

Article

Dynamic Inversion-Enhanced U-Control of Quadrotor Trajectory Tracking

Ahtisham Lone ¹, Quanmin Zhu ^{1,*}, Hamidreza Nemati ¹ and Paolo Mercorelli ²

¹ School of Engineering, University of the West of England, Frenchay Campus, Coldharbour Lane, Bristol BS16 1QY, UK; ahtisham2.lone@live.uwe.ac.uk (A.L.); hamidreza.nemati@uwe.ac.uk (H.N.)

² Control and Drive Systems, Leuphana University of Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany; paolo.mercorelli@leuphana.de

* Correspondence: quan.zhu@uwe.ac.uk

Abstract: This study proposes a universal (U)-control enhanced with dynamic inversion for a class of multiple-input multiple-output (MIMO) quadrotor flight tracking operations. In the technique, a robust compensator is integrated with the U-controller to achieve stabilisation at the equilibrium and setpoint tracking in the presence of any unmodelled uncertainties and external disturbances. A series of bench tests of simulated and real experiments on a Parrot Mambo quadrotor are conducted to show the design framework from the academic formulation to Simulink simulation and real flight tests.

Keywords: U-model realisation; U-dynamic inversion; quadrotor UAV; quadrotor control; trajectory tracking, simulation and flying tests

1. Introduction

Over the last decade, the fast advancement of unmanned aerial vehicles (UAVs) has opened numerous possibilities in various fields such as surveillance, environmental monitoring, and delivery services. Among the many different types of UAVs, quadrotor systems have gained significant attention due to their mechanical simplicity, agile manoeuvrability, hovering capability, and their ability to vertically take off and land without requiring long runways. These factors along with the rapid development and mass production of electronics in general over the last decade have driven the prices of components down, making them much more affordable, and hence spurring a revolutionary impact in this field where high-performance drones are available to hobbyists whereas this was not the case before. Due to this wide-scale availability and adoption, the control of quadrotor systems has been a hot topic of research and development, which has brought substantial challenges, particularly due to their underactuated nature—using less inputs to drive more outputs. This underactuation requires sophisticated control strategies to achieve precise trajectory tracking and stabilisation which reduce the weight of the UAVs.

Regarding quadrotor control, classical control approaches, such as PID control, have been broadly taken as the first choice in applications due to their practical simplicity and ease of implementation [1,2]. Hybrid methods that combine the PID with other methods like Linear Quadratic Regulators (LQRs) [3] and Fuzzy control [4] have also been explored. However, these methods often fall short in handling the nonlinear dynamics and coupling effects inherent in quadrotor systems in flight tests. As a result, more advanced control techniques, including model predictive control (MPC) [5], sliding mode control (SMC) [6], and other nonlinear control approaches [7], have tried to enhance the capacity and robustness of quadrotor control systems.

Even after such tremendous effort has been devoted to UAV control, there are still some improvement domains that could enhance the control system design. Consequently, this paper expands a model-based U-control system design platform [8–11] to achieve trajectory tracking while maintaining robust stability in UAV control, which uses a transparent double



Citation: Lone, A.; Zhu, Q.; Nemati, H.; Mercorelli, P. Dynamic Inversion-Enhanced U-Control of Quadrotor Trajectory Tracking. *Drones* **2024**, *8*, 599. <https://doi.org/10.3390/drones8100599>

Academic Editor: Bo Li

Received: 20 August 2024

Revised: 9 October 2024

Accepted: 16 October 2024

Published: 18 October 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/>).

dynamic inversion within two loops. In the inner loop, the plant dynamic inversion makes the control system design separate from the plant model, and the outer loop dynamic inversion makes the whole system performance specification be achieved with feedback and be unilaterally independent from the plant model. In comparison to some conventional delegate approaches, feedback linearisation control (FLC) has been widely considered as a representative approach [12] in dealing with nonlinear models by transforming into a linear one through coordinate conversions. Another well-known approach is backstepping control (BSC) [13], which is a Lyapunov-based recursive design approach and is able to accomplish the stabilisation and tracking tasks for a specific set of complex nonlinear systems. These two approaches use dynamic inversion in one way or another; however, they suffer from some shortcomings: (1) the FLC first utilises the complex coordinate transformation to design a linear control system, and then transforms the designed controller to the original controller to generate the required controller output. Unfavourably, such conversion needs strong linear algebraic conditions [14]. For BSC, it is normally applicable to a class of triangle-type models. The BSC is effective for second-order dynamics; however, the complexity would exponentially increase with the increase in the system dynamic order [15]. (2) To date, to the author's best knowledge, almost all the existing nonlinear dynamic inversion formulations have assumed linearity with the plant input (that is, output affine plants). Accordingly, a general and concise mode-based dynamic inversion-augmented control method called model-based U-control has been established by using the expression of the U-model, which converts almost all the existing models into U-realisation without referring to the coordinate transform and backstepping process [9,10,16]. It should be mentioned that model-based U-control can dramatically reduce the complexity of designing linear, and particularly nonlinear, control systems, and the U-dynamic inversion and controller design are achieved separately. However, like all the other model-based dynamic inversion approaches, the U-dynamic inversion unavoidably depends on the model accuracy and the robustness against internal uncertainty and external disturbance. Developing an effective model-based U-control method has been the first motivation to balance the universal control flexibility and proper robustness. This study provides a new angle to enhance the objectives and to bench-test the new model-based U-control feasibility and effectiveness through simulations and real flight tests of the UAV.

The key contributions of this work to justify the research motivation and novelty include the following:

- This is the first validity bench test to model-based U-control with UAV flights in a comprehensive procedure, from analytical formulation to simulation demonstration and real flight tests. The applied control system design procedure has general guidance to other industrial applications and is a representative showcase to disseminate U-control methodology.
- In the methodology of U-model-based control, this study proposes a robust compensator to ensure trajectory accuracy and robustness against disturbances in the model-based dynamic inversion. This is supplementary to robust U-control approaches.
- We expand U-sliding mode control (USMC) to MIMO systems from single-input and single-output (SISO) formulation [17].
- We lay a practically effective foundation/platform for establishing the U-control strategy in coping with the challenges posed by underactuation and coupling in the control of quadrotor MIMO systems.
- A series of bench tests of simulated and real experiments on a Parrot Mambo quadrotor are conducted to show the design framework from academic formulation to Simulink simulation and real flight tests. This framework could be used for various UAV or related mobile robot motion control system design.

For the following presentation, Section 2 describes the quadcopter physical model as a problem background and introduces the U-model and U-model dynamic inversion (UMDI) control strategy along with a robust U-model dynamic inversion plus sliding mode control (UDI + USMC) compensator. Section 3 provides the computational simulation setup and

results, validates the proposed approach with the expected performances. Finally, Section 4 summarises the study and gives some potential research directions.

2. Quadrotor Dynamics and U-Control

2.1. Quadcopter System Overview

Taken as a research background for testing the developed approach, this study chooses a Parrot[®] Mambo Minidrone (Parrot S.A, Paris, France) (Figure 1), since it is cost-effective and has a manipulable small size for assessing the practical performance of the designed controllers in a closed-door environment without interference from outside factors like wind, etc. The Simulink Hardware Support Package for Parrot Minidrones, developed by the Original Equipment Manufacturer (OEM), delivers a realistic simulation environment for evaluating controller designs. After being successfully validated with the simulation demonstrations, the controller can then be embedded into the onboard processor by the Hardware Support Package provided to conduct experiments on the hardware to bench-test the outcomes in a real-world scenario.

The minidrone has two sensors on its underside: an ultrasound sensor and a low-resolution (640×480) camera. The ultrasound sensor measures vertical distance accurately up to 13 feet, though its performance varies depending on the floor material. The camera uses optical flow to estimate horizontal motion and speed, but its effectiveness depends on floor colour and texture. Thick carpets can interfere with both sensors. The minidrone's affordability (around EUR 70) makes it appealing for research despite these limitations. Internally, the minidrone has a pressure sensor for Altitude and an Inertial Measurement Unit (IMU) that includes a three-axis accelerometer and gyroscope. The raw sensor data undergo bias correction, frame rotation, and low-pass filtering to remove noise. Two Kalman filters are implemented to estimate Altitude and XY position, while a complementary filter handles Roll, pitch and Yaw. The sampling time interval for the simulation and flight tests is set to 0.005 s. Actuator limitations are also taken into consideration, with controller output being capped with saturation functions to prevent exceeding the defined thrust limits. These processes are standardized by the OEM's Hardware Support Package, ensuring consistent results. The ground effect also poses considerable challenges to this class of drones; due to their inherent low values of inertia and weight, they are easily affected by it.

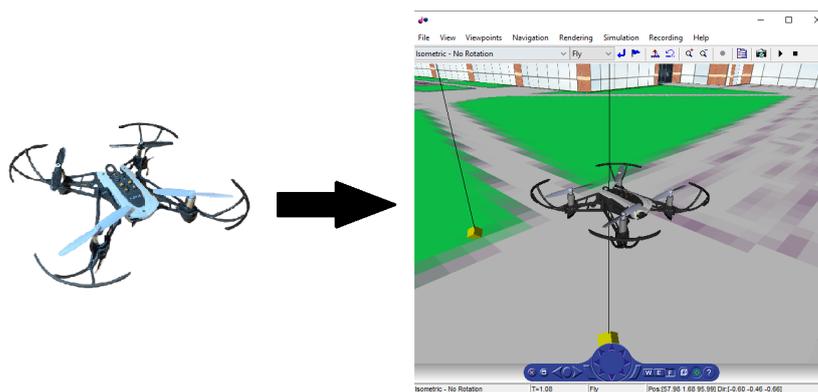


Figure 1. A Parrot Mambo quadrotor and its Simulink simulation environment.

The quadrotor system possesses six degrees of freedom to translate in the three Cartesian directions, i.e., along the X , Y and Z axes, and rotates about these three axes, with Roll ϕ , pitch θ and Yaw ψ denoting the rotations about the X , Y and Z axes, respectively [18].

2.1.1. Assumptions

The dynamic model of the quadrotor system was derived under the assumption that both the structure and rotors are symmetrical and rigid. Another assumption is that the quadrotor systems' centre of mass is not situated at the origin of the body-fixed frame; however, the body-frame axes are aligned with the principal axes of the quadrotor. This

assumption results in a diagonal inertia matrix. Lastly, it is assumed that the generated aerodynamic forces and moments are directly related to the square of the rotor’s speed.

2.1.2. Frames of Reference

The quadrotor’s dynamics are described by employing the following pair of reference frames: the inertial reference frame (IRF) $\mathbb{E}(X_E, Y_E, Z_E)$ and the Body Reference Frame (BRF) $\mathbb{B}(X_B, Y_B, Z_B)$, as illustrated in Figure 2. The quadrotor orientation in the coordinate space is described by $\Theta = [\phi \ \theta \ \psi]^T$, where the Euler angle ϕ represents the Roll (rotation about the X axis), θ represents the pitch (rotation about the Y axis) and ψ represents the Yaw (rotation about the Z axis). These attitude angles are bounded as $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2}), \theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ and $\psi \in (-\pi, \pi)$ because of the various acrobatic flying manoeuvres, which are not admissible. $\omega^B = [p \ q \ r]^T$ represents the quadrotor’s angular velocity in the BRF.

To model the quadrotor UAV, a coordinate transformation matrix between these two reference frames is necessary as the translational equations of motion are described in the IRF while the rotational equations and angular velocities are defined in the BRF. The coordinate transformation matrix is therefore defined as follows ($R_{\phi\theta\psi}$) in Equation (1) [19].

$$R_B^E = \begin{bmatrix} C_\theta C_\psi & S_\theta C_\psi S_\phi - S_\psi C_\phi & S_\theta C_\psi C_\phi + S_\psi S_\phi \\ C_\theta S_\psi & S_\theta S_\psi S_\phi + C_\psi C_\phi & S_\theta S_\psi C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{1}$$

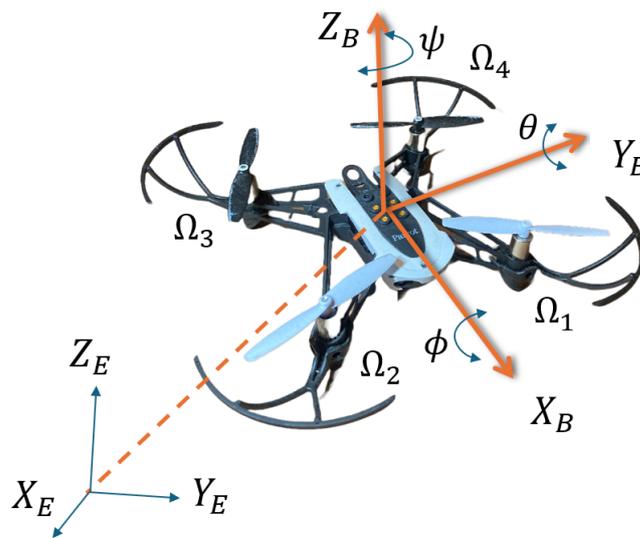


Figure 2. Body Reference Frame and Earth Reference Frame for the Mambo Minidrone.

The transformational relationship between angular velocities calculated in the two different frames of reference, i.e., the IRF and BRF, can be described with the help of the vector \mathbb{M} , which is used to transform the angular velocities ω^B measured in the BRF to Euler rates $\dot{\Theta}$ measured in the IRF and vice versa. Equations (2) and (3) show this relationship and how the transformations can be carried out from one frame of reference to another using the matrix \mathbb{M} and its inverse \mathbb{M}^{-1} as follows when $\omega^B = \mathbb{M}\dot{\Theta}$:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & S_\phi C_\theta \\ 0 & -S_\phi & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

and $\dot{\Theta} = \mathbb{M}^{-1}\omega^B$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_{\phi}T_{\theta} & C_{\phi}T_{\theta} \\ 0 & C_{\phi} & -S_{\phi} \\ 0 & \frac{S_{\phi}}{C_{\theta}} & \frac{C_{\phi}}{C_{\theta}} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3}$$

where “S”, “C” and “T” denote “sin”, “cos” and “tan” trigonometric functions, respectively.

2.1.3. Rotor Dynamics

The thrust T and moment M generated by the i^{th} rotor are calculated as follows:

$$\begin{aligned} T_i &= (C_T \rho A R^2) \Omega_i^2 = (k_{Ti}) \Omega_i^2 \\ M_i &= (C_Q \rho A R^3) \Omega_i^2 = (k_{Mi}) \Omega_i^2 \end{aligned} \tag{4}$$

where Ω_i represents the i^{th} rotor speed. k_{Ti} and k_{Mi} are aerodynamic constants that are associated with the thrust and torque generated by the rotor depending on its shape and configuration. These coefficients can be determined experimentally (Table 1).

Table 1. Parrot Mambo parameters.

Parameter	Value	Units
Thrust Coefficient, C_T	0.0107	
Torque Coefficient, C_Q	7.8264×10^{-4}	
Air density, ρ	1.184	kg/m ³
Rotor Radius, R	0.033	m
Rotor Disk Area, $A = \pi R^2$	0.0034212	m ²
L	0.0624	m
Rotor inertia, J_r	1.0209375×10^{-7}	kg.m ²
Inertia moment along X axis, J_{xx}	0.0000582857	kg.m ²
Inertia moment along Y axis, J_{yy}	0.0000716914	kg.m ²
Inertia moment along Z axis, J_{zz}	0.0001	kg.m ²

Values taken from Parrot Minidrone Simulink hardware package.

The thrust and moment generated by the rotor rely on the rotor’s physical properties, the air density, and the speed at which the rotor is spinning. The term Ω^2 indicates that the thrust and moment are in proportion to the square of the angular velocity of the rotor, which is a key aspect in understanding how changes in rotor speed affect the generated thrust and moment.

The total moments are formulated below:

$$\begin{aligned} \tau_x &= \frac{\sqrt{2}}{2} L (T_1 - T_2 - T_3 + T_4) \\ \tau_y &= \frac{\sqrt{2}}{2} L (-T_1 - T_2 + T_3 + T_4) \\ \tau_z &= (M_1 - M_2 + M_3 - M_4) \end{aligned} \tag{5}$$

where L denotes the distance from the quadrotor mass centre to the rotor rotation axis. The control inputs U_i where $i \in 1, \dots, 4$ are described as follows:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2} L & -\frac{\sqrt{2}}{2} L & -\frac{\sqrt{2}}{2} L & \frac{\sqrt{2}}{2} L \\ -\frac{\sqrt{2}}{2} L & -\frac{\sqrt{2}}{2} L & \frac{\sqrt{2}}{2} L & \frac{\sqrt{2}}{2} L \\ \frac{k_{M1}}{K_{T1}} & \frac{k_{M2}}{K_{T2}} & \frac{k_{M3}}{K_{T3}} & \frac{k_{M4}}{K_{T4}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{6}$$

where U_1 is for the whole thrust added on the quadrotor, U_2 represents the Roll torque τ_x , U_3 represents the pitch torque τ_y and U_4 represents the Yaw torque τ_z .

2.1.4. Translational Subsystem

The equations of motion that describe the system behaviour along the three Cartesian axes in the inertial reference frame (IRF) are

$$m\ddot{X} = mg + R_{\phi\theta\psi}^T u_1 z_E \tag{7}$$

where m is the quadrotor mass, g is the acceleration due to the gravity of Earth, $R_{\phi\theta\psi}^T$ is the coordinate transformation matrix from Equation (1) and $z_E = [0 \ 0 \ 1]^T$ is a unit vector in the inertial frame of reference (IRF).

Equation (7) is then expanded and rewritten as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \begin{bmatrix} C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\phi S_\theta S_\psi - S_\phi C_\psi \\ C_\phi C_\theta \end{bmatrix} \frac{U_1}{m} \tag{8}$$

2.1.5. Rotational Subsystem

The rotational dynamics of a quadrotor are derived by utilizing the Newton–Euler equations as follows:

$$J\dot{\omega}^B + \omega^B \times J\omega^B + M_g = \tau \tag{9}$$

where J is the quadrotor inertia matrix, M_g is the gyroscopic moment generated by the rotors and $\tau = [\tau_x \ \tau_y \ \tau_z]^T$ represents the cumulative effect of moments exerted on the quadrotor. The gyroscopic moment can be characterized as

$$M_g = \omega^B \times [0 \ 0 \ J_r \omega_r]^T \tag{10}$$

where J_r denotes the propeller’s inertia and ω_r represents the relative speed of the propeller, which is defined as

$$\Omega_r = \sum_{i=1}^4 (-1)^{i+1} \omega_i \tag{11}$$

Consequently, the attitude dynamics of the system can be summarised by using Equations (6) and (9) as follows:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \left(\frac{J_{yy} - J_{zz}}{J_{xx}}\right) \dot{\theta} \dot{\psi} \\ \left(\frac{J_{zz} - J_{xx}}{J_{yy}}\right) \dot{\phi} \dot{\psi} \\ \left(\frac{J_{xx} - J_{yy}}{J_{zz}}\right) \dot{\phi} \dot{\theta} \end{bmatrix} - J_r \Omega_r \begin{bmatrix} \dot{\theta} \\ -\dot{\phi} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{U_2}{J_{xx}} \\ \frac{U_3}{J_{yy}} \\ \frac{U_4}{J_{zz}} \end{bmatrix} \tag{12}$$

Subsystems (8) and (12) together represent the full dynamic description of a quadrotor system. As can be observed, the equations that describe the rotational motion of the quadrotor are completely independent of the translational elements X , Y and Z and their derivatives. However, the quadrotor translational motion is dependent on the rotational components ϕ , θ and ψ . Therefore, the overall system is made up of two subsystems with the rotational system in the inner loop, while the translational subsystem in the outer loop generates the required reference signals for the inner loop in the form of desired Roll and pitch angles to track the desired location. Effective attitude control, therefore, is critical to ensure that the UAVs can perform a wide range of tasks that can range from simple hovering to complex aerial manoeuvres while maintaining stability and responding appropriately to external disturbances like wind since the UAV’s position is controlled indirectly by manipulating the pitch, Roll and Yaw.

2.2. Model-Based U-Control Platform

Figure 3 shows the model-based U-control framework [8,20]; the controller consists of two parts, with the first being the universal controller (G_{cl}) which is to be designed

completely independently from the plant, according to the desired indices, and the second part being the plant inverse (G_p^{-1}). Consequently, the primary objective in the entire designing process boils down to the determination of the inverse G_p^{-1} of the plant G_p that is to be regulated and the combination of the plant inverse with the specified universal controller G_{cl} .

The following is a brief overview highlighting the up-to-date research of the U-control design framework approach. In [20], the authors propose a pole placement control strategy for complex dynamic systems by leveraging U-models, demonstrating improved system stability and performance. In [8], a general U-block model-based design procedure is proposed that aims to simplify the control of a class of complex polynomial difference equations. The methodology effectively transforms nonlinear control problems into linear ones using U-models, facilitating easier controller design and implementation. Further developing on the previous work, [9] extends the U-model methodology to rational nonlinear systems, offering insights into controlling more complex dynamics, highlighting the flexibility of U-models in addressing the challenges posed by rational nonlinearities in control systems. In [21], the authors address the issue of input delay in nonlinear processes, using a U-model-based predictive control approach that significantly enhances control accuracy and system responsiveness, showcasing the practical applicability of U-models in real-world scenarios. More recently, [22] explores the application of U-model-augmented control in the context of underactuated unmanned marine robotics. Ref. [10] presents solutions for reversing dynamics based on the U-model and [11] outlines a comprehensive U-control framework for continuous-time systems. Ref. [16] consolidates the U-model and U-control methodologies, providing a detailed analysis of their application in complex dynamic systems.

Remark 1. *It is worth noting that robust flat control (RFC) [23,24] has been a very effective approach using dynamic inversion to enhance the control system design in a large class of wheeled mobile robots, UAVs and other motion systems. In comparison with model-based U-control, RFC needs coordinate transform to achieve the dynamic inversion to determine the control input and the model-based U-control directly takes dynamic inversion from the plant U-model realisation, which is straightforward in the reformulation of the plant model into a control-oriented time-varying model [16]. Regarding the robustness of the two approaches, RFC could be more effective than model-based U-control. Surely, RFC provides a very useful reference to improve U-control robustness and we will provide simulation comparison as well in future publications.*

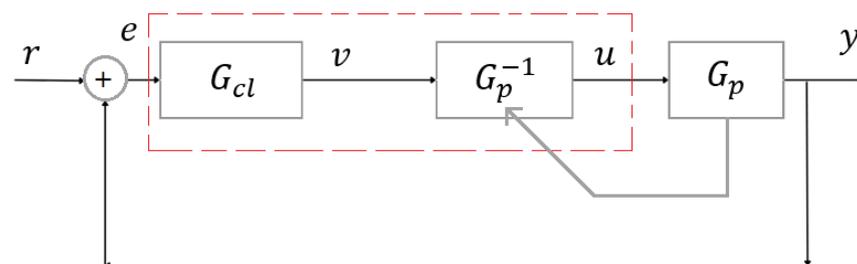


Figure 3. U-control framework.

2.2.1. Design of U-Controller G_{cl}

To begin the process of designing the universal controller, assign the desired system performance in advance with the selection of an ideal closed-loop transfer function as

$$\frac{Y(s)}{R(s)} = G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{4}{s^2 + 4s + 4} \quad (13)$$

where $\zeta = 1$ is the selected damping ratio and $\omega_n = 2$ is the chosen undamped natural frequency, so that the system output response is critically damped to prevent any overshoot,

ensuring that the system returns to the desired state in the shortest possible time without oscillating [25,26].

The determination of the universal controller treats the pair of the plant and its inverse as a unit in the feedback control system, and then takes the inverse of the closed-loop transfer function in Equation (13) as follows:

$$G_{cl} = \frac{G}{1 - G} = \frac{4}{s^2 + 4s} \tag{14}$$

2.2.2. U-Polynomial Realisation

Before calculating the plant inverse, the system needs to be converted to its U-realisation (either U-Polynomial or U-state space) depending on the system. A brief overview of the realisation of the system to its appropriate U-realisation is given in this section before proceeding to controller design.

Consider a general SISO continuous-time polynomial system defined with $(y(t), \lambda(t)$ and $u(t))$ where $y(t) \in \mathbb{R}$ and $u(t) \in \mathbb{R}$ are the system output and system input, respectively, at time $t \in \mathbb{R}^+$, which is described as follows:

$$Y(t)^{(M)} = \sum_{j=0}^J \lambda_j(Y_{M-1}, (\Xi_{N-1}, \mu)) \left(u(t)^{(N)}\right)^j, M \geq N \tag{15}$$

where the term $y(t)^{(M)}$ represents the M^{th} -order derivative of the output $y(t)$ and the term $u(t)^{(N)}$ represents the N^{th} -order derivative of the system input $u(t)$, while the term $\lambda_j(*) \in \mathbb{R}$ represents a time-varying parameter that absorbs all the other inputs and outputs.

$$\begin{aligned} Y_{M-1} &= [y(t)^{(M-1)}, y(t)^{(M-2)}, \dots, y(t)] \in \mathbb{R}^M \\ \Xi_{N-1} &= [u(t)^{(N-1)}, u(t)^{(N-2)}, \dots, u(t)] \in \mathbb{R}^M \\ \mu &= \left(u(t)^{(N)}\right)^j \end{aligned} \tag{16}$$

Consider an example showcasing the U-realisation of a classical nonlinear polynomial representation [27]:

$$\ddot{z} = z + (z - 2)\dot{z} + (4 + z^2)v + (z + \dot{z}^2)v^2 \tag{17}$$

The U-model realisation, then, is

$$\begin{cases} \ddot{z} = \lambda_0 + \lambda_1 v + \lambda_2 v^2 \\ \lambda_0 = z + (z - 2)\dot{z} \\ \lambda_1 = (4 + z^2) \\ \lambda_2 = (z + \dot{z}^2) \end{cases} \tag{18}$$

Comparing Equations (17) and (18), the model's U-realisation is quite forthright. The U-realisation is the same as the original polynomial in attributes but is expressed in a control-oriented structure.

2.2.3. U-State Space Realisation

Consider a general SISO state space description as follows:

$$\begin{cases} \dot{Q} = G(Q, u) \\ y = H(Q) \end{cases} \tag{19}$$

where $u, y, \in \mathbb{R}, Q \in \mathbb{R}^n, G \in \mathbb{R}^n \rightarrow$ and $H \in \mathbb{R}$ are the smooth mappings $u \rightarrow Q \in \mathbb{R} \rightarrow \mathbb{R}^n$ and $Q \rightarrow y \in \mathbb{R}^n \rightarrow \mathbb{R}$, respectively.

This state space representation of the system is expressed as a multilayer polynomial representation as follows:

$$\begin{cases} \dot{q}_1 = G_1(q_1, q_2, \dots, q_n), \\ \dot{q}_2 = G_2(q_1, q_2, \dots, q_n), \\ \vdots \\ \dot{q}_n = G_n(q_1, q_2, \dots, q_n, u), \\ y = H(q_1, q_2, \dots, q_n) \end{cases} \quad (20)$$

Convert this multilayer polynomial expression into a multilayer U-realisation as

$$\begin{cases} \dot{q}_1 = \sum_{i=0}^n \lambda_{1i} g_{1i}(q_2), \\ \dot{q}_2 = \sum_{i=0}^n \lambda_{2i} g_{2i}(q_3), \\ \vdots \\ \dot{q}_n = \sum_{i=0}^n \lambda_{ni} g_{ni}(u), \\ y = \sum_{i=0}^n h_i(q_1, q_2, \dots, q_n) \end{cases} \quad (21)$$

where $\lambda(*)$ and $g(*)$ are the time-varying parameters absorbing the remaining terms in the state space model. It is worth mentioning that the U-state space realisation is a form of expanded multilayer U-Polynomial expression because each line of the U-state space realisation can be considered a U-Polynomial by itself.

In case Equation (21) is unable to exhibit the explicit association between the system output term y and the input term u , differentiate y and substitute to x_i items in system (21) with the polynomial expression for each x_i in Equation (20). This iterative process should be carried out until an explicit expression between $y^{(m)}$ and u is exhibited, where $y^{(m)}$ denotes the m^{th} derivative of the model output y .

2.3. Determining the Dynamic Inverse

After establishing a direct association between $y^{(m)}$ and u in system (21), the system equation can be restructured as follows:

$$y^{(m)} = \lambda_1(q_1, q_2, \dots, q_{n-1}, q_n) + \lambda_2(q_1, q_2, \dots, q_{n-1}, q_n)u \quad (22)$$

where λ_i ($i \in 1, 2, \dots$) is a time-varying gain vector soaking up all the state variables. Equation (22) can be identified as an alternative U-model polynomial described by Equation (15). The output of the plant inverter u_i can be calculated by reorganising and isolating the input term u in Equation (22):

$$u = \frac{y^{(m)} - \lambda_1(q_1, q_2, \dots, q_{n-1}, q_n)}{\lambda_2(q_1, q_2, \dots, q_{n-1}, q_n)} \quad (23)$$

In addition, the state variables q_i are worked out by the integral of \dot{q}_i associated with the plant. This relationship is expressed as $q_i = \dot{q}_i \cdot \frac{1}{s}$, where $\frac{1}{s}$ is the integral operation of the Laplace transform.

For illustration purposes, consider a simple nonlinear plant modelled by

$$\begin{cases} \dot{q}_1 &= q_2 \\ \dot{q}_2 &= -q_1 - \sin(q_2) + u \\ y &= q_1 \end{cases} \tag{24}$$

Since no direct relation between the input u and output y can be observed here, y is differentiated repeatedly till an expression linking the two explicitly is obtained.

$$\ddot{y} = -q_1 - \sin(q_2) + u \tag{25}$$

Equation (25) establishes a direct link between the input and output, making it a viable option for dynamic inversion.

The associated U-realisation and the dynamic inverse is then

$$\begin{cases} \ddot{y} &= \lambda_0 + \lambda_1 u \\ \lambda_0 &= -q_1 - \sin(q_2) ; \lambda_1 = 1 \\ u_i &= \frac{\ddot{y} - \lambda_0}{\lambda_1} = \ddot{y} + q_1 + \sin(q_2) \end{cases} \tag{26}$$

With the aim of assigning the plant output y equivalent to the output (v) of the controller (G_{cl}), the plant dynamic inversion G_p^{-1} can be computed. This involves substituting the highest order of the y term with v of the same order. Following this, the expression is integrated twice to yield y :

$$y = \frac{\lambda_0}{s^2} + \frac{\lambda_1}{s^2} u \tag{27}$$

To directly determine the second-order derivative \ddot{v} , the universal controller can be modified directly to ensure that the same dynamic orders of v and y are in the designed control system.

Remark 2. For the system represented by Equation (24), the method to accomplish this goal is through multiplication of the second-order Laplace operator s^2 with the universal controller G_{cl} to yield $\ddot{v} = s^2 v$. Consequently, the output y is substituted by the second-order derivative of the invariant controller G_{cl} output, denoted as \ddot{v} . Therefore, $u = q_1 + \sin q_2 + \ddot{v}$ results in $G_p G_p^{-1} = 1$. Furthermore, the universal controller G_{cl} from Equation (14) can be rewritten as follows:

$$G_{cl} = \frac{4s^2}{s^2 + 4s} \tag{28}$$

2.4. Compensated Plant Inverter G_c^{-1} Design

As shown in Figure 4, expanded from Figure 3, this study introduces a sliding mode control (SMC) into model-based U-control for MIMO systems to enhance their robustness through the addition of a robust SMC to accommodate errors induced from the plant inverter G_p^{-1} due to the plant uncertainties and system disturbances. To determine G_c^{-1} , for a general second-order system that is defined as

$$\begin{cases} \dot{q}_1 &= g_1(q_1, q_2) \\ \dot{q}_2 &= g_2(q_1, q_2) + h(q_1, q_2)u + d \\ y &= q_1 \end{cases} \tag{29}$$

where $g_i \in \mathbb{R}^n$ is a smooth function absorbing all states, h represents the relationship between state variables and control outputs u and d are the disturbance of where $|d| < \bar{d}$.

The state error vector is defined as follows:

$$\begin{cases} \tilde{q}_1 &= q_1 - q_d \\ \tilde{q}_2 &= \dot{q}_1 - \dot{q}_d \end{cases} \tag{30}$$

where \dot{q}_1 and \dot{q}_d are the derivatives of the plant output q_1 and desired output q_d , respectively. Then, we can define a sliding mode manifold σ by [28]

$$\sigma = c_1 \tilde{q}_1 + \tilde{q}_2 \tag{31}$$

And its derivative $\dot{\sigma}$ is calculated as follows:

$$\dot{\sigma} = c_1(\dot{q}_1 - \dot{q}_d) + (\dot{q}_2 - \dot{q}_d) \tag{32}$$

where $c_i \in \mathbb{R}^+$ and $i \in \mathbb{R}^+$ are the coefficient specifying a stable Hurwitz polynomial and the asymptotically convergent speed [28].

Similar to the state space expression, replace the highest-order derivative of the desired output \ddot{q}_d with vs . Consequently, the lower-order derivatives of the desired output q_d are

$$\begin{aligned} \dot{q}_d &= vs \cdot \left(\frac{1}{s}\right) \\ q_d &= vs \cdot \left(\frac{1}{s}\right)^2 \end{aligned} \tag{33}$$

where the term $\frac{1}{s}$ is the Laplace integration operator. Accordingly, convert error Equations (30) and (32) into

$$\begin{cases} \tilde{q}_1 = q_1 - v \left(\frac{1}{s}\right)^2 \\ \tilde{q}_2 = \dot{q}_1 - v \left(\frac{1}{s}\right) \end{cases} \tag{34}$$

$$\dot{\sigma} = c_1 \left(\dot{q}_1 - v \left(\frac{1}{s}\right) \right) + (\dot{q}_1 - v) \tag{35}$$

The calculation process for the m^{th} -order derivative of state q_1 , i.e., $\frac{m}{q_1}$, is found by continuously differentiating its lower-order derivatives; the detailed process is described in [17]. In the case of a second-order system, as described here, $m = 2$, so \ddot{q}_1 is calculated as

$$\begin{cases} \dot{q}_1 = G_1(q_1, q_2) = \mathcal{G}_1(q_1, q_2) \\ \ddot{q}_1 = \frac{\partial \mathcal{G}_1}{\partial q_1} G_1 + \frac{\partial \mathcal{G}_1}{\partial q_2} (G_2 + hu_s + d) \end{cases} \tag{36}$$

Substituting this value of \ddot{q}_1 in Equation (35), it is then rewritten as follows:

$$\dot{\sigma} = c_1 \left(\dot{q}_1 - v \left(\frac{1}{s}\right) \right) + \left(\left(\frac{\partial \mathcal{G}_1}{\partial q_1} G_1 + \frac{\partial \mathcal{G}_1}{\partial q_2} (G_2 + hu + d) \right) - v \right) \tag{37}$$

The robust sliding mode controller output is made of two parts, with the first being the equivalent control u_{eq} and the second part being the switching control u_{sw} . To obtain the equivalent controller output, let $\dot{\sigma} = 0$ [28].

Similarly, the USMC final controller output consists of two parts, with the first being from the ideal plant inverse G_p^{-1} , i.e., ideal inversion output u_i and the compensated inversion output u_c . Equating them and reorganising the terms gives the following:

$$\begin{aligned} u_i + u_c &= u_{eq} + u_{sw} \\ u_c &= u_{sw} + u_{eq} - u_i \end{aligned} \tag{38}$$

Differentiating the system output y from Equation (29) until you obtain a direct relationship between y and u shows

$$\begin{aligned} \dot{y} &= \dot{q}_1 = q_2 \\ \ddot{y} &= \ddot{q}_1 = \dot{q}_2 = g(q_1, q_2) + u = \lambda_0 + \lambda_1 u \end{aligned} \tag{39}$$

In Equation (39), the term \dot{q}_2 is rewritten into its U-model equivalent form, i.e., $\lambda_0 + \lambda_1 u$. Referring to the relationship between the plant input u and the system output y , the ideal inverted output, denoted as u_i , can be determined as follows:

$$u_i = \frac{\dot{y} - \lambda_0}{\lambda_1} = \frac{v - \lambda_0}{\lambda_1} \tag{40}$$

Replacing the term \dot{q}_2 with its U-model equivalent from Equation (35) gives

$$\dot{\sigma} = c \left(\dot{q}_1 - v \frac{1}{s} \right) + (\lambda_0 + \lambda_1 u - v) \tag{41}$$

Setting $\dot{\sigma} = 0$ to calculate u_{eq} gives

$$u_{eq} = \frac{v - c \left(\dot{q}_1 - v \frac{1}{s} \right) - \lambda_0}{\lambda_1} \tag{42}$$

$$u_{sw} = -(\varepsilon \text{sign}(\sigma) + k\sigma) \tag{43}$$

where $\varepsilon = (\bar{d} + \rho)$ with $\rho > 0$ and $k > 0$. Then, from Equation (38),

$$\begin{aligned} u_c &= u_{sw} + u_{eq} - u_i \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{v - c \left(\dot{q}_2 - v \frac{1}{s} \right) - \lambda_0}{\lambda_1} - \frac{(v - \lambda_0)}{\lambda_1} \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{\cancel{\dot{q}_2} - c \left(\dot{q}_2 - v \frac{1}{s} \right) - \cancel{\lambda_0} - \cancel{\dot{q}_2} + \cancel{\lambda_0}}{\lambda_1} \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{-c \left(\dot{q}_2 - v \frac{1}{s} \right)}{\lambda_1} \end{aligned} \tag{44}$$

where $c \in \mathbb{R}^+, k \in \mathbb{R}^+$ and $\varepsilon \in \mathbb{R}^+$ are design parameters associated with the Sliding Mode Compensator. This u_c represents the compensator output and together with the ideal inverter output u_i forms the final overall controller output u_s , as can be shown in Figure 4.

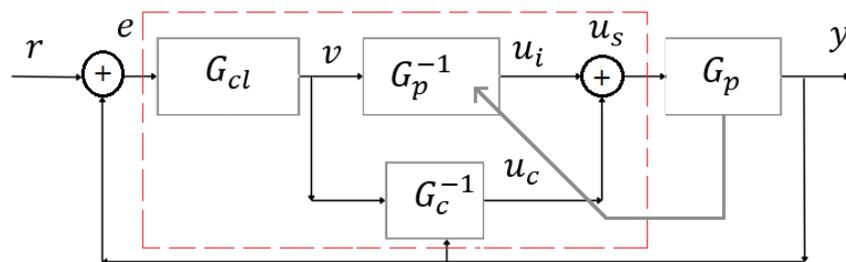


Figure 4. U-model dynamic inversion with compensator.

Remark 3. To reduce the effects of the chattering effect, replace the discontinuous function $\text{sign}(\sigma)$ with a smooth sigmoid function [28] as $\text{sign}(\sigma) \approx \frac{\sigma}{|\sigma| + \xi}$, where ξ is a small positive scalar and it can be observed that, point-wise,

$$\lim_{\xi \rightarrow 0} \frac{\sigma}{|\sigma| + \xi} = \text{sign}(\sigma)$$

Equation (44) can therefore be rewritten by substituting the $\text{sign}(\ast)$ function as follows:

$$u_c = \left(-\left(\varepsilon \frac{\sigma}{|\sigma| + \xi} + k\sigma \right) + \frac{-c \left(\dot{q}_2 - v \frac{1}{s} \right)}{\lambda_1} \right) \tag{45}$$

Taking the final USMC controller output, $u_s = u_c + u_i$, and substituting in the value of u_s in Equation (37), we end up with the following:

$$\dot{\sigma} = -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{\partial \mathcal{G}_1}{\partial q_2} d \tag{46}$$

To analyse the stability of this approach, we use Lyapunov theory ,where a typical Lyapunov candidate function and its time derivative are defined as

$$\begin{aligned} V(\sigma) &= \frac{1}{2}\sigma^2 \geq 0 \\ \dot{V}(\sigma) &= \sigma\dot{\sigma} \end{aligned} \tag{47}$$

using control input u that stabilizes the system satisfying $\dot{\sigma} = -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{\partial \mathcal{G}_1}{\partial q_2} d$, where k is a positive gain and $\varepsilon = (\bar{d} + \rho)$, with $\rho > 0$. Then, substituting the control law into the time derivative,

$$\begin{aligned} \dot{V}(\sigma) &= \sigma\dot{\sigma} = \sigma\left(-\varepsilon \text{sign}(\sigma) - k\sigma + \frac{\partial \mathcal{G}_1}{\partial q_2} d\right) \\ &= -\varepsilon|\sigma| - k\sigma^2 + \sigma\left(\frac{\partial \mathcal{G}_1}{\partial q_2} d\right) \\ &= -(\bar{d} + \rho)|\sigma| - k\sigma^2 + \sigma\left(\frac{\partial \mathcal{G}_1}{\partial q_2} d\right) \leq -\rho|\sigma| - k\sigma^2 \leq 0 \end{aligned} \tag{48}$$

and by selecting gains where $k > 0$ and $\varepsilon = (\bar{d} + \rho)$, where ρ is a positive number that is greater than $|d|$ to cancel out the effect of disturbances, it follows that

$$\dot{V}(\sigma) \leq 0 \tag{49}$$

This demonstrates that when appropriate positive gains are used in the control law, $\dot{V}(\sigma)$ is negative semi-definite, leading to stability.

Secondly, the controller $G_{cl}(s)$ is used to specify the desired system performance in Equation (13), where $\zeta = 1$ represents the selected damping ratio and $\omega_n = 2$ is the undamped natural frequency so that the closed loop control system generates a fast monotonic response and zero steady state error to level reference. The corresponding characteristic equation $D(s)$ and its roots s_1 and s_2 are

$$\begin{aligned} D(s) &= s^2 + 4s + 4 = 0 \\ s_1, s_2 &= -2 \end{aligned} \tag{50}$$

Since the first column of the Routh array is $[1, 4, 4]$ and all its elements are positive, indicating no change of sign, and all the poles are in the left half of the s-plane, the system as a whole is Routh-stable while the inner loop dynamic inversion results in $G_p G_p^{-1} \rightarrow 1$.

3. Main Results

3.1. Model-Based U-Control of the Rotational Subsystem

The Roll of the quadrotor system is represented by the following subsystem:

$$\begin{aligned} \dot{x}_1 &= \dot{\phi} = x_2 \\ \dot{x}_2 &= \ddot{\phi} = \frac{J_{yy} - J_{zz}}{J_{xx}} \dot{\theta} \dot{\psi} - J_r \Omega_r \frac{\dot{\theta}}{J_{xx}} + \frac{U_2}{J_{xx}} \end{aligned} \tag{51}$$

The U-realisation of the Roll subsystem is

$$\begin{cases} \ddot{\phi} = \lambda_0 + \lambda_1 U_2 \\ \lambda_0 = \frac{J_{yy} - J_{zz}}{J_{xx}} \dot{\theta} \dot{\psi} - J_r \Omega_r \frac{\dot{\theta}}{J_{xx}} \\ \lambda_1 = \frac{1}{J_{xx}} \end{cases} \quad (52)$$

The control input term U_2 is determined by

$$U_2 = \frac{\ddot{\phi} - \lambda_0}{\lambda_1} \quad (53)$$

Similarly, for the pitch channel, the U-controller is designed by first converting the state space subsystem into its U-control expression and then extracting the control input term as above.

$$\begin{aligned} \dot{x}_3 &= \dot{\theta} = x_4 \\ \dot{x}_4 &= \ddot{\theta} = \frac{J_{zz} - J_{xx}}{J_{yy}} \dot{\phi} \dot{\psi} - J_r \Omega_r \frac{-\dot{\phi}}{J_{yy}} + \frac{U_3}{J_{yy}} \end{aligned} \quad (54)$$

The U-model realisation of the pitch subsystem is

$$\begin{cases} \ddot{\theta} = \lambda_0 + \lambda_1 U_3 \\ \lambda_0 = \frac{J_{zz} - J_{xx}}{J_{yy}} \dot{\phi} \dot{\psi} - J_r \Omega_r \frac{\dot{\phi}}{J_{yy}} \\ \lambda_1 = \frac{1}{J_{yy}} \end{cases} \quad (55)$$

We can work out the control input term U_3 using

$$U_3 = \frac{\ddot{\theta} - \lambda_0}{\lambda_1} \quad (56)$$

Again, repeating the same process as before by converting the state space subsystem into its U-control realisation, the Yaw subsystem can therefore be written as follows:

$$\begin{aligned} \dot{x}_5 &= \dot{\psi} = x_6 \\ \dot{x}_6 &= \ddot{\psi} = \frac{J_{xx} - J_{yy}}{J_{zz}} \dot{\phi} \dot{\theta} + \frac{U_4}{J_{zz}} \end{aligned} \quad (57)$$

$$\begin{cases} \ddot{\psi} = \lambda_0 + \lambda_1 U_4 \\ \lambda_0 = \frac{J_{xx} - J_{yy}}{J_{zz}} \dot{\phi} \dot{\theta} \\ \lambda_1 = \frac{1}{J_{zz}} \end{cases} \quad (58)$$

The control input term U_4 can now be calculated as

$$U_4 = \frac{\ddot{\psi} - \lambda_0}{\lambda_1} \quad (59)$$

3.2. Model-Based U-Control of Translational Subsystem

For the Altitude channel, since there is a direct control input U_1 for its regulation, we similarly designed the channel to follow that of the rotational subsystems where they are processed as a series of multiple SISO systems. For the position control along the X and Y directions, the control is indirect in the sense that a decoupling algorithm needs to calculate the specified Roll (ϕ_d) and pitch (θ_d) that are passed onto the attitude controller to successfully track the desired trajectories.

3.2.1. Altitude Controller

The Altitude subsystem has the following principle model:

$$\begin{aligned} \dot{x}_7 &= \dot{z} = x_8 \\ \dot{x}_8 &= \ddot{z} = -g + (C_\phi C_\theta) \frac{U_1}{m} \end{aligned} \quad (60)$$

Converting this into the U-control realisation, the Altitude subsystem is

$$\begin{cases} \ddot{z} = \lambda_0 + \lambda_1 U_1 \\ \lambda_0 = -g \\ \lambda_1 = \frac{(C_\phi C_\theta)}{m} \end{cases} \quad (61)$$

The control input term U_4 can now be calculated as follows:

$$U_1 = \frac{\ddot{z} - \lambda_0}{\lambda_1} \quad (62)$$

3.2.2. Model-Based Position U-Controller

From Figure 5, it is clear that the outer loop deals with the positioning by generating the desired setpoints for Roll (ϕ_d) and pitch (θ_d), which are then passed onto the rotational subsystem as reference commands. The desired trajectories x_d and y_d are passed on to the position controller along with the Yaw output and thrust as the change in Yaw affects the heading, and, therefore, trajectories need to be calculated accordingly and the vertical and horizontal thrust components from Roll and pitch manoeuvres will affect the Altitude thrust. In order to address these issues, virtual inputs u_x , u_y and u_z are used as they simplify and decouple the complex underactuated dynamics of the quadrotor, making it easier to design and implement control strategies. The decoupling algorithm simplifies the design of controllers because each virtual input can be associated with a specific axis or motion:

- u_x controls the Roll and lateral motion (sideways movement).
- u_y controls the pitch and longitudinal motion (forward/backward movement).
- u_z controls the thrust and vertical motion (up/down movement).

One can formulate the required pitch θ_d and Roll ϕ_d angles using the following decoupling algorithm [29]:

$$\begin{cases} \phi_d = \arcsin\left(\frac{m(u_x \sin(\psi) - u_y \cos(\psi))}{u_1}\right) \\ \theta_d = \arctan\left(\frac{u_x \cos(\psi) + u_y \sin(\psi)}{u_z}\right) \end{cases} \quad (63)$$

The position of the system is determined by the following subsystem:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} (C_\phi S_\theta C_\psi + S_\phi S_\psi) \\ (C_\phi S_\theta S_\psi - S_\phi C_\psi) \end{bmatrix} \frac{U_1}{m} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (64)$$

Converting these into the U-control realisation, the position control subsystem is as follows:

$$\begin{cases} \ddot{x} = \lambda_0 + \lambda_1 u_x ; \text{ where } \lambda_0 = 0 ; \lambda_1 = 1 \\ \ddot{y} = \lambda_0 + \lambda_1 u_y ; \text{ where } \lambda_0 = 0 ; \lambda_1 = 1 \end{cases} \quad (65)$$

The virtual control inputs u_x and u_y can now be calculated as

$$u_x = \frac{v - \lambda_0}{\lambda_1}; u_y = \frac{v - \lambda_0}{\lambda_1} \tag{66}$$

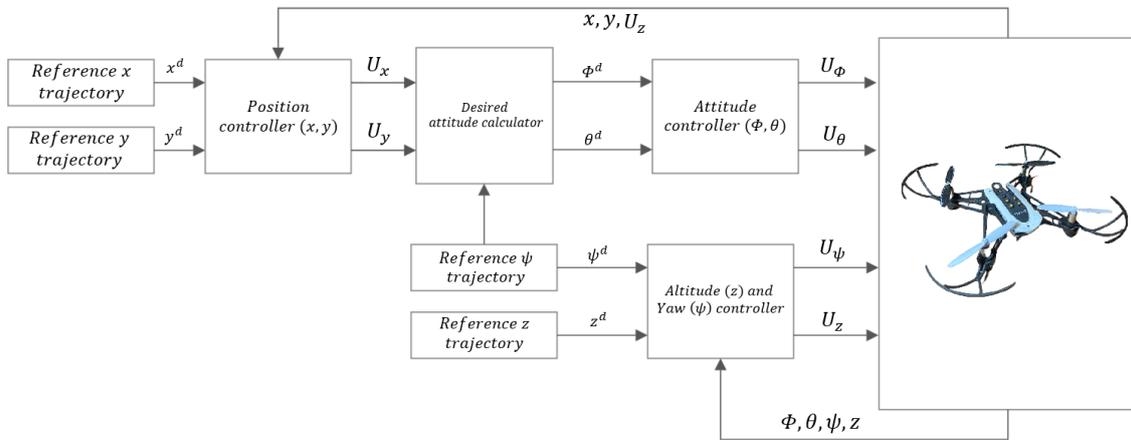


Figure 5. Schematic of the proposed control approach.

Implementing the derived control laws, the simulated system performance can be seen in Figure 6. It is noted that the model-based U-dynamic inversion approach is sensitive to model inaccuracies and its performance deteriorates in the presence of unmodelled dynamics or disturbances, resulting in model–inverse mismatch (i.e., $G_p \times G_p^{-1} \neq 1$). To accommodate the problem, a robust compensator with a Quasi-Sliding Mode [28] is added to strengthen the stability and eliminate convergent error.

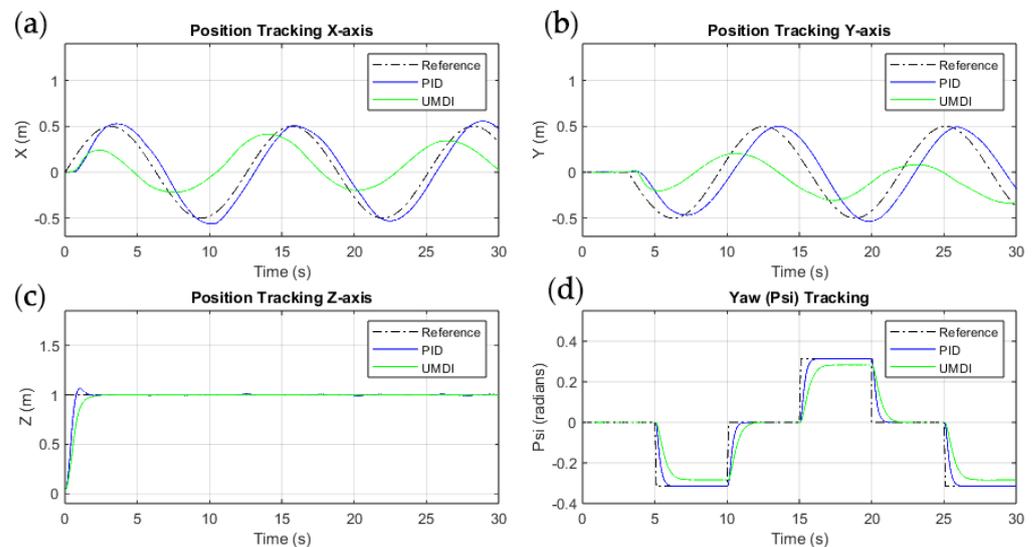


Figure 6. Trajectory tracking performance of UMDI compared with a tuned PID controller. (a) Trajectory in X coordinate. (b) Trajectory in Y coordinate. (c) Trajectory in Z coordinate. (d) Trajectory in Yaw channel.

3.3. Robust Compensator Design Procedure

Following from Section 2.4, a robust Sliding Mode-based Compensator is implemented to improve the performance of the UMDI due to its sensitivity to model inaccuracies. Considering the Yaw subsystem from earlier, defined by its state space Equation (57), U-state space realisation (58) and finally the ideal inversion defined by Equation (59),

$$\begin{aligned} \dot{x}_5 &= \dot{\psi} = x_6 \\ \dot{x}_6 &= \ddot{\psi} = \frac{J_{xx} - J_{yy}}{J_{zz}} \phi \dot{\theta} + \frac{U_4}{J_{zz}} \end{aligned} \tag{67}$$

$$\begin{cases} \ddot{\psi} = \lambda_0 + \lambda_1 U_4 \\ \lambda_0 = \frac{J_{xx} - J_{yy}}{J_{zz}} \phi \dot{\theta} \\ \lambda_1 = \frac{1}{J_{zz}} \end{cases} \tag{68}$$

The ideal inversion output term U_4 is calculated directly as there is a direct relationship between input u and output ψ :

$$U_4 = \frac{\ddot{\psi} - \lambda_0}{\lambda_1} = \frac{v - \lambda_0}{\lambda_1} \tag{69}$$

Based on Equations (34) and (35), the error equations for $m = 2$ (i.e., a second-order system) are as follows:

$$\begin{cases} \tilde{q}_1 = \psi - v \left(\frac{1}{s}\right)^2 \\ \tilde{q}_2 = \dot{\tilde{q}}_1 = \dot{\psi} - v \left(\frac{1}{s}\right) \end{cases} \tag{70}$$

The sliding mode manifold σ and its derivative $\dot{\sigma}$ are given by

$$\begin{aligned} \sigma &= c\tilde{q}_1 + \tilde{q}_2 \\ \dot{\sigma} &= c\dot{\tilde{q}}_1 + \dot{\tilde{q}}_2 = c\left(\dot{\psi} - v\frac{1}{s}\right) + (\ddot{\psi} - v) \end{aligned} \tag{71}$$

Replacing the term $\ddot{\psi}$ with its U-model equivalent from Equation (58) gives

$$\dot{\sigma} = c\left(\dot{\psi} - v\frac{1}{s}\right) + (\lambda_0 + \lambda_1 u - v) \tag{72}$$

Setting $\dot{\sigma} = 0$ to calculate u_{eq} gives

$$u_{eq} = \frac{v - c\left(\dot{\psi} - v\frac{1}{s}\right) - \lambda_0}{\lambda_1} \tag{73}$$

$$u_{sw} = -(\varepsilon \text{sign}(\sigma) + k\sigma) \tag{74}$$

From Equation (38),

$$\begin{aligned} u_c &= u_{sw} + u_{eq} - u_i \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{v - c\left(\dot{\psi} - v\frac{1}{s}\right) - \lambda_0}{\lambda_1} - \frac{(v - \lambda_0)}{\lambda_1} \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{\cancel{\psi} - c\left(\dot{\psi} - v\frac{1}{s}\right) - \cancel{\lambda_0} - \cancel{\psi} + \cancel{\lambda_0}}{\lambda_1} \\ &= -(\varepsilon \text{sign}(\sigma) + k\sigma) + \frac{-c\left(\dot{\psi} - v\frac{1}{s}\right)}{\lambda_1} = -\left(\varepsilon \frac{\sigma}{|\sigma| + \xi} + k\sigma\right) + \frac{-c\left(\dot{\psi} - v\frac{1}{s}\right)}{\lambda_1} \end{aligned} \tag{75}$$

where $c \in \mathbb{R}^+, k \in \mathbb{R}^+$ and $\varepsilon \in \mathbb{R}^+$ are design parameters associated with the Sliding Mode Compensator. The results after the implementation of the compensator can be observed in Figure 7 showcasing elimination of the steady state error and improved response time.

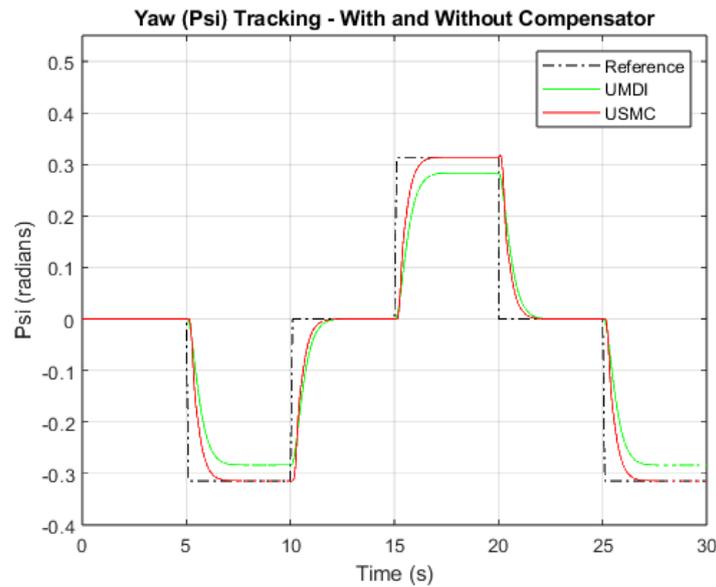


Figure 7. Comparison of the Yaw channel output with and without a compensator.

Following a similar approach, robust compensators for the Roll, pitch and Altitude channels can be designed using the design parameters in Table 2.

Table 2. Compensator design parameters.

	c	k	ξ
Altitude Z (m)	10	1	0.1
Yaw (ψ) (radians)	10	5	0.1
Roll (ϕ) (radians)	10	10	0.03
Pitch (θ) (radians)	10	10	0.03

Remark 4. Equation (75) gives the compensated controller output u_c , where u_{sw} represents the switching controller for driving the system to the sliding surface (or sliding boundary in this specific case), u_{eq} represents the equivalent control that ensures the system state stays inside the sliding boundary and u_i represents the ideal plant inversion calculated using the UMDI controller.

To compare the simulated performance of the designed controllers quantitatively, the Root Mean Square values of the simulated tracking errors are calculated in Table 3. Figure 8 shows the flight trajectories in 3D and 2D coordinates. Figure 9 shows the time sequence of the trajectories in X, Y, Z directions, and Yaw channel respectively. It can be observed that while both PID and the USMC controllers can drive the quadrotor to the desired trajectory, the U-dynamic inversion-based controller suffers from deteriorated performance due to unmodelled dynamics/parameters and therefore is unable to accurately track the desired trajectory.

Table 3. Root Mean Square Errors (RMSEs) in trajectory following.

	PID	UMDI	USMC
X (m)	0.1026	0.2749	0.1113
Y (m)	0.1589	0.2885	0.0860
Z (m)	0.0980	0.1159	0.1244
Roll (ϕ) (radians)	0.0025	0.0304	0.0107
Pitch (θ) (radians)	0.0021	0.0327	0.0105
Yaw (ψ) (radians)	0.0543	0.0809	0.0715

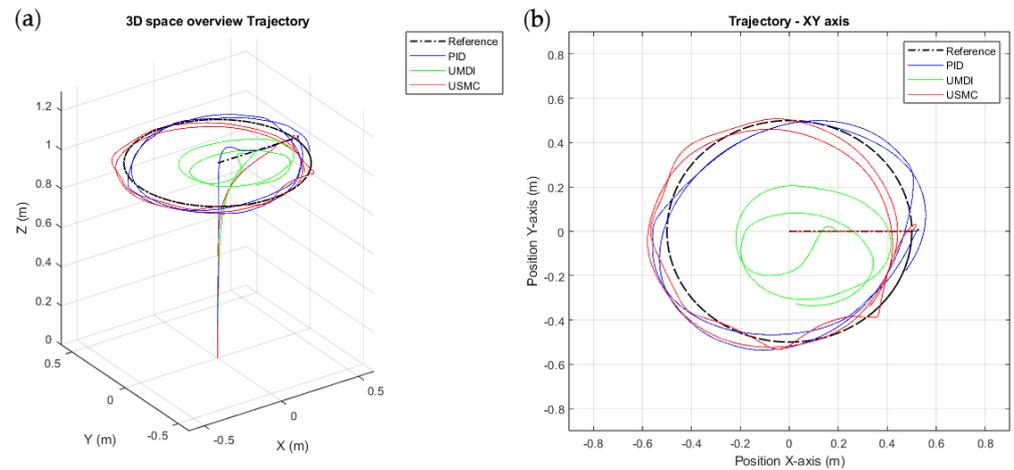


Figure 8. (a) The reference and quadrotor flight trajectory in 3D space using the designed controllers. (b) A 2D overview of the flight trajectory.

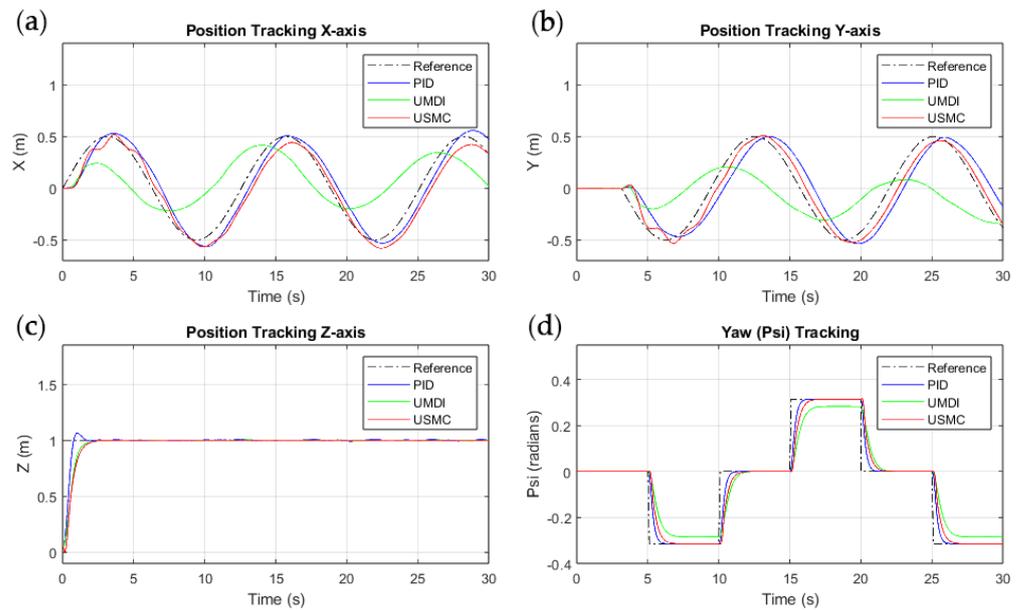


Figure 9. (a) Comparison of positioning in X coordinate. (b) Comparison of positioning in Y coordinate. (c) Comparison of positioning in Z coordinate. (d) Comparison of trajectory tracking in Yaw channel.

Flight tests to verify the simulation results were performed on the Parrot Mambo Minidrone using the same controller parameters. While the fully actuated subsystem (Altitude and Yaw) showed good tracking (Figure 10), the position tracking was unstable and exhibited very high chattering, suggesting that the measures taken while designing the controller, e.g., the introduction of the sigmoid function to replace the discontinuous function $sign(\sigma)$ to reduce system chattering, were inadequate. The outer loop that generates the desired pitch and Roll angles for position control (Equation (63)) uses the Altitude controller output terms u_1, u_z and Yaw ψ ; chattering in these terms generates unstable trajectories, causing the minidrone to become unstable. Since the minidrone is a very light weight and has a small test platform with very low inertia, it cannot handle such mechanical chattering and therefore becomes unstable. However, the designed controller did show improved performance in both Yaw and Altitude channels, reducing the steady state error and achieving an overall smoother hover performance, as can be seen in Figure 11.

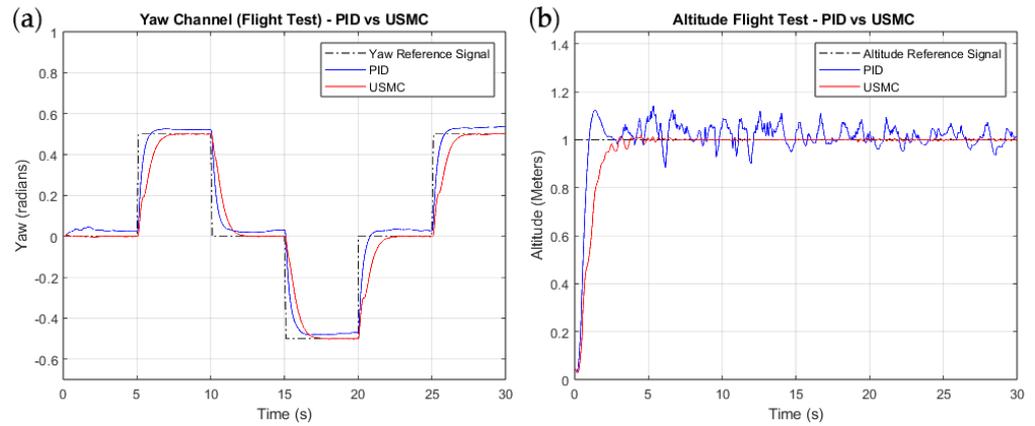


Figure 10. Performance of proposed USMC controller compared with a tuned PID controller. (a) Yaw channel. (b) Altitude channel with hovering command.

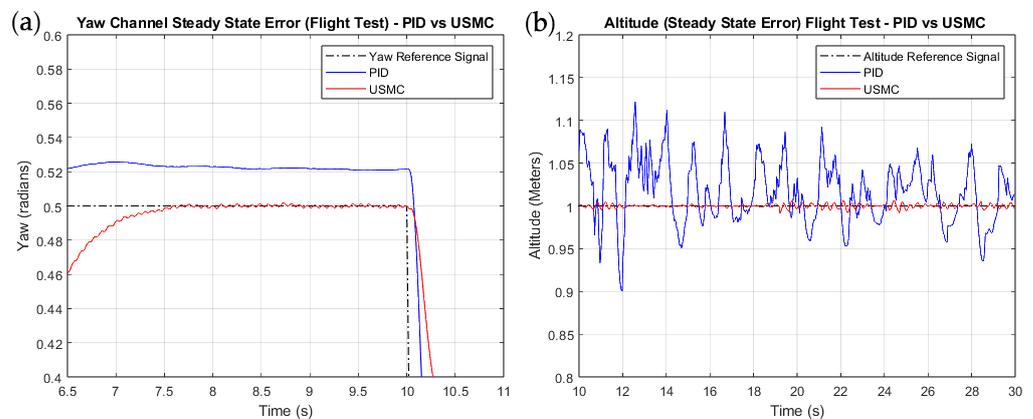


Figure 11. USMC approach results compared with a tuned PID controller. (a) An enlarged view of the Yaw channel showing the steady state error. (b) An enlarged view of the Altitude channel in hovering mode comparing the two controller performances.

Table 4 shows the effectiveness of these experimental results and compares them to the simulated results from earlier by using the RMS errors. It can further be observed from Figure 12 that the experimental results follow the simulated results more closely using the proposed approach when compared to a PID controller. This can be partially explained by the fact that the PID gains that were tuned for simulations need further tuning for flight tests, i.e., the performance of PID controllers varies in simulated and real flight tests. In contrast, the proposed approach after simulations is quite straightforward and does not need any further modifications for successful implementation during flight tests.

Table 4. Root Mean Square Errors (RMSEs) in trajectory following for real flight tests.

	PID	USMC
Z (m)	0.0501	0.0138
Yaw (ψ) (radians)	0.0284	0.0177

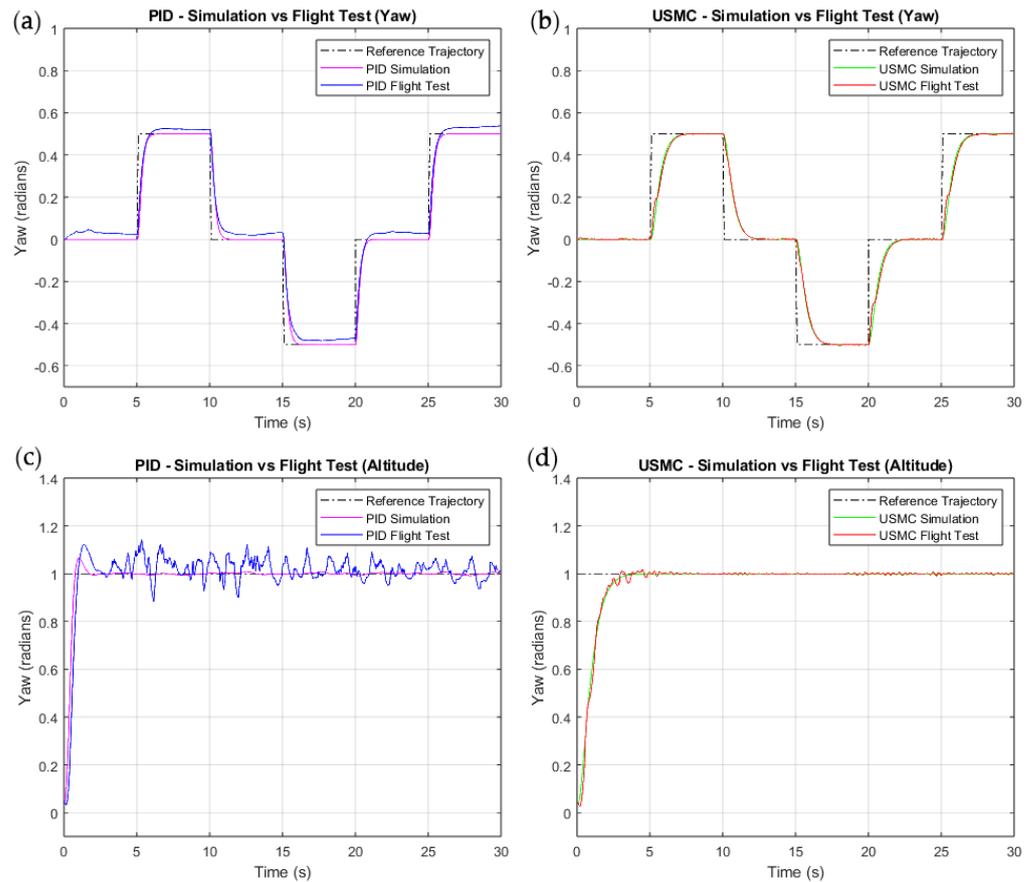


Figure 12. Comparison of simulation and flight test performance for USMC approach and tuned PID controller. (a) PID simulation result compared to flight test result for Yaw channel. (b) USMC simulation result compared to flight test result for Yaw channel. (c) PID simulation result compared to flight test result for Altitude channel. (d) USMC simulation result compared to flight test result for Altitude channel.

4. Discussions

Comparing the results of a tuned PID controller, a UMDI controller and the compensated robust USMC, it can be concluded that the UMDI controller results in deteriorated performance in the presence of unmodelled disturbances, but the addition of a robust compensator improves the performance, as can be seen in Table 3 where the RMSE is taken to statistically analyse the performance on controllers. It can be concluded that if a system is modelled accurately without outside disturbance, the UMDI-based controller is effective since there is no model-inversion mismatch and, therefore, $G_p \cdot G_p^{-1} = 1$. However, real dynamic systems always have some outside interference and unmodelled dynamics; therefore, these disturbances and model inaccuracies need to be taken into account while designing a controller. Therefore, some sort of a compensator is a must while using this approach to impart some robustness to the system. The Sliding Mode-based Compensator, while ensuring robustness, introduces chattering. This is partly fixed by using a Quasi-Sliding Mode control to approximate the $sign(*)$ function; however, that does not drive the dynamics to the sliding surface but rather to a sliding boundary, therefore possibly affecting system performance. Other possible ways to avoid chattering effects need to be studied and integrated with U-control possibly with a higher-order sliding mode approach like super twisting sliding mode. The resultant simulation plots show that the new procedure successfully addresses the challenges involved, like system underactuation, coupling, unmodelled dynamics and disturbances.

There are some warnings that one needs to be aware of in the implementation of the model-based U-control for UAV:

1. The actuator saturation could limit the SMC for large control inputs.
2. Comparing the model-free trial-and-error PID (which is time-consuming in the gain tuning and difficult to achieve desired performance) embedded in the Simulink, the model-based U-control requires a nominal model for each different type of UAV, so it requires more design and implementation expertise/cost in calibration.
3. More effective model-free control with merits in concise implementation and easy robust tuning could be considered in the next stage of this research.

5. Conclusions

This paper integrates a U-model-based dynamic inversion with an SMC compensator to control the trajectory of a quadrotor system flight operation. The designed controller shows strong stability and accurate setpoint tracking, even facing the model uncertainties.

On the whole, after comparing the results and relaxing implementation constraints, the built-in PID control is still preferred over advanced approaches due to its simplicity, even if tuning it involves some trial-and-error and guesstimating. The unmodelled dynamics and especially the external disturbances in the Roll and pitch channels destabilize the system and deteriorate its performance. This can affect the system performance, greatly preventing its implementation on physical systems. While the designed compensator improves the performance of the controller greatly, it might not be a feasible solution when running hardware-in-loop tests due to the chattering effects introduced by sliding mode control. Therefore, to ensure robustness along with practical applicability there is a need to integrate a chattering-free (or reduced chattering) alternative controller that can provide robustness and improve performance. Possible future work should be focused on this where the bounded disturbances/model inaccuracies can be lumped together and studied. Another alternative would be to implement a chattering-free robust controller like a super twisting [28] controller design in future work. Model-free control approaches [30–32] can also provide a subject for exploration and their integration with U-control can improve overall performance and system design simplicity. Furthermore, comprehensive tests of the different UAVs are on the cards for future studies.

Author Contributions: Conceptualization, Q.Z. and H.N.; methodology, A.L.; software, A.L.; validation, A.L. and Q.Z.; formal analysis, A.L., Q.Z. and H.N.; investigation, A.L., Q.Z. and H.N.; resources, A.L.; writing—original draft preparation, A.L.; writing—review and editing, A.L. and Q.Z.; visualization, A.L.; supervision, Q.Z., P.M. and H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors wish to acknowledge the editors and the anonymous reviewers for their contributions in offering helpful feedback and constructive suggestions to improve this paper.

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

Abbreviations

The study has used the following abbreviations throughout the manuscript:

SISO	single-input single-output.
MIMO	multi-input multi-output.
UMDI	U-model based dynamic inverse
USMC	U-dynamic inversion + U-Sliding Mode Control

References

- Bouabdallah, S.; Noth, A.; Siegwart, R. PID vs. LQ control techniques applied to an indoor micro quadrotor. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; pp. 2451–2456. [\[CrossRef\]](#)
- Koszewnik, A. The Parrot UAV Controlled by PID Controllers. *Acta Mech. Autom.* **2014**, *8*, 65–69. [\[CrossRef\]](#)
- Argentim, L.M.; Rezende, W.C.; Santos, P.E.; Aguiar, R.A. PID, LQR and LQR-PID on a quadcopter platform. In Proceedings of the 2013 International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 17–18 May 2013; pp. 1–6. [\[CrossRef\]](#)
- Kaplan, M.R.; Eraslan, A.; Beke, A.; Kumbasar, T. Altitude and Position Control of Parrot Mambo Minidrone with PID and Fuzzy PID Controllers. In Proceedings of the 2019 11th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 28–30 November 2019; pp. 785–789. [\[CrossRef\]](#)
- Cao, G.; Lai, E.M.K.; Alam, F. Gaussian Process Model Predictive Control of an Unmanned Quadrotor. *J. Intell. Robot. Syst.* **2017**, *88*, 147–162. [\[CrossRef\]](#)
- Madani, T.; Benallegue, A. Backstepping Sliding Mode Control Applied to a Miniature Quadrotor Flying Robot. In Proceedings of the IECON 2006—32nd Annual Conference on IEEE Industrial Electronics, Paris, France, 7–10 November 2006; pp. 700–705. [\[CrossRef\]](#)
- Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2520–2525. [\[CrossRef\]](#)
- Zhu, Q.M.; Zhao, D.; Zhang, J. A general U-block model-based design procedure for nonlinear polynomial control systems. *Int. J. Syst. Sci.* **2016**, *47*, 3465–3475. [\[CrossRef\]](#)
- Zhu, Q.M.; Liu, L.; Zhang, W.; Li, S. Control of Complex Nonlinear Dynamic Rational Systems. *Complexity* **2018**, *2018*, 8953035. [\[CrossRef\]](#)
- Li, R.; Zhu, Q.; Kiely, J.; Zhang, W. Algorithms for U-Model-Based Dynamic Inversion (UM-Dynamic Inversion) for Continuous Time Control Systems. *Complexity* **2020**, *2020*, 3640210. [\[CrossRef\]](#)
- Zhu, Q.; Zhang, W.; Na, J.; Sun, B. U-model based Control Design Framework for Continuous-Time Systems. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 106–111. [\[CrossRef\]](#)
- Slotine, J.J.E.; Li, W.P. *Applied Nonlinear Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1991.
- Zhang, L.L.; Deng, C.; Che, W.W.; An, L. Adaptive backstepping control for nonlinear interconnected systems with prespecified-performance-driven output triggering. *Automatica* **2023**, *154*, 111063. [\[CrossRef\]](#)
- Duan, G.R. High-order fully actuated system approaches: Part I: Models and basic procedure. *Int. J. Syst. Sci.* **2021**, *52*, 422–435. [\[CrossRef\]](#)
- Echiheb, F.; Bossoufi, B.; Kafazi, I.E.; Bhiri, B.E. A review backstepping control of a DFIG-based wind power system. In *Digital Technologies and Applications*; Motahhir, S., Bossoufi, B., Eds.; ICDTA 2023, Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 668. [\[CrossRef\]](#)
- Zhang, W.; Zhu, Q.; Mobayen, S.; Yan, H.; Qiu, J.; Narayan, P. U-Model and U-Control Methodology for Nonlinear Dynamic Systems. *Complexity* **2020**, *2020*, 1050254. [\[CrossRef\]](#)
- Wei, Z.; Li, R.; Zhu, Q. Sliding Mode plus U-Control method for continuous time SISO dynamic systems. *Int. J. Cybern. -Cyber-Phys. Syst.* **2021**, *1*, 68–92. [\[CrossRef\]](#)
- Jategaonkar, R.V. *Flight Vehicle System Identification: A Time-Domain Methodology*, 2nd ed.; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2015. [\[CrossRef\]](#)
- Li, J.; Li, Y. Dynamic analysis and PID control for a quadrotor. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August 2011; pp. 573–578. [\[CrossRef\]](#)
- Zhu, Q.M.; Guo, L. A pole placement controller for non-linear dynamic plants. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2002**, *216*, 467–476. [\[CrossRef\]](#)
- Geng, X.P.; Zhu, Q.M.; Liu, T.; Na, J. U-model based predictive control for nonlinear processes with input delay. *J. Process. Control* **2019**, *75*, 156–170. [\[CrossRef\]](#)
- Hussain, N.A.A.; Ali, S.S.A.; Ovinis, M.; Arshad, M.; Al-saggaf, U.M. Underactuated coupled nonlinear adaptive control synthesis using U-model for multivariable unmanned marine robotics. *IEEE Access* **2020**, *8*, 1851–1965. [\[CrossRef\]](#)
- Boubezoula, M.; Hassam, A.; Boutalbi, O. Robust-flatness Controller Design for a Differentially Driven Wheeled Mobile Robot. *Int. J. Control Autom. Syst.* **2018**, *16*, 1895–1904. [\[CrossRef\]](#)
- Sánchez-Meza, V.; Lozano-Hernández, Y.; Gutiérrez-Frías, O.; Lozada-Castillo, N.; Luviano-Juárez, A. Robust Flat Filtering Control of a Two Degrees of Freedom Helicopter Subject to Tail Rotor Disturbances. *Int. J. Appl. Math. Comput. Sci.* **2023**, *33*, 521–535. [\[CrossRef\]](#)
- Ogata, K. *Modern Control Engineering*, 4th ed.; Prentice Hall PTR: Hoboken, NJ, USA, 2001.
- Åström, K.J.; Wittenmark, B. *Adaptive Control*, 2nd ed.; Dover Publications: Mineola, NY, USA, 2008; pp. 92–98. ISBN 9780486462783.
- Billings, S. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*; Wiley: Hoboken, NJ, USA, 2013.
- Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. *Sliding Mode Control and Observation*; Birkhäuser: New York, NY, USA, 2013; pp. 1–41. [\[CrossRef\]](#)

29. Li, R.; Zhu, Q.; Nemati, H.; Yue, X.; Narayan, P. Trajectory tracking of a quadrotor using extend state observer based U-model enhanced double sliding mode control. *J. Frankl. Inst.* **2023**, *360*, 3520–3544. [[CrossRef](#)]
30. Fliess, M.; Join, C. Model-free control. *Int. J. Control* **2013**, *86*, 2228–2252. [[CrossRef](#)]
31. Zhu, Q. Model-free sliding mode enhanced proportional, integral, and derivative (SMPID) control. *Axioms* **2023**, *12*, 721. [[CrossRef](#)]
32. Zhu, Q.; Li, R.; Zhang, J. Model-free robust decoupling control of nonlinear nonaffine dynamic systems. *Int. J. Syst. Sci.* **2023**, *54*, 2590–2607. [[CrossRef](#)]

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.