Madebo, N.W.; Abdissa, C.M.; Lemma, L.N. Enhanced Trajectory Control of Quadrotor UAV Using Fuzzy PID Based Recurrent Neural Network Controller. *IEEE Access* **2024**, *12*, 190454–190469.

18. Chen, F.; Jiang, R.; Zhang, K.; Jiang, B.; Tao, G. Robust Backstepping Sliding-Mode Control and Observer-Based Fault Estimation for a Quadrotor UAV. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5044–5056.
19. Glushchenko, A.; Lastochkin, K. Quadrotor Trajectory Tracking Using Model Reference Adaptive Control, Neural Network-Based Parameter Uncertainty Compensator, and Different Plant Parameterizations. *Computation* **2023**, *11*, 163.
20. Abdelbar, M.; Abdellatif, A.; Hegaze, M. Optimization Based Optimal Controller for the Upper-Limb Exoskeleton (SAMA). *J. Adv. Res. Appl. Mech.* **2025**, *139*, 60–77.
21. Tang, G.; Wang, D.; Liu, A.; Qiao, J. Adjustable Behavior-Guided Adaptive Dynamic Programming for Neural Learning Control. *Neurocomputing* **2025**, *636*, 129986.
22. Lee, T. Geometric Adaptive Tracking Control of a Quadrotor UAV on SE(3). *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 2259–2271.
23. Zhao, Z.; Cao, D.; Yang, J.; Wang, H. High-Order Sliding Mode Observer-Based Trajectory Tracking Control for a Quadrotor UAV with Uncertain Dynamics. *Nonlinear Dyn.* **2020**, *102*, 2583–2596.
24. Wang, J.; Xia, Y. Neural Adaptive Fault-Tolerant Control for Quadrotor UAVs. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 1758–1770.
25. Azhari, A.A.; Ahmed, W.M.; El-Khatib, M.F.; Abdellatif, A. AI-Driven Trajectory Planning of Dentatron: A Compact 4-DOF Dental Robotic Manipulator. *Biomimetics* **2025**, *10*, 803.
26. García-Beltrán, C.D.; Miranda-Araujo, E.M.; Guerrero-Sanchez, M.E.; Valencia-Palomo, G.; Hernández-González, O.; Gómez-Peñate, S. Passivity-Based Control Laws for an Unmanned Powered Parachute Aircraft. *Asian J. Control* **2021**, *23*, 2087–2096.
27. Valencia-Palomo, G.; Rossiter, J.A.; López-Estrada, F.R. Improving the Feed-Forward Compensator in Predictive Control for Setpoint Tracking. *ISA Trans.* **2014**, *53*, 755–766.
28. Alexis, K.; Papachristos, C.; Siegwart, R.; Tzes, A. Robust Model Predictive Flight Control of Unmanned Rotorcrafts. *J. Intell. Robot. Syst.* **2016**, *81*, 443–469.
29. Zhou, Y.; Zhao, H. Deep Reinforcement Learning for UAV Trajectory Tracking: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3537–3552.
30. Nguyen, N.P.; Mung, N.X.; Thanh, H.L.N.N.; Huynh, T.T.; Lam, N.T.; Hong, S.K. Adaptive Sliding Mode Control for Attitude and Altitude System of a Quadcopter UAV via Neural Network. *IEEE Access* **2021**, *9*, 40076–40085.
31. Akella, M.R.; Hutchinson, S. Adaptive Flight Control Design for Unmanned Aerial Vehicles. *IEEE Control Syst. Mag.* **2023**, *43*, 68–89.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.