

Article

A Robust Adaptive PID-like Controller for Quadrotor Unmanned Aerial Vehicle Systems

Ahsene Boubakir ¹, Toufik Souanef ^{2,*}, Salim Labiod ¹ and James F. Whidborne ²

¹ Automatic Control Laboratory of Jijel, Faculty of Science and Technology, University of Jijel, BP. 98, Jijel 18000, Algeria; ahsene.boubakir@univ-jijel.dz (A.B.); salim.labiod@univ-jijel.dz (S.L.)

² Centre for Aeronautics, Cranfield University, College Road, Cranfield MK43 0AL, UK; j.f.whidborne@cranfield.ac.uk

* Correspondence: toufik.souanef@cranfield.ac.uk

Abstract: This paper introduces a stable adaptive PID-like control scheme for quadrotor Unmanned Aerial Vehicle (UAV) systems. The PID-like controller is designed to closely estimate an ideal controller to meet specific control objectives, with its gains being dynamically adjusted through a stable adaptation process. The adaptation process aims to reduce the discrepancy between the ideal controller and the PID-like controller in use. This method is considered model-free, as it does not require knowledge of the system's mathematical model. The stability analysis performed using a Lyapunov method demonstrates that every signal in the closed-loop system is Uniformly Ultimately Bounded (UUB). The effectiveness of the proposed PID-like controller is validated through simulations on a quadrotor for path following, ensuring accurate monitoring of the target positions and yaw angle. Simulation results highlight the performance of this control scheme.

Keywords: unmanned aerial vehicle; PID-like controller; ideal controller; second-order nonlinear system; model-free control



Citation: Boubakir, A.; Souanef, T.; Labiod, S.; Whidborne, J.F. A Robust Adaptive PID-like Controller for Quadrotor Unmanned Aerial Vehicle Systems. *Aerospace* **2024**, *11*, 980. <https://doi.org/10.3390/aerospace11120980>

Academic Editor: Hailong Huang

Received: 16 October 2024

Revised: 20 November 2024

Accepted: 25 November 2024

Published: 27 November 2024



Copyright: © 2024 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last three decades, UAV technology has rapidly gained popularity, driven by advancements in computing and positioning technologies (such as GPS) and the numerous benefits UAVs offer. The field of UAVs has expanded beyond military applications, with numerous potential civil applications emerging. These include fireground surveillance, agriculture, film production, large infrastructure management, reconnaissance, and search and rescue operations [1].

The automatic control of UAV systems, including path tracking and attitude stabilization [2], as well as observer design [3,4], has garnered significant attention from researchers in the control community. Control techniques for UAV systems vary based on the complexity and requirements of the application. PID control [5,6] is widely used for its simplicity, adjusting control inputs based on error correction. Advanced methods like Model Predictive Control (MPC) [7] and Linear Quadratic Regulator (LQR) [8,9] optimize performance and handle multivariable systems but require significant computational resources. Sliding Mode Control (SMC) [10] and adaptive control [11,12] offer robustness against uncertainties, though they can be complex to implement due to the chattering problem and computation time. Backstepping control [13] is a powerful tool for managing the nonlinear dynamics of quadrotor UAVs, offering systematic design benefits. However, its complexity and computational demands need to be carefully managed to ensure effective and reliable control in practical applications. Fuzzy logic and neural network controls [14,15] manage imprecision and nonlinear dynamics effectively, with neural networks learning from data. Reinforcement learning (RL) [16] shows promise for complex environments through trial-and-error learning, albeit with high computational demands. Other control techniques can be found in the literature, such as an auto-tuning adaptive PID controller based on sliding

mode methodology [17], filtered observer-based IDA-PBC control [18], and fault-tolerant control [19]. Each technique offers distinct advantages and limitations, making the choice dependent on the specific needs of the UAV system.

It is important to highlight that most of the control methods proposed in the literature for quadrotor UAVs rely on knowledge of the model, such as in [9,11,13,20] and other works in the literature. Model-based controllers offer precise management but have notable disadvantages. They depend on the accuracy of the UAV model, which can be complex and time-consuming to develop. These approaches also require significant computational resources, which can strain limited onboard processing capabilities. Additionally, a model-based control may struggle with adapting to physical changes or varying environmental conditions, and it can be sensitive to model uncertainties. In contrast, model-free controllers [21–23] simplify implementation by avoiding the need for accurate models and adapt well to changing conditions. They are robust to uncertainties, require less computational power, and provide effective real-time performance by adjusting based on feedback. They can be applied across different drone types and configurations without necessitating model-specific adjustments. Furthermore, they can expedite the deployment of drones in dynamic or unforeseen scenarios where quick and efficient control is paramount, such as in search and rescue operations, disaster response, and other time-critical missions. The primary motivation behind this research is to develop a robust model-free controller that can easily adapt to the effects of uncertainties and disturbances. By avoiding reliance on a precise system model, the proposed approach aims to enhance flexibility and resilience, making it well suited for practical applications in dynamic and uncertain environments.

PID control algorithms are the most widely used in automatic control [24,25], having been employed in process industries since the 1940s. Their popularity is due to their versatility, simple design, reliability, and ease of implementation on various platforms. Today, about 90% of manufacturing systems are managed by PID controllers. Designing an effective PID controller involves tuning three key parameters, proportional gain, integral gain, and derivative gain, to attain the most effective control outcomes. To obtain acceptable control outcomes, the values of the controller gains must be tuned. Tuning PID controllers involves various methods to achieve optimal system performance [26–28]. Common heuristic methods include the Ziegler–Nichols and Cohen–Coon methods, trial-and-error, and rule-based tuning. Analytical methods involve the root locus method, frequency response methods (Bode and Nyquist plots), optimization-based approaches (like gradient descent and genetic algorithms), pole placement, and internal model control. Modern systems also use software-based methods, such as auto-tuning algorithms and simulation-based tuning, to automatically and iteratively adjust PID parameters [27,28]. These methods are selected and often combined based on the specific application and system needs.

This research paper introduces a stable adaptive PID-like controller to ensure the effective path tracking of a quadrotor UAV. We begin by presenting the principle of the proposed stable adaptive PID-like control strategy, specifically designed for a general class of second-order nonlinear systems. The fundamental concept of this control scheme is to employ a PID-like controller to approximate an ideal controller that fulfils control objectives. To this end, the adaptive laws for its gains are developed using the gradient descent technique, with the goal of minimizing the error relative to the unknown ideal controller and the PID-like controller in use. The stability of the entire closed-loop system is analyzed using a Lyapunov approach. This proposed PID-like controller guarantees the boundedness of all variables in the closed-loop system and ensures that the output tracking error converges to a small neighbourhood close to the origin. Given that this strategy does not depend on knowledge of the model, the introduced PID-like controller may be viewed as a straightforward and efficient model-free controller.

In summary, the primary advantages of the proposed control method and the contributions of this work are as follows:

1. The adaptive PID-like controller is designed directly for a general class of second-order nonlinear systems, under certain acceptable assumptions. Therefore, it can be applied to various types of UAV systems. An analysis of the stability of the closed-loop system is conducted using a Lyapunov method, confirming that all signals in the closed-loop system exhibit uniform ultimate boundedness.
2. A notable feature of the proposed controller is that all initial values of the estimated parameters can be set to zero. This simplification facilitates the implementation process and reduces the need for precise initial parameter estimation, enhancing the practicality of the controller in various applications.
3. Thanks to the integral structure of its adaptive law, the proposed robust adaptive PID-like controller generates a smooth control signal, unlike sliding mode-based control approaches as discussed in [10,29,30] and other papers in the literature.
4. In contrast to the backstepping-based control methods described in [13,31,32] and other works in the literature, the proposed approach prevents the issue of “explosion of complexity” because the controller design in this scheme does not require the repeated differentiation of specific functions.
5. Compared to the methods in [9,12,13,20] and other studies in the literature, the proposed robust adaptive PID-like controller does not require knowledge of the controlled UAV system model for its implementation. Therefore, it can be considered as an efficient robust model-free controller for UAV systems.

The structure of this paper is as follows: Section 2 covers the dynamic modelling of a quadrotor UAV and outlines the control objectives. Section 3 details the design of the proposed robust adaptive PID-like controller. Section 4 discusses the application of the designed controller to a quadrotor UAV and presents the simulation results. Finally, Section 5 summarizes and concludes the research work.

2. Dynamic Modelling and Control Objectives for a Quadrotor UAV

2.1. Mathematical Model of a Quadrotor

Figure 1 depicts a quadrotor UAV system comprising four rotors. Two diagonal motors (1 and 3) operate in the same direction, while the others (2 and 4) operate in the opposite direction to counteract torque. Adjusting rotor speed enables control over lift force and movement generation. The equations that dictate the positional movements of a quadrotor UAV are fundamentally aligned with the principles governing a rotating rigid body that possesses six degrees of freedom [33,34].

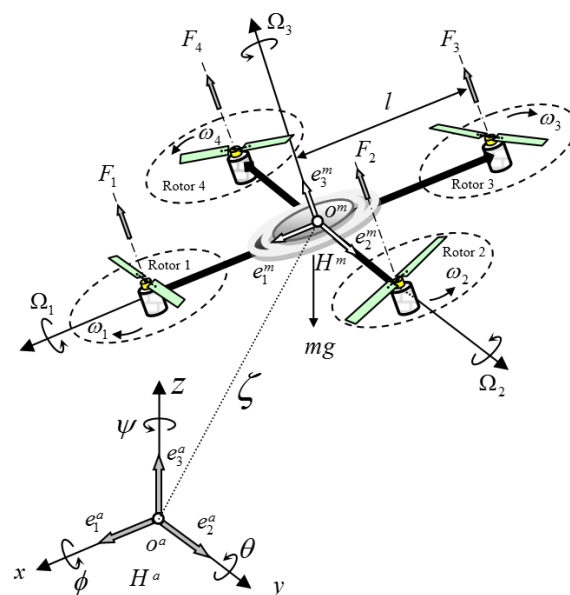


Figure 1. Quadrotor description in inertial and body-fixed frames.

Consider two primary reference frames (illustrated in Figure 1): the Earth-fixed inertial reference frame $H^a(\vec{o}^a, \vec{e}_1^a, \vec{e}_2^a, \vec{e}_3^a)$ and the body-fixed reference frame $H^m(\vec{o}^m, \vec{e}_1^m, \vec{e}_2^m, \vec{e}_3^m)$. The quadrotor's absolute position is described by $\vec{o}^a \vec{o}^m = \zeta = [x, y, z]^T$ and its orientation by the three Euler angles (ϕ, θ, ψ) .

In the reference frame H^m , the translational velocities are defined as $v = [v_1, v_2, v_3]^T$ and the rotational velocities as $\Omega = [\Omega_1, \Omega_2, \Omega_3]^T$. Let $\Gamma = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T \in H^a$ represent the angular velocity and $\dot{\zeta} = [\dot{x}, \dot{y}, \dot{z}]^T \in H^a$ represent the translational velocity of the airframe in the Earth-fixed inertial reference frame. The relationship among the velocity vectors $(\dot{\zeta}, \Gamma)$ and (v, Ω) is given by

$$\begin{cases} \dot{\zeta} = R_t v \\ \Omega = R_r \Gamma \end{cases} \quad (1)$$

Here, R_t and R_r represent the velocity transformation matrix and rotation velocity matrix between H^a and H^m , respectively [34].

By applying Newton's laws in the reference frame H^m , considering external forces $\sum F_{ext}$ and moments $\sum T_{ext}$ acting on the centre of mass, the dynamic equations of motion for the quadrotor UAV [33] are derived as follows:

$$\begin{cases} m\dot{v} + \Omega \times (mv) = \sum F_{ext} \\ I_T \dot{\Omega} + \Omega \times (I_T \Omega) = \sum T_{ext} \end{cases} \quad (2)$$

where m represents the mass, and $I_T = \text{diag}(I_x, I_y, I_z)$ denotes the total inertia matrix of the quadrotor. $\sum F_{ext}$ and $\sum T_{ext}$ encompass external forces and torques acting at the centre of the quadrotor, aligned with the orientation of the reference frame H^m . These forces and torques can be expressed by

$$\begin{cases} \sum F_{ext} = F_{prop} - F_{aero} - F_{grav} \\ \sum T_{ext} = T_{prop} - T_{aero} - T_{gyro} \end{cases} \quad (3)$$

with $F_{aero} = K_t v$ representing aerodynamic friction, with $K_t = \text{diag}(K_{tx}, K_{ty}, K_{tz})$ as the matrix of translation drag coefficients. $T_{aero} = K_r \Omega$ represents aerodynamic drag, with $K_r = \text{diag}(K_{r\phi}, K_{r\theta}, K_{r\psi})$ as the matrix of aerodynamic drag coefficients. $F_{grav} = mR_t^T G$ denotes gravitational force effects, where $G = [0, 0, g]^T$ and $g = 9.81 \text{ m/s}^2$. The gyroscopic torques T_{gyro} arise from the interaction of the airframe rotation speed Ω and the speeds w_i of the four propellers, given by $T_{gyro} = \sum_{i=1}^4 \Omega \times I_R W_i$ with $W_i = [0, 0, (-1)^{i+1} w_i]^T$. The forces and torques produced by the propeller system are denoted as

$$F_{prop} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \sum_{i=1}^4 w_i^2 \end{bmatrix}, \quad T_{prop} = \begin{bmatrix} l b (w_2^2 - w_4^2) \\ l b (w_3^2 - w_1^2) \\ c \sum_{i=1}^4 (-1)^{i+1} w_i^2 \end{bmatrix} \quad (4)$$

where $F_i = b w_i^2$ represents the thrust force produced by rotor i , $i = 1 : 4$, w_i is the rotational speed of rotor i , $b > 0$ is the thrust coefficient, and l is the distance from the centre of the quadrotor to the axes of the rotors, while $c > 0$ denotes the drag coefficient.

By using Equations (2)–(4) and assuming that $(\Omega_1, \Omega_2, \Omega_3) \approx (\dot{\phi}, \dot{\theta}, \dot{\psi})$, the complete dynamic model of the quadrotor is formulated as follows:

$$\begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}((I_y - I_z)/I_x) - (I_R/I_x)\dot{\theta}\bar{w} - (K_{r\phi}/I_x)\dot{\phi} + (l/I_x)u_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}((I_z - I_x)/I_y) + (I_R/I_y)\dot{\phi}\bar{w} - (K_{r\theta}/I_y)\dot{\theta} + (l/I_y)u_3 \\ \ddot{\psi} = \dot{\phi}\dot{\theta}((I_x - I_y)/I_z) - (K_{r\theta}/I_y)\dot{\theta} + (1/I_z)u_4 \\ \ddot{x} = -(K_{tx}/m)\dot{x} + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)(1/m)u_1 \\ \ddot{y} = -(K_{ty}/m)\dot{y} + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)(1/m)u_1 \\ \ddot{z} = -(K_{tz}/m)\dot{z} - g + ((\cos\phi\cos\theta)/m)u_1 \end{cases} \quad (5)$$

with $\bar{w} = \sum_{i=1}^4 (-1)^{i+1} w_i$, and $u = [u_1, u_2, u_3, u_4]^T$ denotes the system inputs such that

$$\begin{cases} u_1 = b(w_1^2 + w_2^2 + w_3^2 + w_4^2) \\ u_2 = b(w_2^2 - w_1^2) \\ u_3 = b(w_3^2 - w_1^2) \\ u_4 = c(w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{cases} \quad (6)$$

where u_1 is the total thrust generated by all four rotors, while u_2 , u_3 , and u_4 denote the roll, the pitch, and the yaw torques, respectively.

Remark 1. Quadrotor UAV systems are underactuated and cannot be linearized using static-state feedback when the tracking outputs are x , y , z , and ψ . A common and effective solution to this problem is the use of virtual control inputs, which simplify the control design by transforming the underactuated problem into a series of fully actuated control problems, thereby facilitating the creation of stable and robust control laws for the quadrotor UAV systems.

2.2. Control Objectives and Control Architecture

This research introduces a stable adaptive PID-like control method for quadrotor UAV systems. We aim to ensure that the positions (x, y, z, ψ) converge towards their respective desired trajectories (x_d, y_d, z_d, ψ_d) . As the movements along the x and y axes correspond to pitch and roll rotations, respectively, we need two controllers to ensure that the angles (ϕ, θ) also converge to their desired values (ϕ_d, θ_d) , as illustrated in Figure 2. In both instances, an adaptive PID-like controller will be employed to achieve this objective.

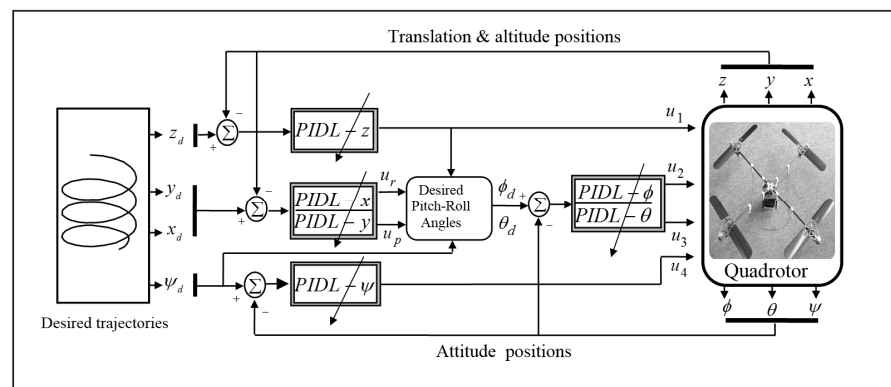


Figure 2. Adaptive PID-like controller for quadrotor UAV systems.

The dynamics model (5) reveals that the motion along the x and y axes is contingent upon the control input u_1 . Specifically, u_1 represents the total thrust vector engineered to achieve the intended linear motion.

Let us define two components u_x and u_y by

$$\begin{cases} u_x = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ u_y = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \end{cases} \quad (7)$$

u_x and u_y can be regarded as the orientations of u_1 that control movement along the x and y axis, respectively. Therefore, using Equation (7), the desired roll angle ϕ_d and the desired pitch angle θ_d , which enable appropriate motions along the x and y axis, can be determined as follows:

$$\begin{cases} \phi_d = \arcsin(u_x \sin(\psi_d) - u_y \cos(\psi_d)) \\ \theta_d = \arcsin((u_x \cos(\psi_d) + u_y \sin(\psi_d)) / \cos(\phi_d)) \end{cases} \quad (8)$$

Utilizing Equations (5) and (7), the translation subsystem can be represented by the following expression:

$$\begin{cases} \ddot{x} = -(K_{tx}/m)\dot{x} + (1/m)u_r \\ \ddot{y} = -(K_{ty}/m)\dot{y} + (1/m)u_p \\ \ddot{z} = -(K_{tz}/m)\dot{z} - g + ((\cos \phi \cos \theta)/m)u_1 \end{cases} \quad (9)$$

where $u_r = u_x u_1$ and $u_p = u_y u_1$.

To simplify applying the adaptive PID-like controller proposed in the previous section to the quadrotor UAV systems, we define the output vector as $Y = [\phi, \theta, \psi, x, y, z]^T$, the control input vector as $u = [u_1, u_2, u_3, u_4, u_r, u_p]^T$, and the state space vector as $X = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}]^T$, $F(X) = [f_1(X), \dots, f_6(X)]^T$, and $D(X) = \text{diag}(d_1(X), \dots, d_6(X))$ with the elements $f_i(X)$ and $d_i(X)$, $i = 1 : 6$, specified as

$$\begin{aligned} f_1(X) &= \dot{\theta}\dot{\psi}((I_y - I_z)/I_x) - (I_R/I_x)\dot{\theta}\bar{\omega} - (K_{r\phi}/I_x)\dot{\phi} & f_4(X) &= -(K_{tx}/m)\dot{x} \\ f_2(X) &= \dot{\phi}\dot{\psi}((I_z - I_x)/I_y) + (I_R/I_y)\dot{\phi}\bar{\omega} - (K_{r\theta}/I_y)\dot{\theta} & f_5(X) &= -(K_{ty}/m)\dot{y} \\ f_3(X) &= \dot{\phi}\dot{\theta}((I_x - I_y)/I_z) - (K_{r\psi}/I_z)\dot{\psi} & f_6(X) &= -(K_{tz}/m)\dot{z} - g \end{aligned}$$

$$\begin{aligned} d_1(X) &= I/I_x, \quad g_2(X) = I/I_y, \quad g_3(X) = I/I_z \\ d_4(X) &= 1/m, \quad g_5(X) = 1/m, \quad g_6(X) = (\cos \phi \cos \theta)/m. \end{aligned}$$

Thus, the quadrotor system described by Equation (5) is represented by

$$\ddot{Y} = F(X) + D(X)u \quad (10)$$

which aligns with the general input–output form outlined in Equations (11) and (13), where $p = 6$ in this case. Furthermore, $-0.5\pi < \phi < +0.5\pi$ and $-0.5\pi < \theta < +0.5\pi$, which ensures that $d_6(X) > 0$, thus affirming the positive definiteness of the matrix $D(X)$.

3. Proposed Robust Adaptive PID-like Controller

3.1. Problem Statement

We introduce, in this section, a stable adaptive PID-like controller for a general second-order nonlinear system Π , consisting of p subsystems Π_i . This class of systems is outlined in the subsequent normal form

$$\Pi_i \begin{cases} \dot{x}_{i1} = x_{i2} \\ \dot{x}_{i2} = f_i(X) + d_i(X) u_i, \quad i = 1, \dots, p \\ y_i = x_{i1} \end{cases} \quad (11)$$

with $X = [x_{11}, x_{12}, x_{21}, x_{22}, \dots, x_{p1}, x_{p2}]^T \in \mathbb{R}^{2p}$ representing the overall state vector, assumed to be measurable; $u = [u_1, u_2, \dots, u_p]^T$ denotes the control input vector and $Y = [y_1, y_2, \dots, y_p]^T$ stands for the output vector; $f_i(X)$ and $d_i(X)$, $i = 1, \dots, p$, represent unknown and smooth nonlinear functions.

In light of the fact that the suggested adaptive PID-like controller functions as a model-free control approach, we will construct this controller directly for the general class of second-order nonlinear systems detailed in (11).

Let us define

$$\ddot{Y} = [\ddot{y}_1, \dots, \ddot{y}_p]^T, F(X) = [f_1(X), \dots, f_p(X)]^T \text{ and } D(X) = \text{diag}(d_1(X), \dots, d_p(X))$$

The dynamic system (11) can be rewritten in the following compact form:

$$\ddot{y}_i = f_i(X) + d_i(X) u_i, \quad i = 1, \dots, p \quad (12)$$

Therefore, the system Π reads as

$$\ddot{Y} = F(X) + D(X)u \quad (13)$$

We seek to establish a controller that maintains the boundedness of all variables within the closed-loop system while ensuring the tracking of a predetermined desired trajectory $Y_d(t) = [y_{d1}(t), \dots, y_{dp}(t)]^T$.

Remark 2. Although the second-order nonlinear system Π is fully actuated, unlike quadrotor UAV systems, the path-tracking problem for a quadrotor UAV with two virtual control inputs results in a dynamic model (10) that can be viewed and treated as a second-order nonlinear system. This transformation simplifies the controller design and enables the application of the proposed robust PID-like controller to quadrotor UAV systems.

In this paper, we base our analysis on the following assumptions:

Assumption 1. The matrix $D(X)$ in (13) is assumed to be positive definite and bounded as $0 < \bar{d}_0 I_p < D(X) < \bar{d}_1 I_p$ where I_p denotes the $p \times p$ identity matrix and $\bar{d}_0$ and $\bar{d}_1$ are two positive constants.

Assumption 2. The desired trajectory $y_{di}(t)$, $i = 1, \dots, p$, is a known function of time that remains bounded, with its derivatives also being bounded and known. Additionally, $y_{di}(t)$ is assumed to be twice differentiable.

Remark 3. It is worth noting that Assumption 1 serves as a sufficient condition to guarantee the regularity of the matrix $D(X)$, thereby enabling the feedback linearization of system (13) through a static state feedback.

To begin the development of the control law $u(t)$, we define the tracking errors as follows:

$$\begin{aligned} e_1(t) &= y_{d1}(t) - y_1(t), \\ &\vdots \\ e_p(t) &= y_{dp}(t) - y_p(t) \end{aligned} \quad (14)$$

and we consider the following filtered tracking errors:

$$\begin{aligned} v_1(t) &= \left(\frac{d}{dt} + \lambda_1 \right) e_1(t), \quad \lambda_1 > 0, \\ &\vdots \\ v_p(t) &= \left(\frac{d}{dt} + \lambda_p \right) e_p(t), \quad \lambda_p > 0 \end{aligned} \quad (15)$$

In (15), when $v_i(t) = 0$, it corresponds to a linear differential equation, and its solution indicates that the tracking error $e_i(t)$ tends to zero at a time constant of $1/\lambda_i$. Moreover, the derivative $\dot{e}_i(t)$ of the error signal $e_i(t)$ also tends to zero [35–37]. Thus, the control objective shifts to designing a controller that maintains $v_i(t)$, $i = 1, \dots, p$, at zero. Consequently, the fundamental challenge of stabilizing the vector $[e_i(t), \dot{e}_i(t)]^T$ is reduced to maintaining the scalar $v_i(t)$ at zero. Furthermore, the bounds on $v_i(t)$ can be directly converted into

bounds on the signal errors $e_i(t)$. In particular, when $|v_i(t)| \leq Q_i$ with Q_i is a positive constant, it can be inferred that [35] $|e_i^{(j)}(t)| \leq 2^j \lambda_i^{j-1} Q_i$, $j = 0, 1$, $i = 1, \dots, p$. Increasing the parameter λ_i can lead to a minimization of the bounds Q_i .

The derivative with respect to time of the filtered errors (15) can be expressed as

$$\begin{aligned} \dot{v}_1 &= \Lambda_1 - f_1(X) - d_1(X) u_1, \\ &\vdots \\ \dot{v}_p &= \Lambda_p - f_p(X) - d_p(X) u_p \end{aligned} \quad (16)$$

with $\beta_1, \dots, \beta_p$ being given by

$$\begin{aligned} \beta_1 &= \ddot{y}_{d1} + \lambda_1 \dot{e}_1, \\ &\vdots \\ \beta_p &= \ddot{y}_{dp} + \lambda_p \dot{e}_p \end{aligned} \quad (17)$$

Let us denote $v = [v_1, \dots, v_p]^T$ and $\beta = [\beta_1, \dots, \beta_p]^T$. As a result, Equation (16) can be written as follows:

$$\dot{v} = \beta - F(X) - D(X)u \quad (18)$$

Given knowledge of the nonlinear functions $F(X)$ and $D(X)$, the control objective can be attained by employing the subsequent ideal control law:

$$u^* = D^{-1}(X)(-F(X) + \beta + Kv + K_0 \tanh(v/\varepsilon_0)) \quad (19)$$

with $K = \text{diag}(k_1, \dots, k_p)$ and $K_0 = \text{diag}(k_{01}, \dots, k_{0p})$, such that $k_{0i} > 0$ and $k_i > 0$ for $i = 1, \dots, p$, ε_0 represents a small positive constant and $\tanh(\cdot)$ stands for the hyperbolic tangent function, which is defined for the vector $v = [v_1, \dots, v_p]^T$ by

$$\tanh(v/\varepsilon_0) = [\tanh(v_1/\varepsilon_0), \dots, \tanh(v_p/\varepsilon_0)]^T \quad (20)$$

When we adopt the control input as $u = u^*$, the dynamics (18) is streamlined to

$$\dot{v} = -Kv - K_0 \tanh(v/\varepsilon_0) \quad (21)$$

or, equivalently,

$$\dot{v}_i = -k_i v_i - k_{0i} \tanh(v_i/\varepsilon_0), \quad i = 1, \dots, p \quad (22)$$

This signifies that $v_i(t) \rightarrow 0$ as $t \rightarrow \infty$, thus leading to the convergence of $e_i(t)$ and $\dot{e}_i(t)$ to zero [30].

Based on the preceding analysis, obtaining the ideal controller (19) is straightforward when the nonlinear terms $F(X)$ and $D(X)$ are known.

Nevertheless, as these terms are assumed to be unknown in this paper, implementing the ideal controller (19) becomes infeasible. Thus, in this study, we suggest designing an adaptive PID-like control system to estimate this unknown ideal controller.

3.2. Adaptive PID-like Control Design

In the preceding subsection, we demonstrated the existence of an optimal control law u^* , as defined by Equation (19), capable of fulfilling control objectives. Nevertheless, this optimal controller remains impractical due to its reliance on unknown functions. To address this issue, in this subsection, we introduce an adaptive PID-like controller as a solution. The considered adaptive PID-like controller aims to estimate the ideal controller, mitigating the impact of unknown functions. The discrepancy between the adaptive PID-like controller and the ideal controller will serve as the basis for updating the free parameters of the PID-like controller.

3.2.1. Control Law

To establish the control law, we presume that each element of the optimal input control vector $u^* = [u_1^*, \dots, u_p^*]^T$ can be estimated using a PID-like controller u_{pidl_i} , $i = 1, \dots, p$, characterized by the following general form:

$$u_{pidl_i} = K_{p_i} e_i(t) + K_{I_i} \int_0^t e_i(\tau) d\tau + K_{d_i} \frac{de_i(t)}{dt} + K_{c_i} \quad (23)$$

where K_{p_i} denotes the proportional gain, K_{I_i} represents the integral gain, K_{d_i} is the derivative gain, and K_{c_i} stands for an unknown control gain independent of the tracking error.

For convenience, let $\Theta_i = [K_{p_i}, K_{I_i}, K_{d_i}, K_{c_i}]^T$ and $\Lambda_i(e_i) = \left[e_i(t), \int_0^t e_i(\tau) d\tau, \frac{de_i(t)}{dt}, 1 \right]^T$.

Thus, we can express Equation (23) as

$$u_{PIDL_i}(e_i, \Theta) = \Lambda_i^T(e_i) \Theta_i \quad (24)$$

Additionally, we hypothesize the presence of an optimal time-varying parameter vector Θ_i^* , which possesses a bounded time derivative, ensuring that the ideal control u_i^* satisfies

$$u_i^* = \Lambda_i^T(e_i) \Theta_i^* + \varepsilon_i(X) \quad (25)$$

Here, $\varepsilon_i(X)$ represents the estimation error, while Θ_i^* denotes an unknown ideal parameter vector that optimizes the term $|\varepsilon_i(X)|$.

Let us define

$$\varepsilon(X) = [\varepsilon_1(X), \dots, \varepsilon_p(X)]^T, \Theta^* = [\Theta_1^{*T}, \dots, \Theta_p^{*T}]^T \text{ and } \Lambda(e) = \text{diag}(\Lambda_1(e_1), \dots, \Lambda_p(e_p))$$

Thus, we can express u^* by

$$u^* = \Lambda^T(e) \Theta^* + \varepsilon(X) \quad (26)$$

Prior to proceeding, it is necessary to establish an assumption concerning the estimation error $\varepsilon(X)$.

Assumption 3. The estimation error $\varepsilon(X)$ in (26) is constrained by the following bounds:

$$\varepsilon^T(X) D(X) \varepsilon(X) \leq \tilde{\varepsilon}_0 v^T D(X) v + \tilde{\varepsilon}_1 \quad (27)$$

where $\tilde{\varepsilon}_0$ and $\tilde{\varepsilon}_1$ are two positive constants.

Since the ideal parameter vector Θ^* is unknown, it needs to be estimated using an appropriate adaptation law. Let Θ represent an approximation of the ideal vector Θ^* and define the control law as the adaptive PID-like approximation of the ideal controller (26). In other words, we adopt a controller for system (13) as

$$u = u_{PIDL} = \Lambda^T(e) \Theta \quad (28)$$

Following the determination of the controller structure, the subsequent stage involves devising an adaptive scheme for the free parameters Θ to ensure that the control law u closely estimates the ideal controller u^* . To achieve this objective, a gradient descent adaptation algorithm will be introduced in the upcoming subsection.

Remark 4. We selected the control law (23) with the unknown gain K_{c_i} to enhance the estimation capabilities of the ideal controller (19). The gain K_{c_i} serves to approximate the unknown constant quantity within the ideal controller, thereby improving the overall accuracy and performance of the adaptive control system. Additionally, this can be beneficial in applications where a base level of control action is desired regardless of the tracking error.

3.2.2. Adaptation Law for PID-like Control

The objective at this phase is to formulate an adaptive law for estimating the parameters Θ , enabling the PID-like controller (28) to accurately approximate the unknown ideal controller (26). The adaptive law must be formulated to reduce the discrepancy between u^* and u . In addition, the adaptive law must guarantee that the parameter estimates remain bounded. To accomplish this, we express the discrepancy between u^* and u in the following manner:

$$e_d = u^* - u \quad (29)$$

The term e_d stands for the actual discrepancy between the ideal control u^* and the control input u_{PIDL} .

From (26) and (28), (29) transforms into

$$e_d = u^* - \Lambda^T(e)\Theta = \Lambda^T(e)\tilde{\Theta} + \varepsilon_i(X) \quad (30)$$

where $\tilde{\Theta} = \Theta^* - \Theta$ denotes the parameter estimation error vector.

The derivation of the error equation, which governs the closed-loop system, is achieved by manipulating the right-hand side of Equation (18) with the addition and subtraction of $D(X)u^*$, as follows:

$$\dot{v} = \Lambda - f(X) - D(X)u + D(X)u^* - D(X)u^* \quad (31)$$

Using (19) and (30), (31) is simplified as follows:

$$\dot{v} = -Kv - K_0 \tanh(v/\varepsilon_0) + D(X)e_d \quad (32)$$

Now, let us introduce a quadratic cost function that quantifies the difference between the ideal controller u^* and the implemented PID-like controller u_{PIDL} , outlined as

$$J(\Theta) = \frac{1}{2} e_d^T D(X) e_d = \frac{1}{2} (u^* - \Lambda^T(e)\Theta)^T D(X) (u^* - \Lambda^T(e)\Theta) \quad (33)$$

The gradient descent technique is adopted in this work to reduce the cost function (33). Therefore, by employing the gradient descent method [35–37], the subsequent first-order differential equation is established as an adaptive law governing the parameters Θ :

$$\dot{\Theta} = -\eta \nabla_{\Theta} J(\Theta) \quad (34)$$

where η represents a positive and constant design parameter.

The gradient of $J(\Theta)$ with respect to Θ can be derived from (33) as follows:

$$\nabla_{\Theta} J(\Theta) = \frac{\partial J(\Theta)}{\partial \Theta} = -\Lambda(e) D(X) e_d \quad (35)$$

Hence, the gradient descent algorithm can be expressed by

$$\dot{\Theta} = \eta \Lambda(e) D(X) e_d \quad (36)$$

It is worth noting that the error signal e_d defined in (29) is unavailable because the ideal controller u^* is unknown. To address this challenge, we will use Equation (32). In fact, Equation (32) demonstrates that despite the unavailability of the error vector e_d , the vector $D(X)e_d$ is accessible, and it is determined by

$$D(X)e_d = \dot{v} + Kv + K_0 \tanh(v/\varepsilon_0) \quad (37)$$

Thus, based on (37), (36) becomes

$$\dot{\Theta} = \eta \Lambda(e) (\dot{v} + Kv + K_0 \tanh(v/\varepsilon_0)) \quad (38)$$

As demonstrated in [38], the adaptive law (38) lacks the capability to ensure that the parameters $\tilde{\Theta}$ remain bounded despite the unavoidable estimation errors that are typically present in adaptive methods. To enhance the robustness of the adaptive law (38) against such approximation errors, we introduce a σ -modification term as follows [38]:

$$\dot{\Theta} = \eta \Lambda(e)(\dot{v} + Kv + K_0 \tanh(v/\varepsilon_0)) - \eta \sigma \Theta \quad (39)$$

where σ represents a small positive constant.

The overall structure of the proposed adaptive PID-like controller is illustrated in the block diagram shown in Figure 3.

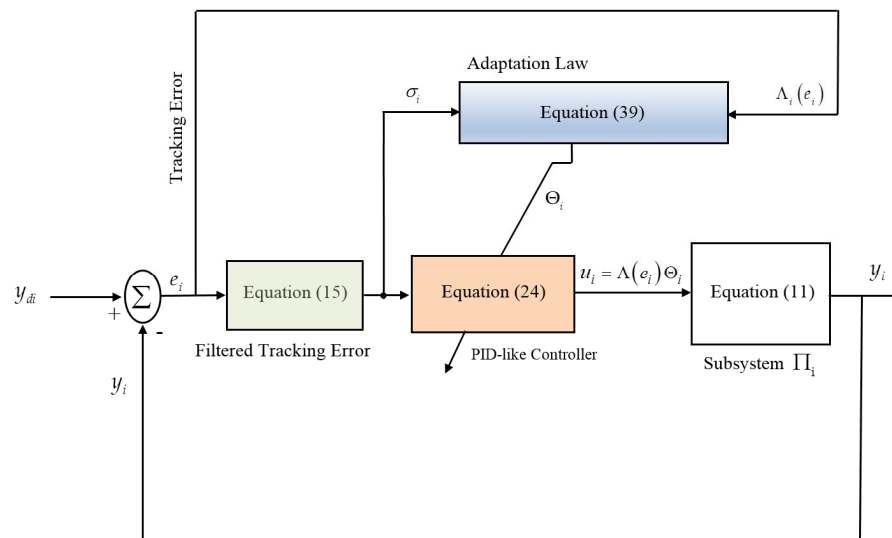


Figure 3. Block diagram of the designed robust adaptive PID-like controller.

3.3. Stability Analysis of the Closed-Loop System

For the purpose of investigating the convergence of the tracking errors $e_i(t)$ and the stability of the closed-loop system, we will make use of the subsequent Lyapunov-like function.

$$V = \frac{1}{2} v^T v + \frac{1}{2\eta} \tilde{\Theta}^T \tilde{\Theta} \quad (40)$$

The derivative with respect to time of Equation (40) can be expressed as

$$\dot{V} = v^T \dot{v} - \frac{1}{\eta} \tilde{\Theta}^T \dot{\tilde{\Theta}} + \frac{1}{\eta} \tilde{\Theta}^T \dot{\tilde{\Theta}}^* \quad (41)$$

By employing (32), (36), and (39), Equation (41) transforms into

$$\dot{V} = v^T (-Kv - K_0 \tanh(v/\varepsilon_0) + D(X)e_d) - \tilde{\Theta}^T (\Lambda(e)D(X)e_d - \sigma \Theta) + \frac{1}{\eta} \tilde{\Theta}^T \dot{\tilde{\Theta}}^* \quad (42)$$

Given (30), we can express (42) by

$$\dot{V} = -v^T Kv - v^T K_0 \tanh(v/\varepsilon_0) + v^T D(X)e_d - (e_d^T - \varepsilon^T(X))D(X)e_d + \sigma \tilde{\Theta}^T \Theta + \frac{1}{\eta} \tilde{\Theta}^T \dot{\tilde{\Theta}}^* \quad (43)$$

or

$$\dot{V} = -v^T Kv - v^T K_0 \tanh(v/\varepsilon_0) + v^T D(X)e_d - e_d^T D(X)e_d + \varepsilon^T(X)D(X)e_d + \sigma \tilde{\Theta}^T \Theta + \frac{1}{\eta} \tilde{\Theta}^T \dot{\tilde{\Theta}}^* \quad (44)$$

We utilize the following inequalities:

$$\sigma \tilde{\Theta}^T \Theta \leq -\frac{\sigma}{2} \|\tilde{\Theta}\|^2 + \frac{\sigma}{2} \|\Theta^*\|^2 \quad (45)$$

$$\varepsilon^T(X)D(X)e_d \leq \frac{1}{4}e_d^T D(X)e_d + \varepsilon^T(X)D(X)\varepsilon(X) \quad (46)$$

$$\frac{1}{\eta} \tilde{\Theta}^T \dot{\Theta}^* \leq \frac{\sigma}{4} \|\tilde{\Theta}\|^2 + \frac{1}{\sigma\eta^2} \|\dot{\Theta}^*\|^2 \quad (47)$$

$$v^T D(X)e_d \leq \frac{1}{4}e_d^T D(X)e_d + v^T D(X)v \quad (48)$$

We can establish a bound for (44) as follows:

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2}e_d^T D(X)e_d - v^T K_0 \tanh(v/\varepsilon_0) - v^T (K - D(X))v - \frac{\sigma}{4} \|\tilde{\Theta}\|^2 + \frac{\sigma}{2} \|\Theta^*\|^2 + \frac{1}{\sigma\eta^2} \|\dot{\Theta}^*\|^2 \\ & + \varepsilon^T(X)D(X)\varepsilon(X) \end{aligned} \quad (49)$$

Applying inequality (27), we obtain

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2}e_d^T D(X)e_d - v^T K_0 \tanh(v/\varepsilon_0) - v^T (K - (1 + \tilde{\varepsilon}_0)D(X))v - \frac{\sigma}{4} \|\tilde{\Theta}\|^2 + \frac{\sigma}{2} \|\Theta^*\|^2 \\ & + \frac{1}{\sigma\eta^2} \|\dot{\Theta}^*\|^2 + \tilde{\varepsilon}_1 \end{aligned} \quad (50)$$

As this paper assumes that the parameters Θ^* and its time derivative $\dot{\Theta}^*$, along with the functions $\varepsilon(X)$ and $D(X)$, are bounded, we are able to establish a positive constant bound, referred to as Δ , such that

$$\Delta = \sup_t \left(\frac{\sigma}{2} \|\Theta^*(t)\|^2 + \frac{1}{\sigma\eta^2} \|\dot{\Theta}^*(t)\|^2 + \tilde{\varepsilon}_1 \right) \quad (51)$$

Suppose the design parameter k_i is selected to yield $k_i > (1 + \tilde{\varepsilon}_0)\bar{g}_1$, $i = 1, \dots, p$, and $\alpha = \min(2\min_{1 \leq i \leq p}(k_i - (1 + \tilde{\varepsilon}_0)\bar{g}_1), 0.5\sigma\eta)$. Then, inequality (50) can be expressed as follows:

$$\dot{V} \leq -\frac{1}{2}e_d^T D(X)e_d - v^T K_0 \tanh(v/\varepsilon_0) - \frac{\alpha}{2} v^T v - \frac{\alpha}{2\eta} \|\tilde{\Theta}\|^2 + \Delta \quad (52)$$

or

$$\dot{V} \leq -\alpha V + \Delta \quad (53)$$

Now, we are able to establish the subsequent theorem, which demonstrates the boundedness of all variables within the closed-loop system.

Theorem 1. *Given system (13) and assuming that Assumptions 1 and 2 hold and the design parameter k_i is selected such that $k_i > (1 + \tilde{\varepsilon}_0)\bar{d}_1$, $i = 1, \dots, p$, the controller given by (27), along with the adaptation law specified by Equation (39), ensures that the closed-loop system exhibits UUB stability, and the output tracking errors $e_i(t)$ converge to a small vicinity around the origin.*

Proof of Theorem 1. Based on Equation (53), it results in

$$V(t) \leq V(0)e^{-\alpha t} + \frac{\Delta}{\alpha} \quad (54)$$

Thus, using (54), it can be demonstrated that for $V \geq \Delta/\alpha$, we have $\dot{V} < 0$. As per a conventional Lyapunov theorem, the signals $v(t)$, $\tilde{\Theta}(t)$, and $u(t)$ within the closed-loop system remain bounded. Additionally, by utilizing (40) and (54), we can establish that

$\|v(t)\| \leq \sqrt{\|v(0)\|^2 + \frac{1}{\eta} \|\tilde{\Theta}(0)\|^2} e^{-0.5\alpha t} + \sqrt{\frac{2\Delta}{\alpha}}$. To ensure the convergence of tracking

error to a small vicinity near zero, the design parameters k_i , σ , and η should be selected suitably. Consequently, it becomes feasible to minimize $\sqrt{\frac{2\Delta}{\alpha}}$ to any desired extent by denoting $Q = \sqrt{\frac{2\Delta}{\alpha}}$ and given that $e^{-0.5\alpha t} \rightarrow 0$ for $t \rightarrow \infty$. Thus, there exists T such that $\|v(t)\| \leq Q$ for $t > T$, which implies that the tracking errors converge towards residual sets in the following manner: $\|e_i^{(j)}(t)\| \leq 2^j \lambda_i^{j-1} Q$, $j = 0, 1$, $i = 1, \dots, p$. $\square$

4. Application of the Robust Adaptive PID-like Controller

At this stage, we demonstrate the application of the suggested adaptive PID-like controller to control both the rotational and translational subsystems of a quadrotor UAV system. The primary aim of this control is to guarantee the monitoring of the specified positions (x_d, y_d, z_d, ψ_d) . To achieve this, we have designed six adaptive PID-like controllers.

4.1. Controller Design for the Rotation Subsystem

In the rotation subsystem, achieving the convergence of angular positions (ϕ, θ, ψ) towards the desired trajectory $(\phi_d, \theta_d, \psi_d)$ is possible through the use of the control inputs (u_2, u_3, u_4) . This subsection details the application of our control methodology outlined in Section 3 for governing the roll, pitch, and yaw angles. To start, the tracking errors, denoted as $(e_\phi, e_\theta, e_\psi)$, and the filtering errors, represented as $(v_\phi(t), v_\theta(t), v_\psi(t))$, are defined in the following manner:

$$\begin{cases} v_\phi(t) = \dot{e}_\phi(t) + \lambda_\phi e_\phi(t), & e_\phi = \phi - \phi_d \\ v_\theta(t) = \dot{e}_\theta(t) + \lambda_\theta e_\theta(t), & e_\theta = \theta - \theta_d \\ v_\psi(t) = \dot{e}_\psi(t) + \lambda_\psi e_\psi(t), & e_\psi = \psi - \psi_d \end{cases} \quad (55)$$

The inputs (u_2, u_3, u_4) are selected as the outputs of three PID-like controllers expressed as

$$\begin{cases} u_2 = K_{p_\phi} e_\phi(t) + K_{I_\phi} \int_0^t e_\phi(\tau) d\tau + K_{d_\phi} \frac{de_\phi(t)}{dt} + K_{c_\phi} \\ u_3 = K_{p_\theta} e_\theta(t) + K_{I_\theta} \int_0^t e_\theta(\tau) d\tau + K_{d_\theta} \frac{de_\theta(t)}{dt} + K_{c_\theta} \\ u_4 = K_{p_\psi} e_\psi(t) + K_{I_\psi} \int_0^t e_\psi(\tau) d\tau + K_{d_\psi} \frac{de_\psi(t)}{dt} + K_{c_\psi} \end{cases} \quad (56)$$

which are continuously updated online using the subsequent parameter adaptation laws:

$$\begin{cases} \Theta_\phi = [K_{p_\phi}, K_{I_\phi}, K_{d_\phi}, K_{c_\phi}], & \dot{\Theta}_\phi = \eta \Lambda(e_\phi) (\dot{v}_\phi + k_\phi v_\phi + k_{0_\phi} \tanh(v_\phi/\varepsilon_0)) - \eta \sigma \Theta_\phi \\ \Theta_\theta = [K_{p_\theta}, K_{I_\theta}, K_{d_\theta}, K_{c_\theta}], & \dot{\Theta}_\theta = \eta \Lambda(e_\theta) (\dot{v}_\theta + k_\theta v_\theta + k_{0_\theta} \tanh(v_\theta/\varepsilon_0)) - \eta \sigma \Theta_\theta \\ \Theta_\psi = [K_{p_\psi}, K_{I_\psi}, K_{d_\psi}, K_{c_\psi}], & \dot{\Theta}_\psi = \eta \Lambda(e_\psi) (\dot{v}_\psi + k_\psi v_\psi + k_{0_\psi} \tanh(v_\psi/\varepsilon_0)) - \eta \sigma \Theta_\psi \end{cases} \quad (57)$$

As previously shown, the PID-like controllers defined by (56) and adjusted using the adaptive laws (57) ensure the convergence of the angles (ϕ, θ, ψ) to their respective desired values $(\phi_d, \theta_d, \psi_d)$.

4.2. Controller Design for the Translation Subsystem

The three inputs (u_1, u_r, u_p) pertaining to the translation subsystem can also be synthesized in a similar manner. To successfully control the altitude z of the quadrotor, the input u_1 is determined as the output of an adaptive PID-like altitude controller. Additionally, to achieve desired translation motions (x_d, y_d) , the values of (u_r, u_p) utilized in computing the desired pitch and roll angles are individually derived from adaptive PID-like controllers. To accomplish this, the tracking errors (e_x, e_y, e_z) and the filtering errors $(v_x(t), v_y(t), v_z(t))$ are defined as follows:

$$\begin{cases} v_x(t) = \dot{e}_x(t) + \lambda_x e_x(t), & e_x = x - x_d \\ v_y(t) = \dot{e}_y(t) + \lambda_y e_y(t), & e_y = y - y_d \\ v_z(t) = \dot{e}_z(t) + \lambda_z e_z(t), & e_z = z - z_d \end{cases} \quad (58)$$

and the computation of the control inputs (u_1, u_r, u_p) is performed through

$$\begin{cases} u_1 = K_{p_z} e_z(t) + K_{I_z} \int_0^t e_z(\tau) d\tau + K_{d_z} \frac{de_z(t)}{dt} + K_{c_z} \\ u_r = K_{p_x} e_x(t) + K_{I_x} \int_0^t e_x(\tau) d\tau + K_{d_x} \frac{de_x(t)}{dt} + K_{c_x} \\ u_p = K_{p_y} e_y(t) + K_{I_y} \int_0^t e_y(\tau) d\tau + K_{d_y} \frac{de_y(t)}{dt} + K_{c_y} \end{cases} \quad (59)$$

where the gains in (61) are dynamically adjusted online in the following manner:

$$\begin{cases} \Theta_z = [K_{p_z}, K_{I_z}, K_{d_z}, K_{c_z}], & \dot{\Theta}_z = \eta \Lambda(e_z) (\dot{v}_z + k_z v_z + k_{0_z} \tanh(v_z/\varepsilon_0)) - \eta \sigma \Theta_z \\ \Theta_x = [K_{p_x}, K_{I_x}, K_{d_x}, K_{c_x}], & \dot{\Theta}_x = \eta \Lambda(e_x) (\dot{v}_x + k_x v_x + k_{0_x} \tanh(v_x/\varepsilon_0)) - \eta \sigma \Theta_x \\ \Theta_y = [K_{p_y}, K_{I_y}, K_{d_y}, K_{c_y}], & \dot{\Theta}_y = \eta \Lambda(e_y) (\dot{v}_y + k_y v_y + k_{0_y} \tanh(v_y/\varepsilon_0)) - \eta \sigma \Theta_y \end{cases} \quad (60)$$

The PID-like controllers described in (59) and adapted with the dynamic laws (60) guarantee that the positions (x, y, z) converge to their corresponding desired positions (x, y, z) .

4.3. Simulation Results and Discussions

In this part, we examine the efficiency of the proposed stable adaptive PID-like controller for a quadrotor UAV through a simulation study. Figure 2 illustrates the control structure, and the quadrotor parameters used are as follows [34]: $m = 0.5$ kg, $l = 0.24$ m, $I_x = I_y = 3.8 \times 10^{-3}$ Nm.s²/rad, $I_z = 7.1 \times 10^{-3}$ Nm.s²/rad, $I_R = 2.8 \times 10^{-5}$ Nm.s²/rad, $b = 3.0 \times 10^{-5}$ N.s²/rad², $c = 3.2 \times 10^{-7}$ Nm.s²/rad², $K_t = \text{diag}(3.2, 3.2, 4.8) \times 10^{-2}$ N.s/m, $K_r = \text{diag}(5.6, 5.6, 6.4) \times 10^{-4}$ Nm.s/rad, and $g = 9.81$ m/s².

It is important to note that in this simulation, the parameters of the quadrotor UAV are assumed to be entirely unknown, meaning the control system does not rely on knowing the system's model. The quadrotor UAV system's dynamic model is needed strictly for simulation purposes.

The simulation results discussed in this paper pertain to the path tracking issue. A vertical helix serves as the reference trajectory and is defined as $x_d = 0.5 \cos(0.5t)$, $y_d = 0.5 \sin(0.5t)$, $z_d = 1 + \frac{t}{10}$, and $\psi_d = \frac{\pi}{3}$. The initial conditions of the quadrotor UAV system are set as $(x, y, z) = (0, 0, 0.5)$ m and $(\phi, \theta, \psi) = (0, 0, 0.5)$ rad. The controller parameters used in the simulation are as follows: $\lambda_z = \lambda_\phi = \lambda_\theta = \lambda_\psi = 5$, $\lambda_x = \lambda_y = 7$, $k_z = k_\phi = k_\theta = k_\psi = k_x = k_y = 3$, $k_{0_z} = k_{0_x} = 0.1$, $k_{0_\phi} = k_{0_\theta} = 5$, $k_{0_\psi} = k_{0_y} = 1$, $\eta = 5$, $\sigma = 0.001$, and $\varepsilon_0 = 0.01$. In this simulation study, we set the initial values of the parameter estimates as follows: $\Theta_\phi(0) = \Theta_\theta(0) = \Theta_\psi(0) = \Theta_x(0) = \Theta_z(0) = [0, 0, 0, 0]^T$ and $\Theta_z(0) = [0.01, 0, 0, 0]^T$. To evaluate the robustness of the introduced adaptive PID-like controller under external disturbances, an external disturbance $D\pi_i(t) = 0.01(10 + 5 \sin(2\pi t))$, $i = 1, \dots, 6$, is introduced for each subsystem.

Figures 4–11 depict the simulation outcomes regarding the path tracking of the quadrotor UAV. It is evident from Figure 4, Figure 5, and Figure 8 that the actual trajectories (x, y, z, ψ) converge to (x_d, y_d, z_d, ψ_d) , demonstrating the controller's ability to achieve accurate tracking with a short transient response. The positions z , x , y , and the yaw angle ψ converge rapidly to their reference trajectories, with response times of less than 2 s. Additionally, the roll and pitch angles (ϕ and θ) exhibit even quicker responses, with convergence times of less than 0.05 s. Additionally, as illustrated by Figures 6 and 7, the control signals exhibit smoothness and practical feasibility, while the parameter estimates shown in Figures 9–11 remain within bounds. The bounded nature of these parameters

ensures reliable system performance without risk of divergence, even in the presence of potential model uncertainties or disturbances. The absence of excessive oscillations or abrupt changes in the control signals further highlights the controller's efficiency. These simulation findings effectively showcase the tracking proficiency of the proposed adaptive PID-like controller within the quadrotor UAV control domain.

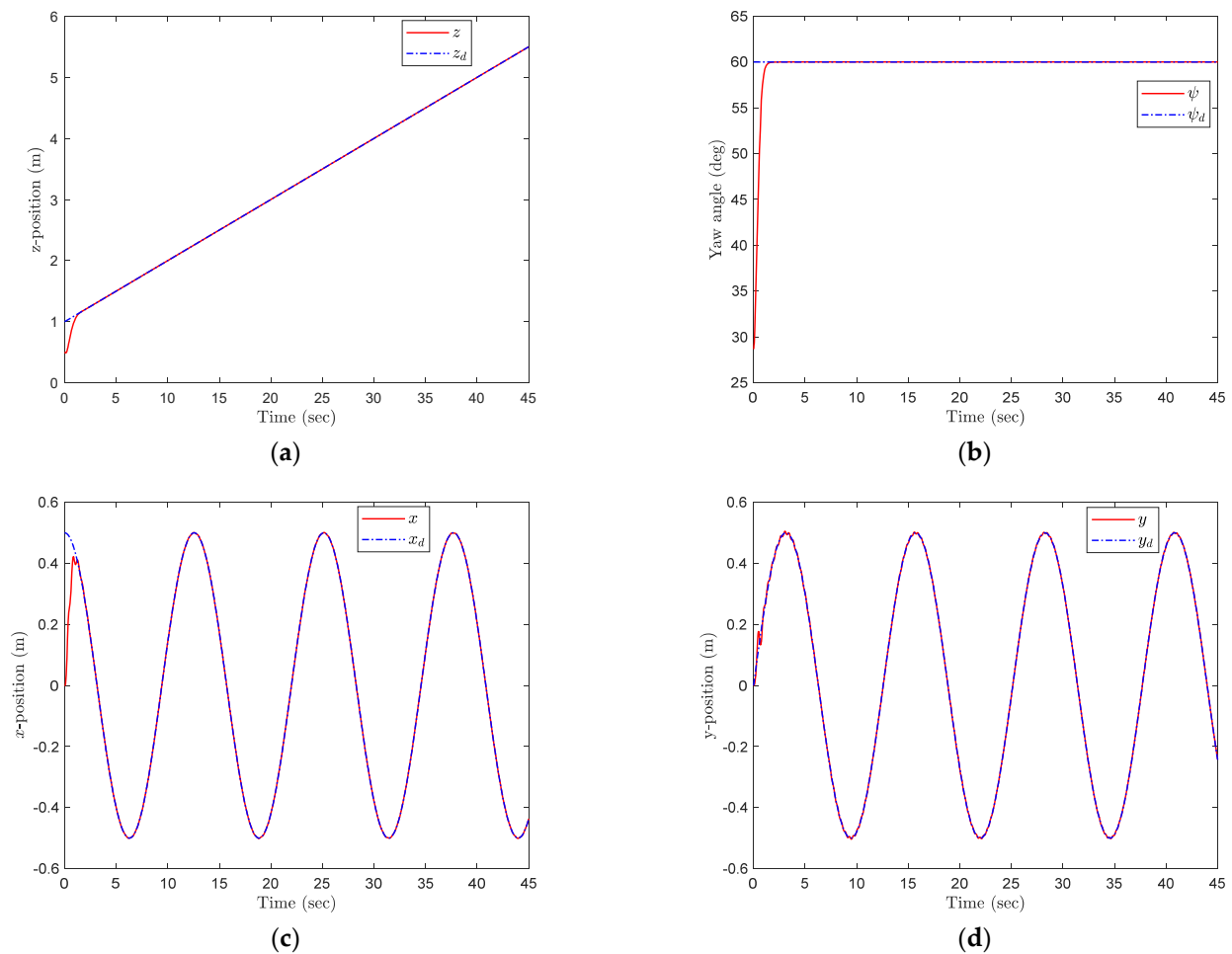


Figure 4. Actual positions (x, y, z, ψ) and desired positions (x_d, y_d, z_d, ψ_d): (a) z-position; (b) yaw angle; (c) x-position; (d) y-position.

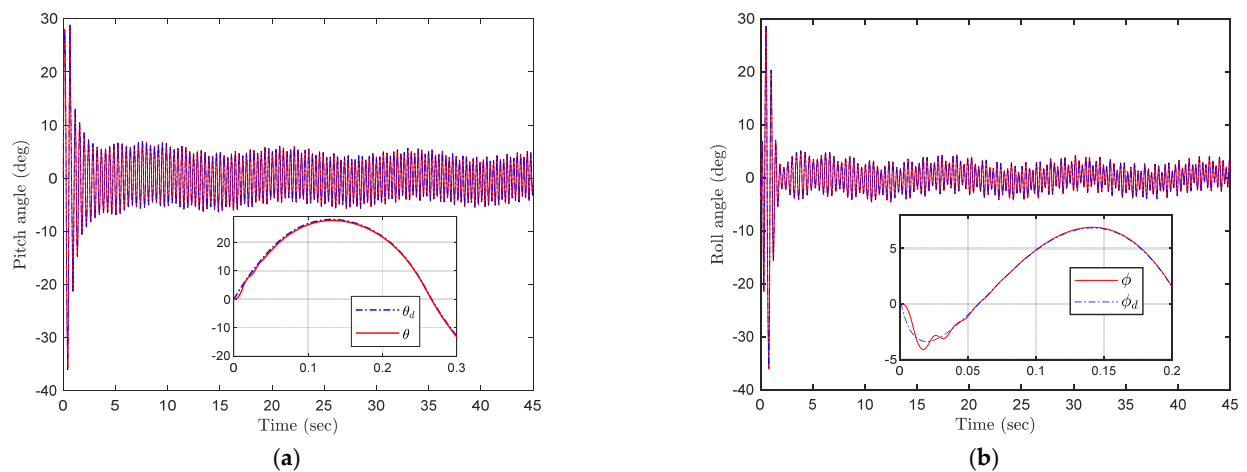


Figure 5. Actual angles (ϕ, θ) and desired angles (ϕ_d, θ_d): (a) pitch angle; (b) roll angle.

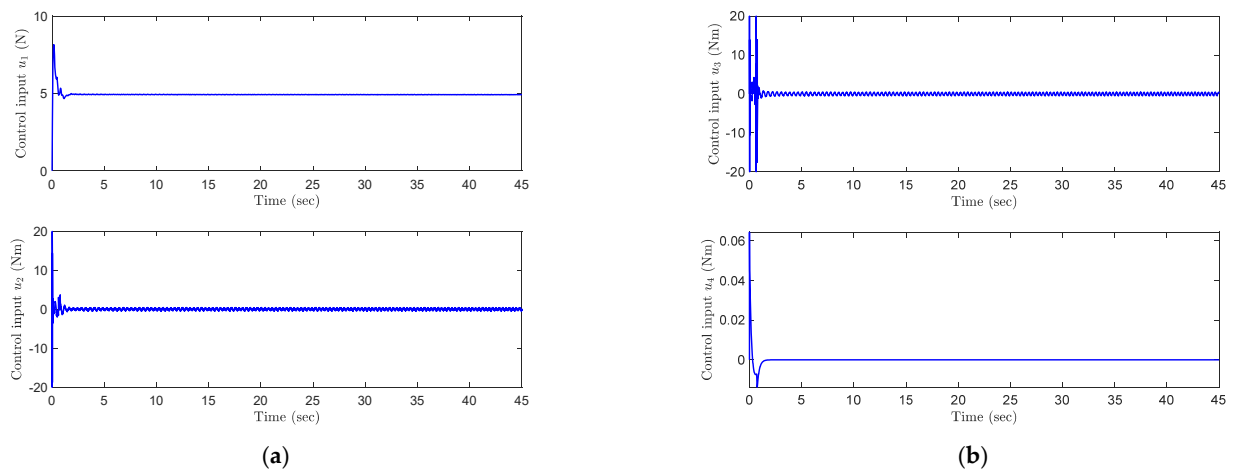


Figure 6. Applied controls: (a) control inputs (u_1, u_2); (b) control inputs (u_3, u_4).

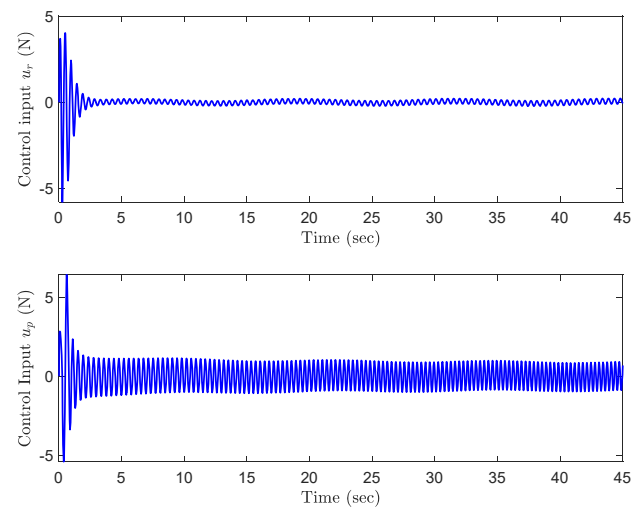


Figure 7. Control inputs (u_r, u_p).

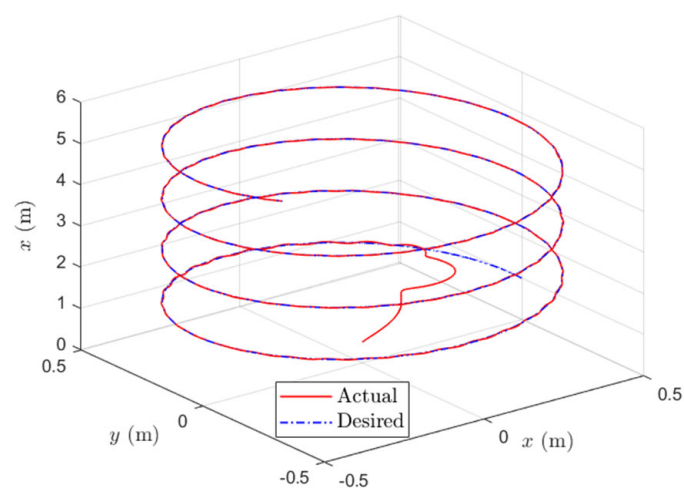


Figure 8. Three-dimensional flight path of the quadrotor.

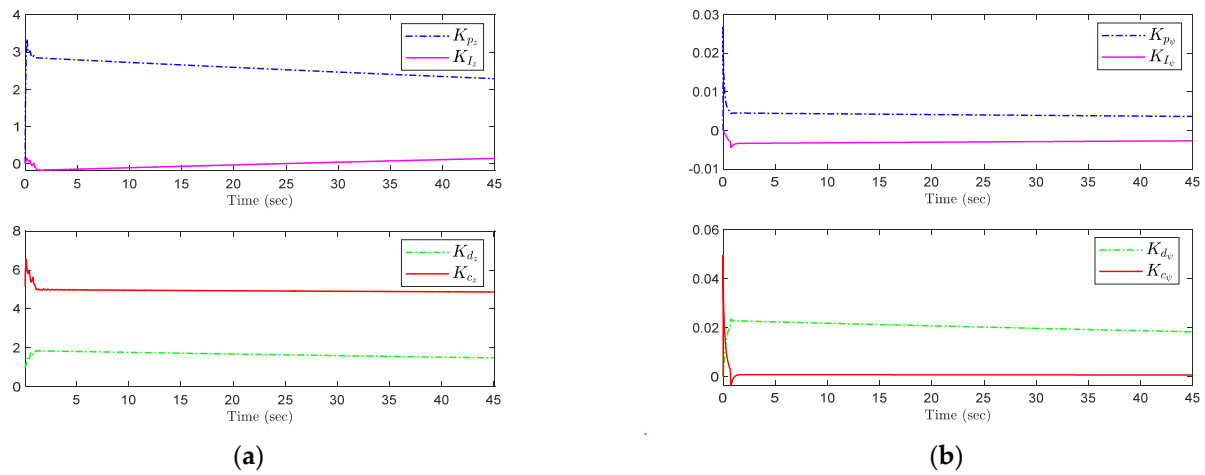


Figure 9. Time evolution of the estimated gains: (a) the gain vector Θ_z ; (b) the gain vector Θ_ψ .

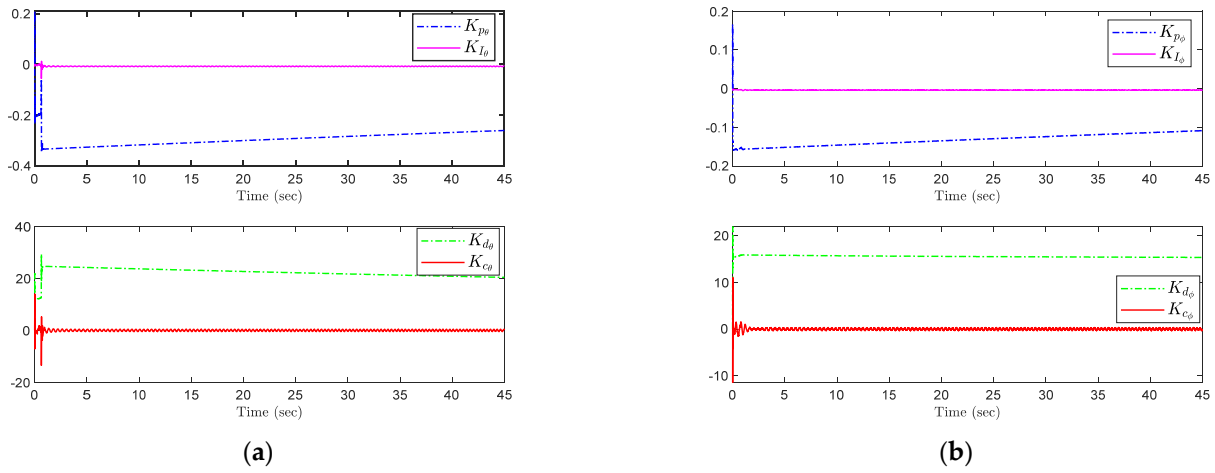


Figure 10. Time evolution of the estimated gains: (a) the gain vector Θ_θ ; (b) the gain vector Θ_ϕ .

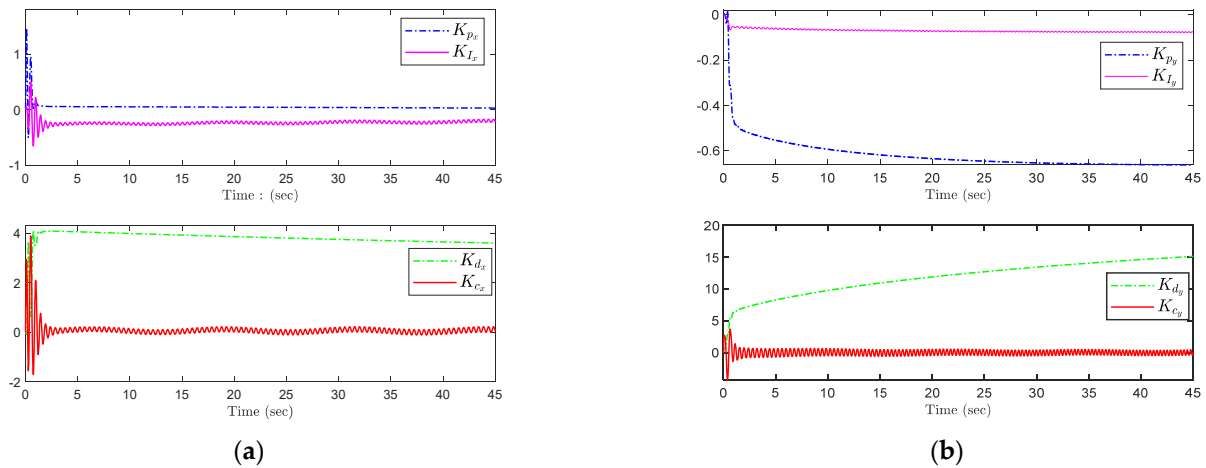


Figure 11. Time evolution of the estimated gains: (a) the gain vector Θ_x ; (b) the gain vector Θ_y .

Figure 12 illustrates the 3D flight trajectories of the quadrotor resulting from a comparative study between the robust adaptive PID-like controller and a PID controller with constant gains. The parameters of the PID controller are listed in Table 1; these gains were obtained through trial-and-error and are the ones that provided the best performance. The performance evaluation was based on the key characteristics of the tracking errors over the

time interval $t \in [5, 45]$ s. To assess tracking accuracy, we analyzed the average absolute values of the errors, along with their standard deviations and ranges, over the test duration. Clearly, the results in Table 2 and the curves in Figure 12 demonstrate that the robust adaptive PID-like controller provides significantly better tracking performance compared to the PID controller.

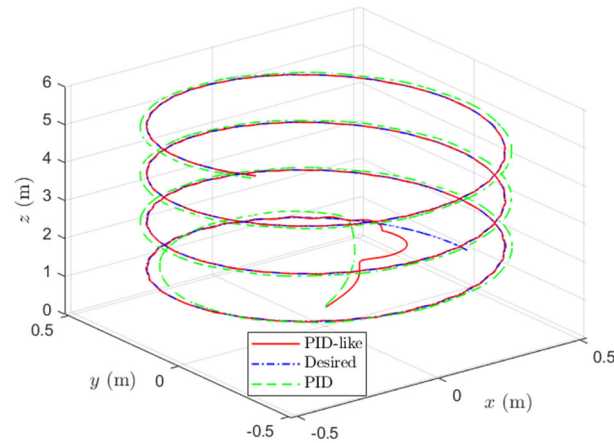


Figure 12. Three-dimensional flight trajectories of the quadrotor resulting from the comparative study.

Table 1. PID controller gains.

Gains	x	y	z	ϕ	θ	ψ
K_p	3	4.5	37	0.5	0.2	0.8
K_i	1	1	15	1	1	0
K_d	3	2	24	1	0.5	0.06

Table 2. Comparison of tracking accuracy between the proposed robust adaptive PID-like controller and a PID controller.

Controllers		x (m)	y (m)	z (m)	ψ (deg)
PID-like Controller	Avg($ e_i $)	7.3×10^{-4}	1.9×10^{-3}	1.9×10^{-4}	1.8×10^{-5}
	Rge e_i	$(-18, 2) \times 10^{-4}$	$(-4.9, 5) \times 10^{-3}$	$(-3.2, 4.3) \times 10^{-4}$	$(-3, 3) \times 10^{-5}$
	Std e_i	8.6×10^{-4}	2.2×10^{-3}	2.1×10^{-4}	2×10^{-5}
PID Controller	Avg($ e_i $)	144×10^{-4}	8.9×10^{-3}	5.4×10^{-4}	5080×10^{-5}
	Rge e_i	$(-619, 235) \times 10^{-4}$	$(-21, 16) \times 10^{-3}$	$(-101, 1.9) \times 10^{-4}$	$(-8240, -1930) \times 10^{-5}$
	Std e_i	165×10^{-4}	10×10^{-3}	15×10^{-4}	2230×10^{-5}

5. Conclusions

This paper proposes a stable adaptive PID-like controller for quadrotor UAVs. The control scheme is designed for a general class of second-order nonlinear systems and applied to a quadrotor UAV system. The method includes an adaptive PID-like controller with an adaptive law, using the gradient descent technique to update parameters and minimize the error relative to the unknown ideal controller and the PID-like controller in use. This approach does not rely on knowledge of the system's mathematical model, ensures the boundedness of all signals in the closed-loop system, and guarantees the convergence of tracking errors to a neighbourhood of the origin. Simulations demonstrate the controller's effectiveness in guiding the quadrotor along a predetermined trajectory of Cartesian positions and yaw angle while maintaining overall system stability. Future

work will focus on enhancing this approach by integrating a state observer for state vector estimation.

Author Contributions: Conceptualization, A.B. and T.S.; methodology, A.B. and S.L.; software, A.B.; validation, A.B. and T.S.; formal analysis, A.B. and S.L.; investigation, A.B. and S.L.; resources, A.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, A.B. and T.S.; visualization, A.B. and J.F.W.; supervision, J.F.W. and T.S.; project administration, T.S. and J.F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Jha, A.R. *Theory, Design, and Applications of Unmanned Aerial Vehicles*; CRC Press: Boca Raton, FL, USA, 2016; ISBN 9781498715430.
2. Idrissi, M.; Salami, M.; Annaz, F. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design, and Control Algorithms. *J. Intell. Robot. Syst.* **2022**, *104*, 22. [\[CrossRef\]](#)
3. Hernández-González, O.; Targui, B.; Valencia-Palomo, G.; Guerrero-Sánchez, M.E. Robust Cascade Observer for a Disturbance Unmanned Aerial Vehicle Carrying a Load under Multiple Time-Varying Delays and Uncertainties. *Int. J. Syst. Sci.* **2024**, *55*, 1056–1072. [\[CrossRef\]](#)
4. Boubakir, A.; Labiod, S.; Boudjema, F.; Plestan, F. Model-Free Controller with an Observer Applied in Real-Time to a 3-DOF Helicopter. *Turk. J. Electr. Eng. Comput. Sci.* **2014**, *22*, 1564–1581. [\[CrossRef\]](#)
5. Kim, S.-J.; Suh, J.-H. Model-Free RBF Neural Network Intelligent-PID Control Applying Adaptive Robust Term for Quadrotor System. *Drones* **2024**, *8*, 179. [\[CrossRef\]](#)
6. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z. Real-Time Implementation of an Adaptive PID Controller for the Quadrotor MAV Embedded Flight Control System. *Aerospace* **2023**, *10*, 59. [\[CrossRef\]](#)
7. Huanca, C.P.E.Y.; Incremona, G.P.; Colaneri, P. Design of a Distributed Switching Model Predictive Control for Quadrotor UAVs Aggregation. *IEEE Control Syst. Lett.* **2023**, *7*, 2964–2969. [\[CrossRef\]](#)
8. Cowling, I.D.; Yakimenko, O.A.; Whidborne, J.F.; Cooke, A.K. A Prototype of an Autonomous Controller for a Quadrotor UAV. In Proceedings of the 2007 European Control Conference (ECC), Kos, Greece, 2–5 July 2007; pp. 4001–4008.
9. Sir Elkhateem, A.; Engin, S.N. Robust LQR and LQR-PI Control Strategies Based on Adaptive Weighting Matrix Selection for a UAV Position and Attitude Tracking Control. *Alex. Eng. J.* **2022**, *61*, 6275–6292. [\[CrossRef\]](#)
10. Baek, J.; Kang, M. A Synthesized Sliding-Mode Control for Attitude Trajectory Tracking of Quadrotor UAV Systems. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 2189–2199. [\[CrossRef\]](#)
11. Xu, D.; Whidborne, J.F.; Cooke, A. Fault tolerant control of a quadrotor using $\mathcal{L}_1$ adaptive control. *Int. J. Intell. Unmanned Syst.* **2016**, *4*, 43–66. [\[CrossRef\]](#)
12. Souanef, T.; Whidborne, J.; Boubakir, A. $\mathcal{L}_1$ Adaptive Control of Quadrotor UAVs in Case of Inversion of the Torque Direction. *Drones Auton. Veh.* **2023**, *1*, 10004. [\[CrossRef\]](#)
13. Qi, G.; Deng, J.; Li, X.; Yu, X. Compensation Function Observer-Based Model-Compensation Backstepping Control and Application in Anti-Inference of Quadrotor UAV. *Control. Eng. Pract.* **2023**, *140*, 105633. [\[CrossRef\]](#)
14. Kim, H.S.; Lee, K.; Joo, Y.H. Decentralized Sampled-Data Fuzzy Controller Design for a VTOL UAV. *J. Frankl. Inst.* **2021**, *358*, 1888–1914. [\[CrossRef\]](#)
15. Qin, X.; Zhao, Z.; Huang, P.; Li, J. Multiple Feedback Recurrent Neural Network Based Super-Twisting Predefined-Time Nonsingular Terminal Sliding Mode Control for Quad-Rotor UAV. *Aerosp. Sci. Technol.* **2024**, *151*, 109282. [\[CrossRef\]](#)
16. Pi, C.-H.; Hu, K.-C.; Cheng, S.; Wu, I.-C. Low-Level Autonomous Control and Tracking of Quadrotor Using Reinforcement Learning. *Control Eng. Pract.* **2020**, *95*, 104222. [\[CrossRef\]](#)
17. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z.; Mat Lazim, I. Adaptive PID Controller Using Sliding Mode Control Approaches for Quadrotor UAV Attitude and Position Stabilization. *Arab. J. Sci. Eng.* **2021**, *46*, 963–981. [\[CrossRef\]](#)
18. Guerrero-Sánchez, M.E.; Hernández-González, O.; Valencia-Palomo, G.; López-Estrada, F.R.; Rodríguez-Mata, A.E.; Garrido, J. Filtered Observer-Based IDA-PBC Control for Trajectory Tracking of a Quadrotor. *IEEE Access* **2021**, *9*, 114821–114835. [\[CrossRef\]](#)
19. Mai, G.; Wang, H.; Wang, Y.; Wu, X.; Jiang, P.; Feng, G. Nonlinear Extended State Observer and Prescribed Performance Fault-Tolerant Control of Quadrotor Unmanned Aerial Vehicles Against Compound Faults. *Aerospace* **2024**, *11*, 903. [\[CrossRef\]](#)
20. Liu, W.; Cheng, X.; Zhang, J. Command Filter-Based Adaptive Fuzzy Integral Backstepping Control for Quadrotor UAV with Input Saturation. *J. Frankl. Inst.* **2023**, *360*, 484–507. [\[CrossRef\]](#)

21. Li, Z.; Ma, X.; Li, Y. Model-Free Control of a Quadrotor Using Adaptive Proportional Derivative-Sliding Mode Control and Robust Integral of the Signum of the Error. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418800885. [\[CrossRef\]](#)
22. An, C.; Jia, S.; Zhou, J.; Wang, C. Fast Model-Free Learning for Controlling a Quadrotor UAV with Designed Error Trajectory. *IEEE Access* **2022**, *10*, 79669–79680. [\[CrossRef\]](#)
23. Zou, X.; Liu, Z.; Wang, B.; Zhao, W.; Dang, Q. Model-Free Control-Based Trajectory Tracking Control of a Tail-Sitter UAV in Hovering Mode. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–20. [\[CrossRef\]](#)
24. Graf, J. *PID Control Fundamentals*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2016; ISBN 9781535358668.
25. Potter, A. *Principles and Applications of PID Controllers*; NY Research Press: Forest Hills, NY, USA, 2015; ISBN 9781632383709.
26. Visioli, A. PID Control System Design and Automatic Tuning Using MATLAB/SIMULINK. *IEEE Control Syst. Mag.* **2021**, *41*, 99–100. [\[CrossRef\]](#)
27. Somefun, O.; Akingbade, K.; Dahunsi, F. The Dilemma of PID Tuning. *Annu. Rev. Control* **2021**, *52*, 65–74. [\[CrossRef\]](#)
28. Borase, R.P.; Maghade, D.; Sondkar, S.; Pawar, S. A Review of PID Control, Tuning Methods, and Applications. *Int. J. Dyn. Control* **2020**, *9*, 818–827. [\[CrossRef\]](#)
29. Perozzi, G.; Efimov, D.; Biannic, J.-M.; Planckaert, L. Trajectory Tracking for a Quadrotor under Wind Perturbations: Sliding Mode Control with State-Dependent Gains. *J. Frankl. Inst.* **2018**, *355*, 4809–4838. [\[CrossRef\]](#)
30. Shao, X.; Sun, G.; Yao, W.; Liu, J.; Wu, L. Adaptive Sliding Mode Control for Quadrotor UAVs with Input Saturation. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 1498–1509. [\[CrossRef\]](#)
31. Liu, K.; Wang, R. Antisaturation Command Filtered Backstepping Control-Based Disturbance Rejection for a Quadrotor UAV. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3577–3581.
32. Koksai, N.; An, H.; Fidan, B. Backstepping-Based Adaptive Control of a Quadrotor UAV with Guaranteed Tracking Performance. *ISA Trans.* **2020**, *105*, 98–110. [\[CrossRef\]](#)
33. Gomes, S.B.V.; Ramos, J.G. Airship Dynamic Modeling for Autonomous Operation. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 20 May 1998; Volume 4, pp. 3462–3467.
34. Madani, T.; Benallegue, A. Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 1515–1520.
35. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Upper Saddle River, NJ, USA, 1991.
36. Boubakir, A.; Labiod, S.; Boudjema, F.; Plestan, F. Linear Adaptive Control of a Class of SISO Nonaffine Nonlinear Systems. *Int. J. Syst. Sci.* **2014**, *45*, 2490–2498. [\[CrossRef\]](#)
37. Boubakir, A.; Labiod, S.; Boudjema, F.; Plestan, F. Design and experimentation of a self-tuning PID control applied to the 3DOF helicopter. *Arch. Control Sci.* **2013**, *23*, 311–331. [\[CrossRef\]](#)
38. Ioannou, P.A.; Sun, J. *Robust Adaptive Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1996.

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.



DSPACE

<https://dspace.org/>

A robust adaptive PID-like controller for quadrotor unmanned aerial vehicle systems

Boubakir, Ahsene; Souanef, Toufik; Labiod, Salim; Whidborne, James F.

2024-12-01

MDPI

<https://doi.org/10.3390/aerospace11120980>

Boubakir A, Souanef T, Labiod S, Whidborne JF. (2024) A robust adaptive PID-like controller for quadrotor unmanned aerial vehicle systems. Aerospace, Volume 11, Issue 12, December 2024, Article number 980