


Article

# A Robust Disturbance-Rejection Controller Using Model Predictive Control for Quadrotor UAV in Tracking Aggressive Trajectory

Zhixiong Xu <sup>1</sup> , Li Fan <sup>1,2,\*</sup>, Wei Qiu <sup>1</sup>, Guangwei Wen <sup>2</sup> and Yunhan He <sup>1</sup>

<sup>1</sup> College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China; 22232005@zju.edu.cn (Z.X.); cyanilius@yeah.net (W.Q.); heyunhan@zju.edu.cn (Y.H.)

<sup>2</sup> Huzhou Institute, Zhejiang University, Huzhou 313000, China; wengw@hizju.org

\* Correspondence: fanli77@zju.edu.cn

**Abstract:** A robust controller for the waypoint tracking of a quadrotor unmanned aerial vehicle (UAV) is proposed in this paper, in which position control and attitude control are effectively decoupled. Model predictive control (MPC) is employed in the position controller. The constraints of motors are imposed on the state and input variables of the optimization equation. This design effectively mitigates the nonlinearity of the attitude loop and enhances the planning efficiency of the position controller. The attitude controller is designed using a nonlinear and robust control law based on  $SO(3)$  space, which enables continuous control on the  $SO(3)$  manifold. By extending the differential flatness of the quadrotor-UAV to the angular acceleration level, the mapping of the control reference from the position controller to the attitude controller is achieved. Simulations are carried out to demonstrate the capability of the proposed controller. In the simulations, multiple aggressive flight trajectories and severe external disturbances are designed. The results show that the controller is robust, with superior accuracy in tracking aggressive trajectories.

**Keywords:** waypoint tracking; model predictive control; nonlinear attitude controller; differential flatness



**Citation:** Xu, Z.; Fan, L.; Qiu, W.; Wen, G.; He, Y. A Robust Disturbance-Rejection Controller Using Model Predictive Control for Quadrotor UAV in Tracking Aggressive Trajectory. *Drones* **2023**, *7*, 557. <https://doi.org/10.3390/drones7090557>

Academic Editor: Vaios Lappas

Received: 30 June 2023

Revised: 10 August 2023

Accepted: 21 August 2023

Published: 29 August 2023



**Copyright:** © 2023 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

Nowadays, quadrotor UAVs have been widely employed in various applications, such as agricultural plant protection [1], transportation and logistics distribution [2], and post-disaster rescue operations [3]. To meet the demands of different missions and stay robust under challenging environmental conditions, the control methods for quadrotor UAVs have been continuously developed.

In a general task, the controller is responsible for accurately tracking a time-dependent trajectory designated by the planner. As an underactuated system, a quadrotor-UAV has far more state variables than input variables, and the nonlinear characteristics of the system make the control problem challenging. The cascaded PID control method is commonly utilized for quadrotor-UAV control, in which the position of the next time step is considered a reference input. The inner loop controls the attitude, and the outer loop takes charge of the position. The mapping from the position loop to the attitude loop is completed using the differential-flatness attribute of quadrotor UAVs [4–6]. However, this error-based control method often performs well while tracing a smooth and slow trajectory, but struggles in tracking an aggressive trajectory.

To tackle the problem, a variety of advanced control algorithms for quadrotor UAVs have been proposed, among which MPC has received significant attention. The MPC method constructs an optimization problem based on the quadrotor-UAV model and applies constraints on the states and inputs of the system, and it generates control commands over the predicted time horizon by solving the optimization problem. References [7–9] have

demonstrated the exceptional control efficiency of this approach. Furthermore, linearized model predictive control (LMPC) and nonlinear model predictive control (NMPC) have been developed and compared in many studies. In the former method, Taylor expansion is generally conducted around the hover point to linearize the quadrotor-UAV model. Linearization saves the computation cost, but downgrades the performance in terms of aggressive trajectory tracking, as the higher-order terms are truncated in the model. On the contrary, the nonlinear model is employed in NMPC for predictive control, resulting in superior tracking accuracy and robustness. However, the optimization offered by this method is usually non-convex, which leads to longer computational time and risks of solver non-convergence. In addition, the reinforcement learning-based control methods have been demonstrated to be effective for quadrotor-UAV control [10–13]. Han et al. proposed a robust controller based on a hierarchical control framework and reinforcement learning [14]. A robust control performance is achieved without prior knowledge of quadrotor-UAV dynamics. Kaufmann et al. showcased the application of learning-based approaches in acrobatic quadrotor-UAV flights [12]. However, the end-to-end training approach limits the reusability and generalization of the method.

As the quadrotor-UAV is an underactuated system, the state variables are coupled with each other, which makes it difficult to accurately track all the states. The accuracy of attitude tracking is directly related to the accuracy of position tracking. Therefore, attitude control for quadrotor UAVs has been widely investigated in the literature. Traditional attitude control methods for quadrotor-UAVs often rely on Euler angles to represent the attitude, where each Euler angle is individually tracked to achieve attitude control. However, Euler angles reach singularity at specific attitudes. To avoid these issues, quaternion is employed to represent the attitude of quadrotor UAVs, as can be seen in references [15–18], resulting in improved attitude control performance. In geometric theory, the changes in attitude are correlated to the evolution of the  $SO(3)$  manifold. Therefore, Lee et al. and Yu et al. designed attitude controllers based on Lie groups on  $SO(3)$  space [19–21], which enables continuous control on  $SO(3)$  space. Also, with the representation of the attitude using this method, singularities are avoided during the control process. What is more, many researchers integrated the classical robust control algorithms into attitude control for robustness. For instance, Lee et al. [22] came up with an adaptive control algorithm. An adaptive law is employed to provide robust control with bounded errors even in the presence of unknown quadrotor-UAV internal parameters. Considering that quadrotor-UAVs are often subjected to external disturbances during flight, sliding-mode control algorithms are employed for attitude control in references [23–25]. By incorporating sliding-mode-control terms into the control laws, the system disturbance rejection capability is, therefore, enhanced, and asymptotic convergences are ensured.

In most of the missions for a quadrotor-UAV, the objective of control is accurately tracking the position. The control of the attitude loop is essentially to fulfill the demands of the position loop. Most control approaches usually employ the entire dynamics model of quadrotor UAVs in the optimization process for predictive control, which introduces challenges due to the nonlinear nature of the attitude loop. However, if the desired values can be accurately traced in the attitude loop, the attitude loop can be decoupled from the position loop in the MPC optimization process. The nonlinearity in the attitude loop can, therefore, be avoided. Based on this idea, this paper aims to develop a quadrotor-UAV controller that can accurately track waypoints with temporal information. Contributions to this paper are listed as follows:

- (1) A robust controller for quadrotor-UAV waypoint tracking is proposed, in which position control and attitude control are decoupled. The control efficiency is enhanced while maintaining robustness.
- (2) The LMPC algorithm is used in position control, where the motor constraints are incorporated into both the input and state variables of the optimization problem. This approach effectively mitigates the nonlinear characteristics of the attitude loop and enhances the planning efficiency of the position controller.

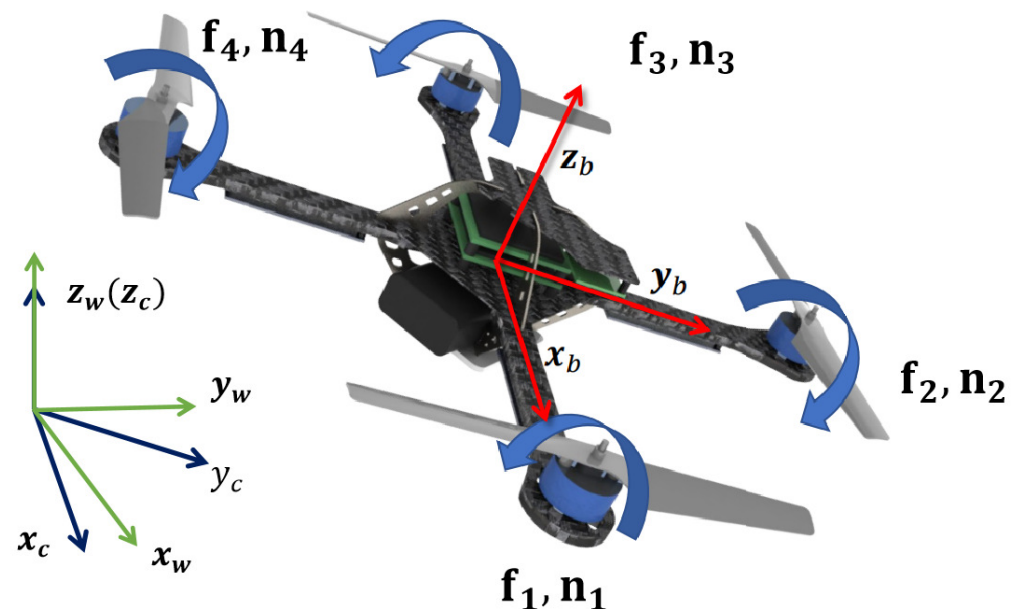
- (3) For the attitude controller, a nonlinear robust control law is designed based on the  $SO(3)$  space, which guarantees global asymptotic stability in the sense of Lyapunov.
- (4) Extensive simulations have verified the excellent control performance of the proposed controller, which can achieve high-precision tracking of the desired trajectory even under severe external disturbances.

The paper is divided into five sections. In Section 2, the model of the quadrotor-UAV as well as the coordinate systems used in current research are introduced. In Section 3, the framework of the proposed controller is elaborated. Descriptions of each module are provided in detail. In Section 4, control performances that have been demonstrated by multiple numerical simulations are provided. Finally, summaries and conclusions are provided in Section 5.

## 2. Model

In this article, variables are defined in a unified format. Matrices and vectors are denoted by bold non-italic capital letters and bold non-italic lowercase letters, respectively. Scalars are written in italic letters. The subscripts of a vector represent the components of that vector. For example,  $v_{wx}$ ,  $v_{wy}$  and  $v_{wz}$  are the components of the velocity vector along the three axes in the world coordinate system, respectively.

The coordinate systems employed in this paper are the world system  $\mathcal{W}$  and the body system  $\mathcal{B}$ . For the body coordinate system, the origin is located at the center of mass of the vehicle. The  $x$ -axis and  $y$ -axis are parallel to the two arms of the quadrotor-UAV as shown in Figure 1 and it is a right-handed system. For the world coordinate system, the  $z$ -axis points in the opposite direction of Earth's gravity. Similar to the paper by Mellinger and Kumar [26], an intermediate coordinate system  $\mathcal{C}$  between the system  $\mathcal{W}$  and system  $\mathcal{B}$  is defined, it shares the same  $z$ -axis as the world coordinate system. The numbering of the four motors is illustrated in Figure 1.



**Figure 1.** Coordinate system definition and motor numbering.

The propulsion system of the quadrotor-UAV is modeled as a second-order model. Following the discussion in reference [26], it is assumed that the force  $f_i$  and torque  $n_i$  produced by each motor are proportional to the square of its rotational speed  $\omega_i$ , as indicated in Equation (1). Within the equation, both  $k_f$  and  $k_n$  are constants, while  $k_m$  stands

for the motor response constant. In addition, considering the issue of motor saturation, there exists a maximum rotational speed constraint for each motor.

$$\ddot{\omega}_i = k_m(\omega_i^d - \omega_i), \quad f_i = k_f \omega_i^2, \quad n_i = k_n \omega_i^2 \tag{1}$$

$$0 \leq \omega_i \leq \omega_{\max} \tag{2}$$

In the body coordinate system, the thrust  $T$  and the three-axis torque  $\tau$  produced by the four motors are

$$\begin{bmatrix} \tau \\ T \end{bmatrix} = \mathbf{D}\boldsymbol{\omega}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{3}$$

$$\mathbf{D} = \begin{bmatrix} 0 & k_f l & 0 & -k_f l \\ -k_f l & 0 & k_f l & 0 \\ -k_n & k_n & -k_n & k_n \\ k_f & k_f & k_f & k_f \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{4}$$

where  $l$  represents the length from the motor’s rotational axis to the center of mass of the quadrotor-UAV.

Following the convention, all angular velocities are represented in the body frame, while position and velocity vectors are represented in the world frame. The quadrotor-UAV is subjected to the forces of gravity and the thrust generated by the motors along the z-axis of the body frame. Newton’s second law related to the position control is

$$m\mathbf{a} = -mg\mathbf{z}_w + T\mathbf{z}_b \tag{5}$$

in which  $\mathbf{z}_w$  and  $\mathbf{z}_b$  indicate the unit vectors of the z-axis for the world and body frames. The mass of the quadrotor-UAV is  $m$  and the gravitational acceleration is  $g$ . The Euler equation related to the attitude control is

$$\mathbf{J}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \mathbf{J}\boldsymbol{\omega}_b = \boldsymbol{\tau} \tag{6}$$

where  $\mathbf{J}$  denotes the inertia tensor. The roll, pitch, and yaw angles ( $\phi$ ,  $\theta$  and  $\psi$ ) based on the Z-X-Y Euler angle sequence are employed to represent the attitude. The relation between the first derivative of Euler angles and the angular velocities in the body frame is

$$\begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\cos \phi \sin \theta \\ 0 & 1 & \sin \phi \\ \sin \theta & 0 & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{7}$$

### 3. Methodology

For the proposed controller, a sequence of reference temporal waypoints is the control input. The information of each waypoint mainly consists of the three-dimensional (3D) position and yaw angle  $\begin{bmatrix} p_{wx}^d & p_{wy}^d & p_{wz}^d & \psi^d \end{bmatrix}$ . Figure 2 provides the framework of the controller. The LMPC algorithm is employed in the position loop. The relevant model constraints are implemented on the state and input variables, thus planning the reference trajectory within a future time domain. Based on the differential flatness property of the quadrotor-UAV, the planned trajectory in the position loop can be mapped to the desired input for the attitude controller. In addition, an attitude control law is designed based on the  $SO(3)$  space. The nonlinear robust control law achieves accurate tracking of the desired attitude angles. Regarding the control allocation of the quadrotor-UAV, an inverse kinematics solution is obtained based on the optimization approach. Each module will be described in detail in the following sections.

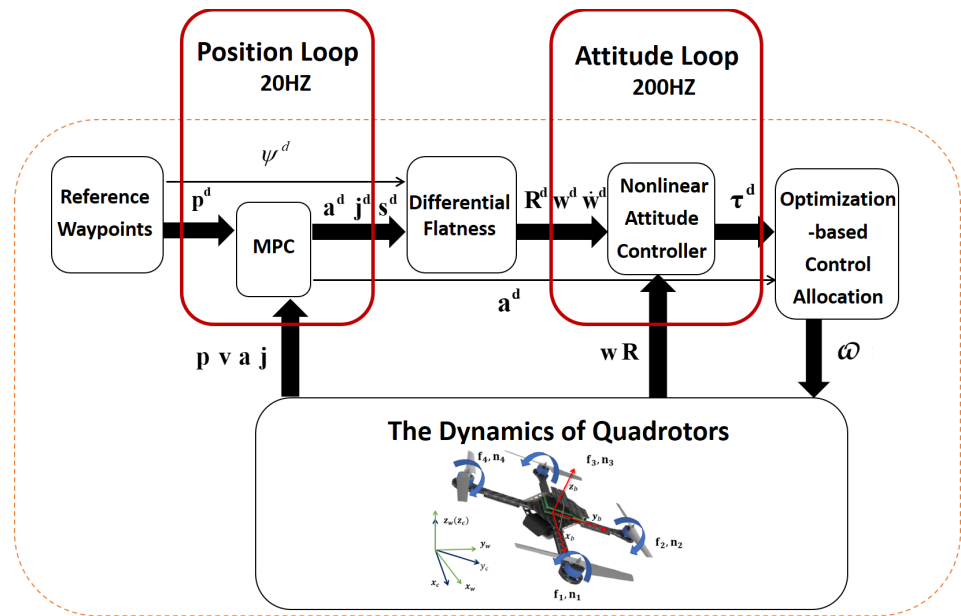


Figure 2. Framework of proposed the quadrotor-UAV controller based on LMPC.

### 3.1. Linear Model Predictive Control

The ability to predict over a wide time domain and consider the constraints of an actual model are the reasons why MPC is widely employed. In this paper, the LMPC is employed in the position controller, and the model constraints primarily incorporate considerations of motor speed limitations and motor response constraints. At each time step, the optimal control sequence for the entire prediction horizon is computed, but only the first control input is applied to the system. This process is iterated in a receding horizon manner, where optimization is performed repeatedly at each time step.

As shown in Figure 2, the selected state variables are the position, velocities, accelerations, and jerks along three axes. These variables are denoted by their respective initials.

$$x = \begin{bmatrix} p \\ v \\ a \\ j \end{bmatrix}, y = [p], u = [s], r = [p^d] \tag{8}$$

The output of the system  $y$  is the 3D position, and the control input is the snap, which is the second derivative of acceleration. The reference input  $r$  corresponds to the desired position. Among these variables,  $p, v, a, j, s \in \mathbb{R}^3, y \in \mathbb{R}^3, u \in \mathbb{R}^3, r \in \mathbb{R}^3$ . The reasons for optimizing for the snap will be thoroughly analyzed in the subsequent sections.

The corresponding difference equation can be formulated based on the aforementioned input and state variables.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k+1) &= Cx(k+1) \end{aligned} \tag{9}$$

where  $A, B,$  and  $C$  are defined as

$$A = \begin{bmatrix} I_{3 \times 3} & Ts * I_{3 \times 3} & \frac{1}{2} Ts^2 * I_{3 \times 3} & \frac{1}{6} Ts^3 * I_{3 \times 3} \\ 0 & I_{3 \times 3} & Ts * I_{3 \times 3} & \frac{1}{2} Ts^2 * I_{3 \times 3} \\ 0 & 0 & I_{3 \times 3} & Ts * I_{3 \times 3} \\ 0 & 0 & 0 & I_{3 \times 3} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{24} Ts^4 * I_{3 \times 3} \\ \frac{1}{6} Ts^3 * I_{3 \times 3} \\ \frac{1}{2} Ts^2 * I_{3 \times 3} \\ Ts * I_{3 \times 3} \end{bmatrix}, C = [ I_{3 \times 3} \ 0 \ 0 \ 0 ] \tag{10}$$

in which  $T_s$  represents the discrete time interval, and  $\mathbf{I}_{3 \times 3}$  denotes the identity matrix. Given the prediction horizon  $N$ , we define the state vector  $\mathbf{X}$ , the reference input vector  $\mathbf{\Gamma}$ , the output vector  $\mathbf{Y}$ , and the control input vector  $\mathbf{U}$  over the predicted time horizon.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}(k+2) \\ \mathbf{x}(k+3) \\ \vdots \\ \mathbf{x}(k+N) \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \mathbf{u}(k+2) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \\ \mathbf{y}(k+3) \\ \vdots \\ \mathbf{y}(k+N) \end{bmatrix} \quad \mathbf{\Gamma} = \begin{bmatrix} \mathbf{r}(k+1) \\ \mathbf{r}(k+2) \\ \mathbf{r}(k+3) \\ \vdots \\ \mathbf{r}(k+N) \end{bmatrix} \quad (11)$$

The relationship between them can be expressed as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}^2\mathbf{B} & \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \mathbf{A}^{N-3}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \mathbf{U} + \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ \vdots \\ \mathbf{A}^N \end{bmatrix} \mathbf{x}(k) = \mathbf{PU} + \mathbf{Mx}(k) \quad (12)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C} \end{bmatrix} \mathbf{X} = \mathbf{EX} \quad (13)$$

$$\Delta\mathbf{U} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{U} + \begin{bmatrix} -\mathbf{I}_{3 \times 3} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k-1) = \mathbf{SU} + \mathbf{Hu}(k-1) \quad (14)$$

where  $\mathbf{x}(k)$  represents the state values at time step  $k$ , and  $\mathbf{u}(k-1)$  represents the control input at the previous time step.

Considering the limitations of the control inputs and the tracking accuracy, a problem aiming to minimize tracking errors and control input variations can be formulated as

$$\min_{\mathbf{U}} (\mathbf{\Gamma} - \mathbf{Y})^T \mathbf{G}_1 (\mathbf{\Gamma} - \mathbf{Y}) + \Delta\mathbf{U}^T \mathbf{G}_2 \Delta\mathbf{U} \quad (15)$$

where  $\mathbf{G}_1$  is the weight matrix related to tracking errors, and  $\mathbf{G}_2$  is the weight matrix related to the control input. Both  $\mathbf{G}_1$  and  $\mathbf{G}_2$  are positive definite diagonal matrices.

The model constraints considered in the controller are the motor speed constraints and motor response constraints. Combining Equations (1) and (5), we obtain

$$m\mathbf{a} = \mathbf{z}_b \sum_{i=1}^4 k_f \omega_i^2 - mg\mathbf{z}_w \quad (16)$$

The motor's maximum speed constraint is imposed on the acceleration term of the state variables.  $[\cdot]_{\max}$  and  $[\cdot]_{\min}$  represent the maximum and minimum values of each component in the vector, respectively. The attitude of quadrotor-UAV is assumed to be unchanged within a single time step. This assumption may lead to motor speed saturation in some extreme cases. However, the optimization-based control allocation approach significantly reduces the impact of motor saturation.

$$\mathbf{a}_{\min} = \left[ \frac{\mathbf{z}_b}{m} \sum_{i=1}^4 k_f \omega_i^2 - g \mathbf{z}_w \right]_{\min} \quad \mathbf{a}_{\max} = \left[ \frac{\mathbf{z}_b}{m} \sum_{i=1}^4 k_f \omega_i^2 - g \mathbf{z}_w \right]_{\max} \quad (17)$$

By taking the second derivative of both sides of Equation (5) and applying the motor response constraints to the snap term of the input variables, we have

$$m\mathbf{s} = 2\mathbf{z}_b \sum_{i=1}^4 k_f (\omega_i \ddot{\omega}_i + \dot{\omega}_i^2) \quad (18)$$

$$\mathbf{s}_{\min} = \frac{2\mathbf{z}_b}{m} \sum_{i=1}^4 k_f (\omega_i \ddot{\omega}_{i,\min} + \dot{\omega}_i^2), \quad \mathbf{s}_{\max} = \frac{2\mathbf{z}_b}{m} \sum_{i=1}^4 k_f (\omega_i \ddot{\omega}_{i,\max} + \dot{\omega}_i^2) \quad (19)$$

$$\ddot{\omega}_{i,\min} = k_m (\omega_{\min}^d - \omega_i), \quad \ddot{\omega}_{i,\max} = k_m (\omega_{\max}^d - \omega_i) \quad (20)$$

where  $\ddot{\omega}_{i,\min}$  and  $\ddot{\omega}_{i,\max}$  represent the minimum and maximum motor responsiveness, respectively. Additional constraints, such as velocity limitations, can also be incorporated into the current algorithm. By substituting the variables in Equation (15), an optimization problem is formulated as

$$\begin{aligned} \min_{\mathbf{U}} & \mathbf{U}^T (\mathbf{P}^T \mathbf{E}^T \mathbf{G}_1 \mathbf{E} \mathbf{P} + \mathbf{S}^T \mathbf{G}_2 \mathbf{S}) \mathbf{U} + 2(\mathbf{x}(k))^T \mathbf{M}^T \mathbf{E}^T \mathbf{G}_1 \mathbf{E} \mathbf{P} - \mathbf{\Gamma}^T \mathbf{G}_1 \mathbf{E} \mathbf{P} + \mathbf{u}(k-1)^T \mathbf{H}^T \mathbf{G}_2 \mathbf{S} \mathbf{U} \\ \text{s.t.} & \mathbf{a}_{\min} \leq \mathbf{a} \leq \mathbf{a}_{\max} \\ & \mathbf{U}_{\min} \leq \mathbf{U} \leq \mathbf{U}_{\max} \end{aligned} \quad (21)$$

It is worth mentioning that the aforementioned optimization problem can be simplified into a quadratic programming problem due to the absence of nonlinearity in the attitude loop. As a result, it can be solved quickly in real-time. Through solving this optimization equation, the desired position, velocity, acceleration, jerk, and snap can be effectively planned within the predicted time horizon. The first control sequence of the optimization results can then be applied to the controller in the next time step.

### 3.2. Nonlinear Attitude Controller Based on SO(3)

As errors in attitude significantly undermine the position tracking accuracy, an attitude controller that can achieve high-precision tracking is very important to the overall controller. In this paper, a nonlinear and robust attitude control law is designed based on the SO(3) space, which ensures the attitude control is continuous on the SO(3) manifold. Furthermore, the proposed attitude controller is proved to be global asymptotically stable in the sense of Lyapunov.

The Euler equation for the attitude control is

$$\mathbf{J} \dot{\mathbf{w}}_b + \mathbf{w}_b \times \mathbf{J} \mathbf{w}_b = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{dis}} \quad (22)$$

where  $\boldsymbol{\tau}_{\text{dis}}$  represents external disturbances and  $\boldsymbol{\tau}$  is the input moment. Equation (22) can be rewritten as

$$\begin{aligned} \dot{\mathbf{w}}_b &= \mathbf{J}^{-1} \boldsymbol{\tau} + \mathbf{J}^{-1} (-\mathbf{w}_b \times \mathbf{J} \mathbf{w}_b) + \mathbf{J}^{-1} \boldsymbol{\tau}_{\text{dis}} = \boldsymbol{\eta} \mathbf{u} + \boldsymbol{\zeta} + \mathbf{h} \\ \boldsymbol{\eta} &= \mathbf{J}^{-1}, \boldsymbol{\zeta} = \mathbf{J}^{-1} (-\mathbf{w}_b \times \mathbf{J} \mathbf{w}_b), \mathbf{h} = \mathbf{J}^{-1} \boldsymbol{\tau}_{\text{dis}} \end{aligned} \quad (23)$$

where  $\mathbf{h}$  represents the external perturbation term, satisfying  $\|\mathbf{h}\|_{\infty} \leq h_{\max}$ .

According to the Euler rotation theorem, any rotation matrix of SO(3) can be equivalently achieved by rotating around a specific axis by a certain angle. Here,  ${}^w_b \mathbf{R}$  denotes the rotation matrix of the current body coordinate system with respect to the world coordinate system, while  ${}^w_b \mathbf{R}^d$  signifies the desired rotation matrix of the body coordinate system with respect to the world coordinate system. Without causing any ambiguity, they are denoted as  $\mathbf{R}$  and  $\mathbf{R}^d$  for simplicity, respectively. The error of the rotation matrix can be obtained as

$$\mathbf{R}_e = \mathbf{R}^T \mathbf{R}^d \quad (24)$$

Following the definition of the error of the rotation matrix, the angle error can be calculated as

$$\theta_e = \cos^{-1}\left(\frac{\text{tr}(\mathbf{R}_e - 1)}{2}\right) \tag{25}$$

The angular velocity error is defined as the difference between the desired angular velocity  $\mathbf{w}_b^d$  and the actual angular velocity  $\mathbf{w}_b$ :

$$\mathbf{w}_e = \mathbf{w}_b^d - \mathbf{w}_b \tag{26}$$

Hence, the attitude control law can be designed based on the angle error and the angular velocity error:

$$\mathbf{u} = \boldsymbol{\eta}^{-1}[\dot{\mathbf{w}}_b^d - \boldsymbol{\zeta} + h_{max}\text{sgn}(\mathbf{w}_e) + K_1\mathbf{w}_e + \frac{|\theta_e|\|\mathbf{w}_b^d\| + K_2\theta_e^2}{\|\mathbf{w}_e\|^2}\mathbf{w}_e + \log(\mathbf{R}_e)] \tag{27}$$

Referring to the exponential mapping and algebraic mapping between Lie groups and Lie algebras [27], the calculation of the desired rotation axis based on the error of the rotation matrix can be carried out as follows:

$$\log(\mathbf{R}_e) = \frac{\theta_e}{2 \sin \theta_e} (\mathbf{R}_e - \mathbf{R}_e^T)^\vee \in so(3) \tag{28}$$

where the operator  $\wedge$  denotes a conversion of a three-dimensional vector into a  $3 \times 3$  skew-symmetric matrix, while the operator  $\vee$  represents the inverse operation of the operator  $\wedge$ .

$$\mathbf{c} \times \mathbf{d} = \mathbf{c}^\wedge \mathbf{d}, \quad \forall \mathbf{c}, \mathbf{d} \in \mathbb{R}^3 \tag{29}$$

The term  $h_{max}\text{sgn}(\mathbf{w}_e)$  in the control law corresponds to the sliding mode control term for angular velocity control, and it effectively mitigates the influence of external disturbances. Moreover, this term exhibits negligible numerical values, thus ensuring the stability of the controller without inducing any instability concerns.

To demonstrate the robustness of the control law, a Lyapunov function is conducted as follows:

$$V = \frac{1}{2}\theta_e^2 + \frac{1}{2}\mathbf{w}_e^T \mathbf{w}_e \tag{30}$$

By taking the derivative of the Lyapunov function, we have

$$\dot{V} = \dot{\theta}_e \theta_e + \mathbf{w}_e^T \dot{\mathbf{w}}_e = \dot{V}_1 + \dot{V}_2 \tag{31}$$

The derivative of the angle term is expanded and the calculation is derived as

$$\begin{aligned} \dot{V}_1 &= \dot{\theta}_e \theta_e \\ &= \theta_e \frac{d(\cos^{-1}(\frac{\text{tr}(\mathbf{R}_e - 1)}{2}))}{dt} \\ &= -\theta_e \frac{1}{2\sqrt{1 - \cos^{-1}(\frac{\text{tr}(\mathbf{R}_e - 1)}{2})^2}} \frac{d\text{tr}(\mathbf{R}_e)}{dt} \\ &= -\frac{\theta_e}{2 \sin \theta_e} \frac{d\text{tr}(\mathbf{R}^T \mathbf{R}^d)}{dt} \\ &= -\frac{\theta_e}{2 \sin \theta_e} \text{tr}\left(\frac{d\mathbf{R}^T}{dt} \mathbf{R}^d + \mathbf{R}^T \frac{d\mathbf{R}^d}{dt}\right) \\ &= -\frac{\theta_e}{2 \sin \theta_e} \text{tr}(\dot{\mathbf{w}}_b^T \mathbf{R}^T \mathbf{R}^d + \mathbf{R}^T \mathbf{R}^d \dot{\mathbf{w}}_b^d) \\ &= \log(\mathbf{R}_e)^T \mathbf{w}_e \end{aligned} \tag{32}$$



Substituting the above equation and the control law into Equation (31) and scaling it appropriately, we have

$$\begin{aligned}
 \dot{V} &= \dot{\theta}_e \theta_e + \mathbf{w}_e^T \dot{\mathbf{w}}_e \\
 &= \log(\mathbf{R}_e)^T \mathbf{w}_e + \mathbf{w}_e^T (\dot{\mathbf{w}}_b^d - \boldsymbol{\zeta} - \mathbf{h} - \dot{\mathbf{w}}_b^d + \boldsymbol{\zeta} - h_{max} \text{sgn}(\mathbf{w}_e) \\
 &\quad - K_1 \mathbf{w}_e - \frac{|\theta_e| \|\mathbf{w}_b^d\| + K_2 \theta_e^2}{\|\mathbf{w}_e\|^2} \mathbf{w}_e - \log(\mathbf{R}_e)) \\
 &= -K_1 \mathbf{w}_e^T \mathbf{w}_e - (h_{max} \text{sum}(\|\mathbf{w}_e\|) + \mathbf{h}^T \mathbf{w}_e) - \|\mathbf{w}_b^d\| |\theta_e| - K_2 \theta_e^2 \\
 &\leq -K_1 \mathbf{w}_e^T \mathbf{w}_e - K_2 \theta_e^2 \\
 &\leq -4 \min(K_1, K_2) V
 \end{aligned} \tag{33}$$

where  $\text{sum}(\|\mathbf{w}_e\|)$  represents the sum of the absolute values of each element in  $\mathbf{w}_e$ . It is evident that the constructed Lyapunov function is positive definite, and its derivative is negative definite. Consequently, it can be concluded that the control law guarantees the global asymptotic stability of the system in the sense of Lyapunov. By adjusting the coefficients  $K_1$  and  $K_2$ , the convergence rate of the system can be tailored.

The attitude control law is designed based on the  $SO(3)$  space to avoid singularities. Concurrently, the control outputs are more efficient on the  $SO(3)$  manifold. Simulations presented in Section 4 have demonstrated the exceptional performance of the proposed attitude control law, even in the presence of disturbances and observation uncertainties. The precise tracking of the attitude loop ensures its capability to fulfill the demands of the position loop. Although the attitude loop is excluded from the MPC optimization, its nonlinearity can still be compensated by the attitude controller, thus the overall control performance remains superior.

### 3.3. Differential Flatness

Given the above control scheme, the position and attitude controller can be designed independently. By leveraging the differential flatness properties of the quadrotor-UAV, a mapping from the position loop to the attitude loop is achieved. Differential flatness is referred to as the selection of appropriate variables from the state space and describing the entire state space using these variables and their derivatives. Similar to most articles in the literature, the following flat outputs are selected, namely the 3D position and the yaw angle:

$$\sigma = [p_{wx}, p_{wy}, p_{wz}, \psi]^T \tag{34}$$

The position is planned using the LMPC method within the position loop, and the yaw angle information is pre-defined. A direct mapping from the flat outputs to the desired attitude, angular rate, and angular acceleration is obtained based on these variables and their derivatives.

According to Mellinger and Kumar [26], the desired attitude is calculated through acceleration. As mentioned in Section 2, the coordinate system  $\mathcal{C}$  is an intermediate coordinate system obtained by rotating the world coordinate system around the z-axis with the yaw angle, and  $\mathbf{x}_c$  indicates the unit vectors of the x-axis for this coordinate system.

$$\begin{aligned}
 \mathbf{z}_b &= \frac{\mathbf{a}^d + g \mathbf{z}_w}{\|\mathbf{a}^d + g \mathbf{z}_w\|} \\
 \mathbf{x}_c &= [\cos \psi, \sin \psi, 0]^T \\
 \mathbf{y}_b &= \frac{\mathbf{z}_b \times \mathbf{x}_c}{\|\mathbf{z}_b \times \mathbf{x}_c\|} \\
 \mathbf{x}_b &= \mathbf{y}_b \times \mathbf{z}_b, \quad {}^w \mathbf{R}^d = [\mathbf{x}_b \quad \mathbf{y}_b \quad \mathbf{z}_b]
 \end{aligned} \tag{35}$$

By differentiating both sides of Equation (5), the jerk planned in the position loop can be mapped into the desired angular velocity.

$$\begin{aligned} m\dot{\mathbf{a}} &= \dot{T}\mathbf{z}_b + T(\mathbf{w}_w \times \mathbf{z}_b) \\ \dot{T} &= \mathbf{z}_b \cdot m\dot{\mathbf{a}} \\ \mathbf{h} &= \mathbf{w}_w \times \mathbf{z}_b = \frac{m}{T}(\mathbf{j}^d - (\mathbf{z}_b \cdot \mathbf{j}^d)\mathbf{z}_b) \\ w_{bx} &= -\mathbf{h} \cdot \mathbf{y}_b, \quad w_{by} = \mathbf{h} \cdot \mathbf{x}_b \end{aligned} \tag{36}$$

Also in paper [26], however, the z-axis angular velocity is taken as zero for a constant yaw setpoint, which is actually nonzero. In this paper, the z-axis angular velocity is calculated through Equation (7). By taking the inverse of Equation (7), we obtain

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ \tan \phi \sin \theta & 1 & -\tan \phi \cos \theta \\ -\sec \phi \sin \theta & 0 & \sec \phi \cos \theta \end{bmatrix} \begin{bmatrix} w_{bx} \\ w_{by} \\ w_{bz} \end{bmatrix} \tag{37}$$

By solving Equation (38), the z-axis angular velocity can be determined based on the derivative value of the yaw angle.

$$\dot{\psi} \cos \phi = -w_{bx} \sin \theta + w_{bz} \cos \theta \tag{38}$$

In the attitude controller, both angular velocity and angular acceleration are required as feed-forward terms to enhance tracking accuracy in attitude control. Thus, the differential flatness property is extended to the angular acceleration to calculate its desired value. Similarly, by taking the second derivative of Equation (5), we have

$$m\mathbf{s}^d = \ddot{T}\mathbf{z}_b + 2\dot{T}(\mathbf{w}_w \times \mathbf{z}_b) + T \frac{\delta(\mathbf{w}_w \times \mathbf{z}_b)}{\delta t} \tag{39}$$

The equation for calculating the second derivative of thrust  $T$  is

$$\ddot{T} = (\mathbf{w}_w \times \mathbf{z}_b) \cdot m\dot{\mathbf{a}} + \mathbf{z}_b \cdot m\mathbf{s}^d \tag{40}$$

Expanding Equation (39), the cross product of the angular acceleration with the z-axis of the world coordinate system can be calculated. By projecting this quantity onto the x-axis and y-axis of the body coordinate system, the values of  $\dot{w}_{bx}$  and  $\dot{w}_{by}$  can be derived as

$$\begin{aligned} \frac{\delta(\mathbf{w}_w \times \mathbf{z}_b)}{\delta t} &= (\hat{\mathbf{w}}_w \frac{w}{\delta t} \mathbf{R} \mathbf{w}_b + \frac{w}{b} \mathbf{R}^{-1} \frac{\delta \mathbf{w}_b}{\delta t}) \times \mathbf{z}_b + \mathbf{w}_w \times (\mathbf{w}_w \times \mathbf{z}_b) \\ \boldsymbol{\zeta} &= \hat{\mathbf{w}}_w \times \mathbf{z}_b = \frac{w}{b} \mathbf{R}^{-1} \frac{\delta \mathbf{w}_b}{\delta t} \times \mathbf{z}_b \\ \dot{w}_{bx} &= -\boldsymbol{\zeta} \cdot \mathbf{y}_b, \quad \dot{w}_{by} = \boldsymbol{\zeta} \cdot \mathbf{x}_b \end{aligned} \tag{41}$$

Furthermore, we take the derivative of Equation (38):

$$\ddot{\psi} \cos \phi = \dot{w}_{bz} \cos \theta - \dot{w}_{bx} \sin \theta - \dot{\theta} \dot{\phi} + \dot{\phi} \dot{\psi} \sin \phi \tag{42}$$

By solving Equation (42), the angular acceleration of the z-axis in the body coordinate system can be calculated. Based on differential flatness, the information in the position controller,  $a^d, j^d, s^d$ , can be mapped to the inputs of the attitude controller,  $R^d, w^d, \dot{w}^d$ . Therefore, the second derivative of acceleration, i.e., the snap variable, must be considered in the position controller. Apart from better handling of the constraint of motor responsiveness, another crucial reason for this is to calculate the desired angular acceleration for the attitude controller.

### 3.4. Optimization-Based Control Allocation

The desired thrust can be calculated based on the desired acceleration from the position controller. When combined with the desired torque from the attitude controller, the desired control inputs can be obtained. Under normal conditions, it is sufficient to compute

the desired rotational speeds for each motor by solving the system of linear Equations (3). Nevertheless, considering potential inaccuracies in the applied constraints and the possibility of excessive torque required during extreme flight conditions, the desired motor rotational speed might still exceed the actual speed limitation. To minimize the effects of motor saturation, the final module employs an optimization-based approach for control allocation. The algorithm is expressed as

$$\begin{aligned} \min_{\omega} & \left( \begin{bmatrix} \tau \\ T \end{bmatrix} - \mathbf{D}\omega \right)^T \mathbf{Q} \left( \begin{bmatrix} \tau \\ T \end{bmatrix} - \mathbf{D}\omega \right) \\ \text{s.t.} & \omega_{\min} \leq \omega \leq \omega_{\max} \end{aligned} \quad (43)$$

where  $\mathbf{Q}$  represents the weight matrix. The desired rotational speeds for the four motors are determined by solving the aforementioned optimization problem. Subsequently, these desired speeds are translated into the corresponding PWM signals and transmitted to the motors.

#### 4. Results

The simulations were performed within the MATLAB software, and the LMPC optimization equations were solved using the OSQP solver. All modules were individually implemented on a HP OMEN8 Plus laptop, which is equipped with an Intel Core i7-12700H processor and 16 GB RAM. Under this specific hardware configuration, the average computation time for solving the MPC optimization problem is 4 ms.

A series of simulation scenarios were conducted to validate the tracking performance of the proposed controller. The relevant parameters of the quadrotor-UAV are provided in Table 1, which is referred to [28]. The performance of the proposed attitude controller in terms of attitude angle tracking is demonstrated in Section 4.1. The trajectory tracking performance of the entire controller is presented in Section 4.2.

**Table 1.** Parameters of the quadrotor-UAV in simulation.

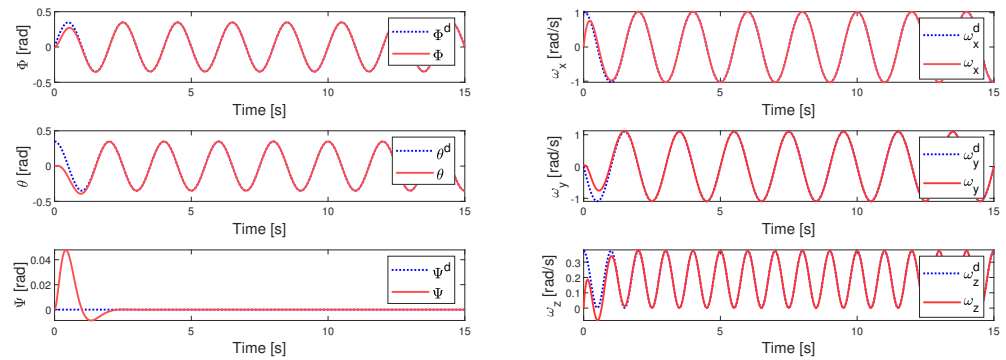
Variables	Description	Values
$m$	The mass of the quadrotor-UAV (kg)	0.98
$\mathbf{J}$	The inertia tensor of the quadrotor-UAV	$\text{Diag}\{2.64 \times 10^{-3}, 2.64 \times 10^{-3}, 4.96 \times 10^{-3}\}$
$k_f$	The force scaling factor of motor	$8.98132 \times 10^{-9}$
$k_n$	The moment scaling factor of motor	$1.1694 \times 10^{-10}$
$k_m$	Motor Response Constant	10
$l$	The arm length of the quadrotor-UAV(m)	0.26
$\omega_{\max}$	Maximum motor speed (RPM)	42.000

##### 4.1. Nonlinear Attitude Controller Based on SO(3)

It is crucial to substantiate the control performance and robustness of the attitude controller, given its direct relevance to the overall controller. In the attitude control law as shown in Equation (27), both  $K_1$  and  $K_2$  were set as 10. The desired tracking attitude is given as

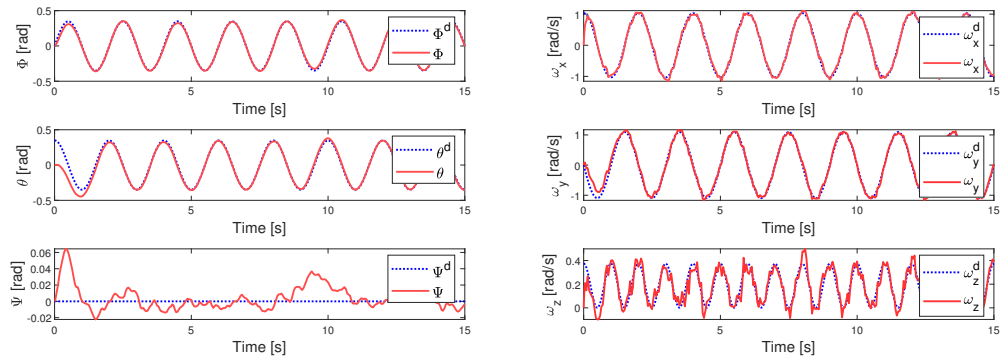
$$\phi(t) = \frac{\pi}{9} \sin(\pi t), \theta(t) = \frac{\pi}{9} \cos(\pi t), \psi(t) = 0 \quad (44)$$

Firstly, the simulation was carried out without the external disturbances. In Figure 3, the actual and desired attitude and angular rate in three axes are plotted over time. As can be observed in the graph, even with an initial deviation between the actual and desired pitch angle, the system stabilized within 2 seconds. Subsequently, the system accurately tracked the specified attitude angles and angular velocities throughout the ensuing timeframe.



**Figure 3.** Actual and desired attitude and angular rate without external disturbances.

Next, observation errors with an amplitude of 0.1 were added to the measured values of attitude and angular rate, while white noise sampled every 0.2 s with an amplitude of 20% of the controller normal operating conditions was employed to simulate the external disturbances. The actual and desired attitude and angular rate are plotted over time in Figure 4. Despite such a large initial pitch angle deviation and a massive amount of disturbances, the system stabilized and converged to the desired states quickly. It is important to note that there were several significant peaks in the actual angular velocities. This was caused by the presence of large random disturbances and observation errors at that particular moment. The attitude controller compensated for tracking errors by outputting a large torque. Actually, the angular velocity remained stable, but there were certain instances where it exhibits higher rates of variation.



**Figure 4.** Actual and desired attitude and angular rate with external disturbances.

The results shown above have proved that the proposed nonlinear attitude controller based on  $SO(3)$  space has a robust control performance even under considerable external disturbances.

#### 4.2. Trajectory Tracking

In order to validate the tracking performance of the proposed LMPC-based robust controller, four simulations were conducted: (a) aggressive trajectory tracking, (b) tracking lemniscate of Bernoulli, (c) acrobatic flight trajectory tracking, and (d) aggressive trajectory tracking under observation errors and extreme external disturbances. In these simulations, other controllers were employed for the purpose of comparison.

In the simulations, the prediction horizon for the position control was set to 50, which means that the LMPC optimizes the reference trajectory of the position loop for the next 50 time steps. Relevant parameters in the LMPC algorithm are presented in Table 2. The planning frequency of the position controller was set as 20 Hz, while the control frequency of the attitude controller was set as 200 Hz.

**Table 2.** MPC parameter settings in simulation.

Variables	Description	Values
$N$	The prediction horizon	50
$G_1$	The weight matrix of tracking error	$Diag\{20, 20, \dots, 20, 200\}$
$G_2$	The weight matrix of control input	$Diag\{0.01, 0.005, \dots, 0.005\}$
$Q$	The weight matrix for control allocation	$Diag\{5, 5, 5, 10\}$
$K_1$	The proportional coefficient in attitude control law	10
$K_2$	The proportional coefficient in attitude control law	10

#### 4.2.1. Aggressive Trajectory Tracking

In the first simulation scenario, the initial position for the quadrotor-UAV was at  $[0, 0, 0]$ . Both velocity and acceleration were configured as zero at the initial time. The given reference waypoints are

$$\begin{aligned} p_{x,k}^d &= 3 \cos\left(\frac{2\pi}{5} t_k\right) \\ p_{y,k}^d &= 2 \sin\left(\frac{2\pi}{5} t_k\right) \\ p_{z,k}^d &= 4 - 2 \cos\left(\frac{2\pi}{5} t_k\right) \end{aligned} \quad (45)$$

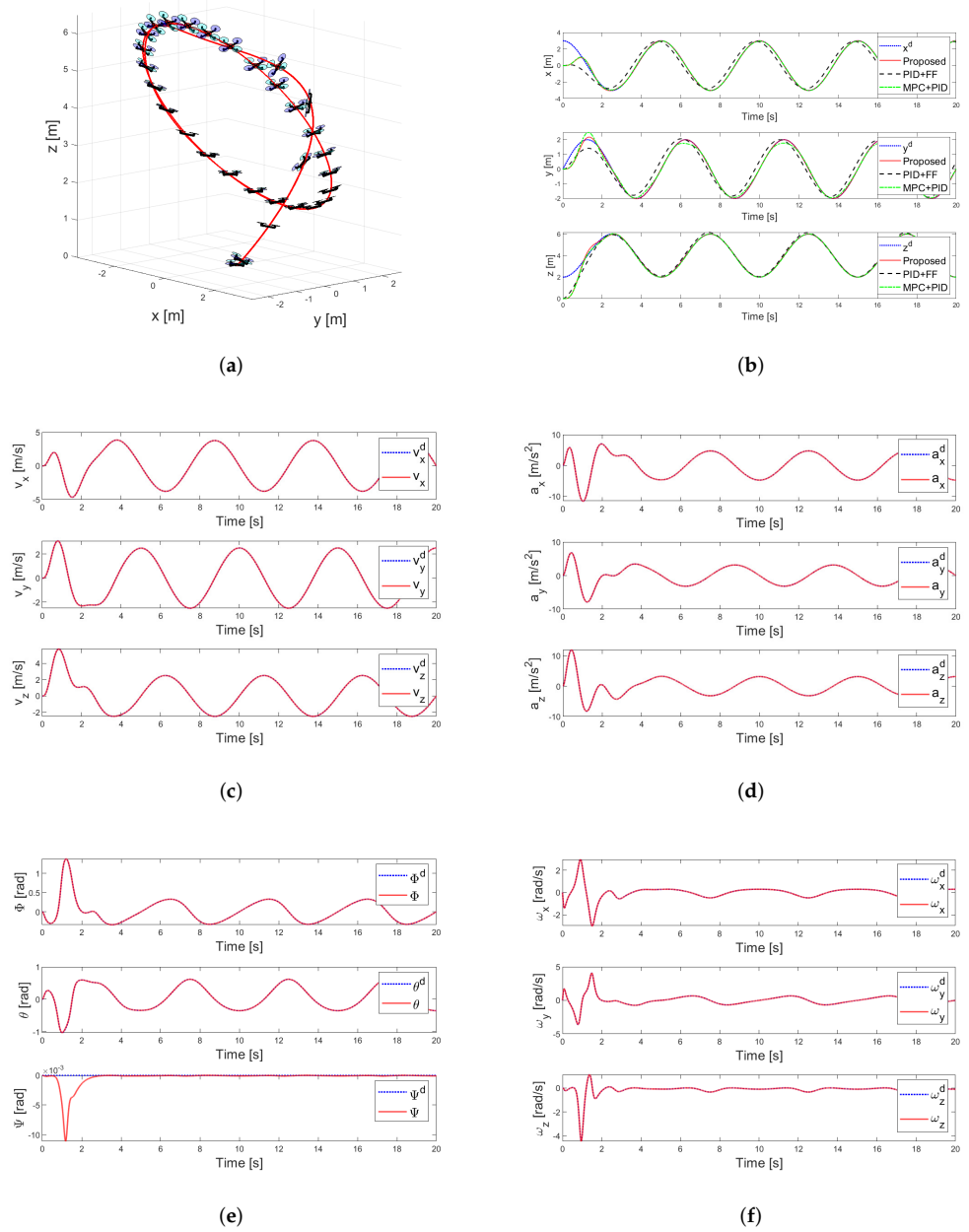
which requires aggressive attitude changes in both roll and pitch angles. A differential flatness-based cascaded PID controller with velocity and acceleration feedforward [24] was employed for comparison. Furthermore, a controller that combines MPC for position control with PID for attitude control was conducted as a comparative experiment to validate the performance of our attitude controller. The tracking performance is evaluated by comparing the desired and actual dynamics states (i.e., position, velocity, acceleration, attitude, and angular rate), as shown in Figure 5.

The mean square error (MSE) of position for the aforementioned three controllers is shown in Table 3.

**Table 3.** The MSE in tracking an aggressive trajectory.

Algorithms/Time	0–10 s	10–20 s	0–20 s
Proposed	[0.4670, 0.0373, 0.2753]	$[6.507 \times 10^{-6}, 2.917 \times 10^{-6}, 3.559 \times 10^{-6}]$	[0.2335, 0.0186, 0.1376]
PID + FF	[0.8527, 0.1652, 0.2536]	[0.2183, 0.1099, 0.0098]	[0.5355, 0.135, 0.1317]
MPC + PID	[0.4632, 0.0507, 0.2781]	[0.0008, 0.0097, 0.0007]	[0.2320, 0.0302, 0.1394]

In this scenario, substantial initial deviations occurred in both the  $x$  and  $z$ -axes. Moreover, there were also significant attitude changes in roll and pitch angles with an amplitude of 1 radius during flight. As shown in Figure 5 and Table 3, the PID controller with feedforward failed to track the trajectory, leading to observable phase lag in tracking. This shortfall could be attributed to the presence of initial errors, impeding the provision of precise feedforward information. For the controller that combines MPC and PID, although it exhibited a good tracking performance overall, there was a significant tracking error in the  $y$ -axis position compared to the proposed controller. This disparity became particularly conspicuous during peaks in the  $y$ -axis position, where substantial tracking errors manifest. On the contrary, the proposed controller tracked the given trajectory with high spatial and temporal precision.



**Figure 5.** Aggressive trajectory tracking without disturbances. (a) Aggressive trajectory tracking. (b) Desired position (blue curve), actual position by the proposed controller (red curve), actual position by the controller that combines MPC for position control with PID for attitude control (green curve), and actual position by the PID controller with feedforward (dashed black curve). (c) Actual and desired velocity. (d) Actual and desired acceleration. (e) Actual and desired attitude. (f) Actual and desired angular rate.

#### 4.2.2. Comparison of Different Controllers in Tracking Lemniscate of Bernoulli

To examine the performance of the proposed controller in waypoint tracking, a simulation was conducted utilizing the Bernoulli lemniscate as the reference trajectory. Similarly, the above two controllers (the PID controller with velocity and acceleration feedforward and the controller that combines MPC for position control with PID for attitude control) were implemented for comparison. The given reference waypoints are as follows. The

initial position of the quadrotor-UAV was set at  $[0, 4, 1]$ , and both velocity and acceleration were set to zero at the initial time.

$$\begin{aligned}
 R &= 4, w = \frac{\pi}{3} \\
 p_{x,k}^d &= -\frac{R \cos(wt_k) \sin(wt_k)}{\sin^2(wt_k) + 1} \\
 p_{y,k}^d &= \frac{R \cos(wt_k)}{\sin^2(wt_k) + 1} \\
 p_{z,k}^d &= 1
 \end{aligned} \tag{46}$$

The waypoints and the actual flight path by the three controllers are illustrated in Figure 6. The comparisons between desired and actual values of position, velocity, acceleration, attitude, and angular rate are plotted in separate subplots of Figure 6.

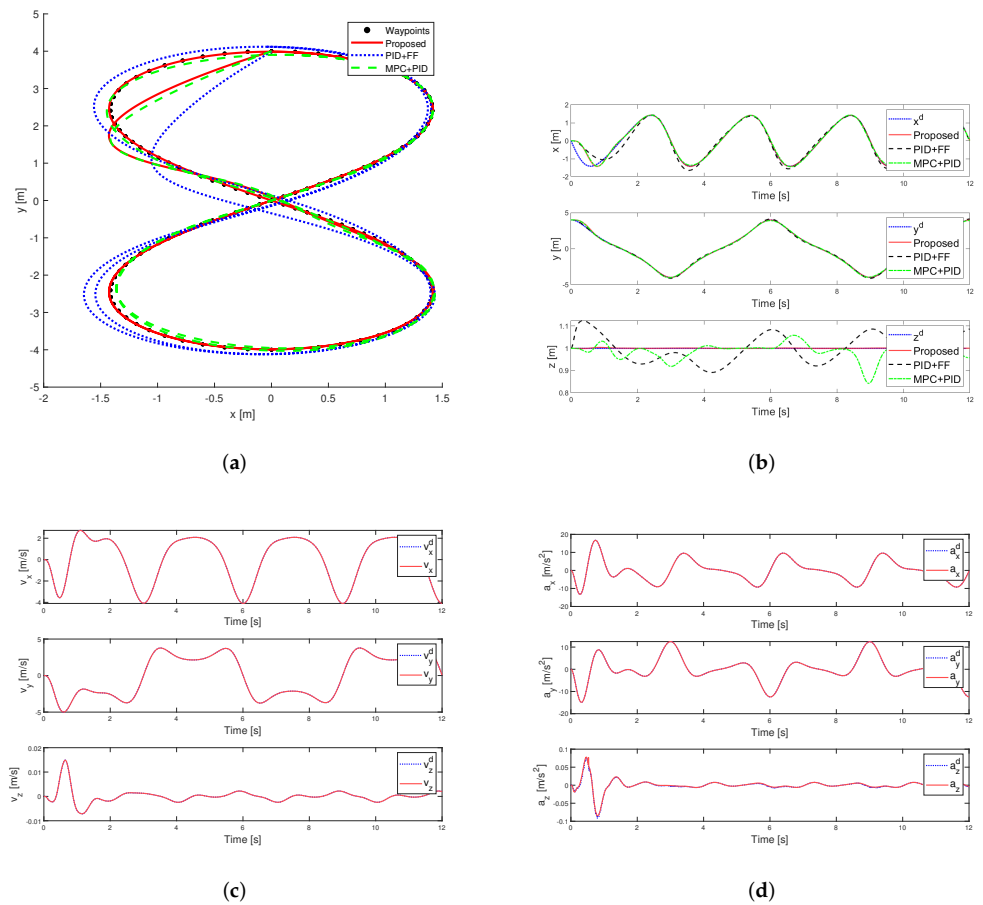
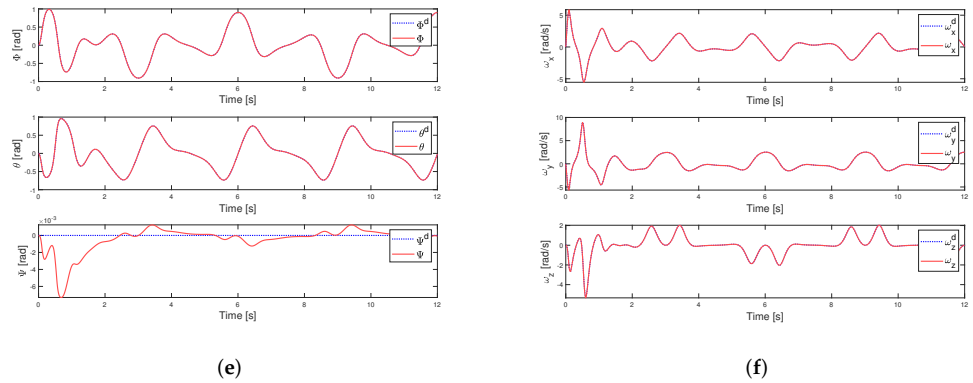


Figure 6. Cont.



**Figure 6.** Lemniscate of Bernoulli tracking. (a) Trajectory of different controllers. (b) Actual and desired position. (c) Actual and desired velocity. (d) Actual and desired acceleration. (e) Actual and desired attitude. (f) Actual and desired angular rate.

The MSE of position for the aforementioned three controllers is presented in Table 4.

**Table 4.** The MSE in tracking Lemniscate of Bernoulli.

Algorithms/Time	0–6 s	6–12 s	0–12 s
Proposed	[0.0420, 0.0074, $1.4256 \times 10^{-6}$ ]	[ $5.027 \times 10^{-5}$ , $5.128 \times 10^{-5}$ , $4.127 \times 10^{-7}$ ]	[0.0210, 0.0037, $9.191 \times 10^{-7}$ ]
PID + FF	[0.0832, 0.0049, 0.0045]	[0.0079, 0.0075, 0.0032]	[0.0455, 0.0062, 0.0038]
MPC + PID	[0.0451, 0.0070, 0.0008]	[0.0013, 0.0025, 0.0020]	[0.0232, 0.0047, 0.0014]

In this simulation, the PID feedforward controller accurately provided the velocity and acceleration feedforward information. However, its effectiveness was limited by the position loop, which only output the desired acceleration. Consequently, it failed to provide additional feedforward information (desired angular velocity and desired angular acceleration) to the attitude control loop. As a result, significant tracking errors could be observed. The MPC combined with the PID controller showcased a satisfying tracking performance. However, as presented in Figure 6 and Table 4, this controller achieved precise tracking along straight paths, but exhibited larger tracking errors during aggressive turns. This discrepancy arose due to the requirement of the quadrotor-UAV for substantial adjustments in attitude during the turns, which the PID attitude controller failed to track accurately. In contrast, the proposed controller exhibited superior tracking performance. It accurately traced the desired trajectory after the initial adjustments.

#### 4.2.3. Acrobatic Flight Trajectory Tracking

Since the design of the attitude controller is based on the  $SO(3)$  space, it possesses the capacity to track dynamic trajectories encompassing substantial attitude maneuvers. An acrobatic flying trajectory in the X-Z plane was prescribed in Figure 7 for testing the proposed controller. The controller that combines MPC for position control with PID for attitude control was added as a comparative experiment. The given reference waypoints are

$$\begin{aligned}
 w_k^d &= \frac{\sqrt{v_{\min}^2 + 2gR(1 + \cos \beta_k)}}{R} \\
 \beta_{k+1} &= \beta_k + w_k^d(t_{k+1} - t_k) \\
 p_{x,k}^d &= R \sin \beta_k \\
 p_{y,k}^d &= 0 \\
 p_{z,k}^d &= R(1 - \cos \beta_k) + 1 \\
 v_{\min} &\geq \sqrt{gR}
 \end{aligned} \tag{47}$$



where  $R$  was set to 2 m, and  $v_{min}$  was set to 6 m/s. The initial position for the quadrotor-UAV was established as  $[0, 0, 1]$ , and both velocity and acceleration were set to zero at the initial time. In order to track the given waypoints accurately, the pitch angle of the quadrotor-UAV was required to complete a full attitude maneuver from 0 to  $2\pi$ . Additionally, the large steps in the velocity and acceleration terms at the initial time made it more challenging for the controller.

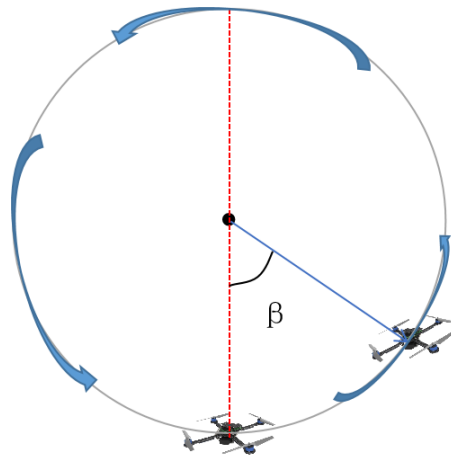


Figure 7. Acrobatic flight trajectory.

The performance in terms of tracking the acrobatic flight trajectory is evaluated by comparing the desired position, velocity, acceleration, attitude, and angular rate to the actual values and they are plotted in Figure 8.

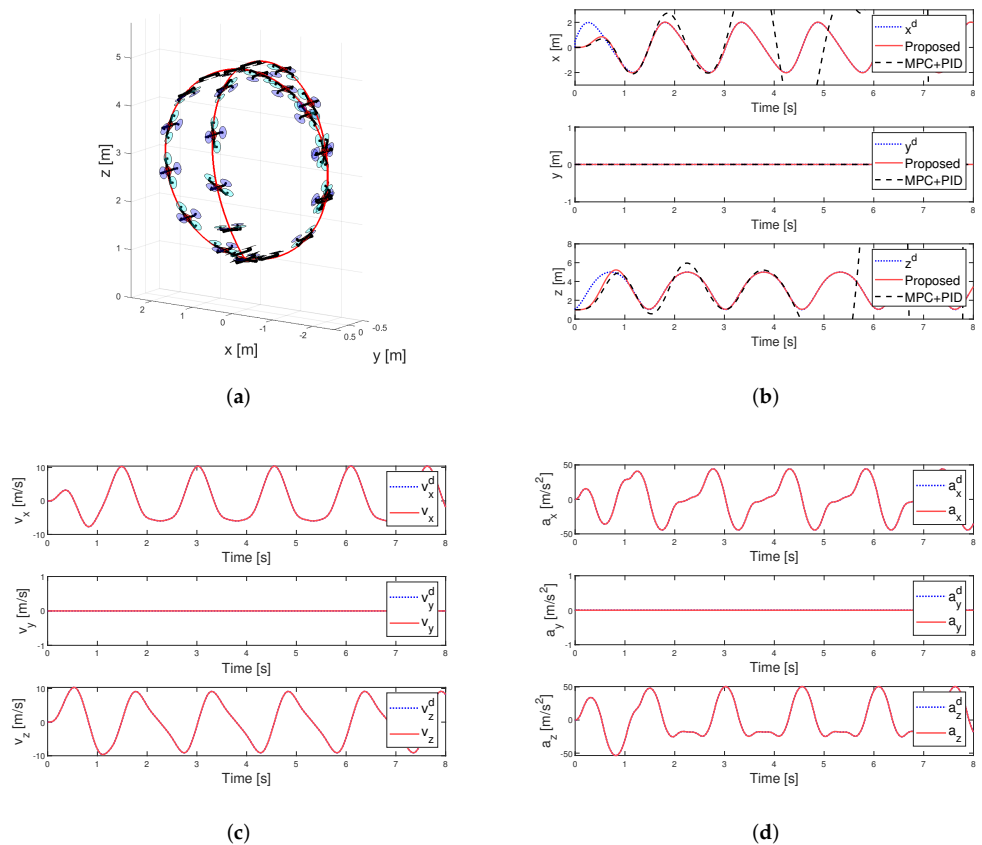
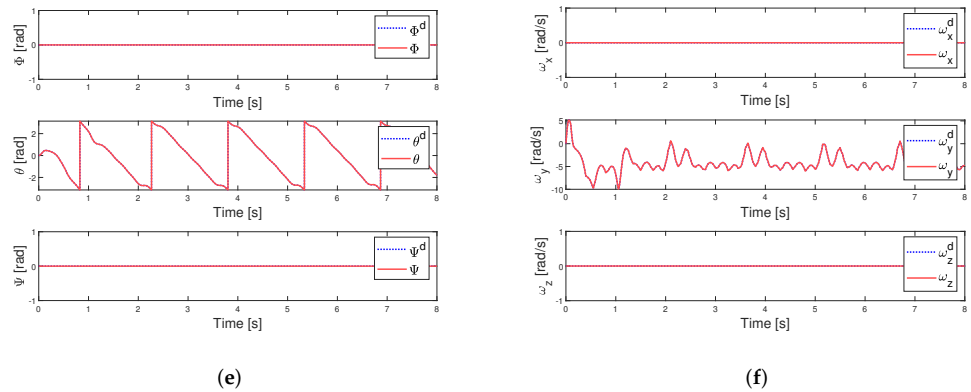


Figure 8. Cont.



**Figure 8.** Acrobatic flight trajectory tracking of the quadrotor-UAV without disturbances. (a) Acrobatic flight trajectory tracking. (b) Actual and desired position. (c) Actual and desired velocity. (d) Actual and desired acceleration. (e) Actual and desired attitude. (f) Actual and desired angular rate.

The MSE of position for tracking the acrobatic flight trajectory is shown in Table 5.

**Table 5.** The MSE in tracking acrobatic flight trajectory.

Algorithms/Time	0–4 s	4–8 s	0–8 s
Proposed	[0.2486, 0, 0.3217]	$[2.404 \times 10^{-4}, 0, 3.26 \times 10^{-4}]$	[0.1244, 0, 0.1610]

In the simulation, given the trajectory involving full attitude maneuvers, our attitude controller achieved stable tracking without encountering any singularities. Despite the presence of significant errors in both velocity and acceleration at the initial time, good accuracy in tracking the desired waypoints could be observed in Figure 8 and Table 5. As a comparative experiment, the controller that combines MPC with PID failed to track the given waypoints. This inadequacy could be attributed to the PID attitude controller’s limited ability to accurately trace the desired attitude, particularly when dealing with aggressive attitude changes.

#### 4.2.4. Aggressive Trajectory Tracking under Observation Errors and Extreme External Disturbances

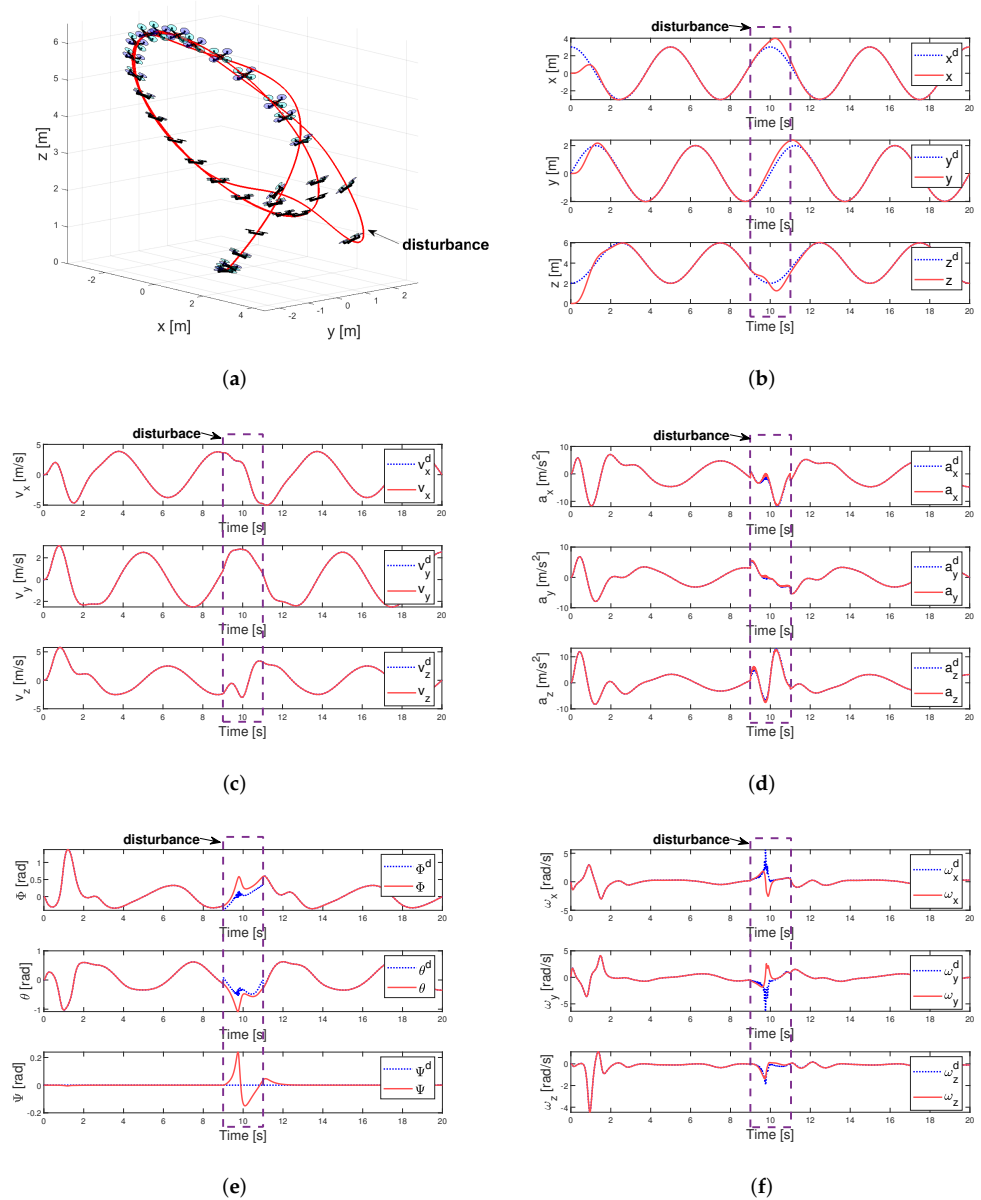
Considering the complex environmental factors the quadrotor-UAV may encounter during the actual flight, in the simulation, the artificial state observation errors were introduced for simulating the real-world conditions. Specifically, the random values were added to the state variables to emulate the measurement values. The amplitudes of the random values associated with each state variable are presented in Table 6.

**Table 6.** Observation errors in state variables.

Variables	Observation Error Amplitude
Displacement	0.05 [m]
Velocity	0.05 [m/s]
Acceleration	0.03 [m/s <sup>2</sup> ]
Attitude angle	0.02 [rad]
Angular rate	0.01 [rad/s]

The reference waypoints in this case are defined by Equation (45). The initial position for the quadrotor-UAV was set at  $[0, 0, 0]$ , with both velocity and acceleration initialized to zero at the initial time. Moreover, a significant external disturbance was introduced during the 9th to 11th seconds as shown in Figure 9. During this time interval, a constant force was

exerted on the quadrotor-UAV to simulate external disturbances, with magnitudes of 2.5N, 2N, and 1.5N along the three axes, respectively. Given the desired trajectory, observation errors, and external disturbances, the tracking performance of the controller is evaluated by comparing the desired position, velocity, acceleration, attitude, and angular velocity to the actual values in Figure 9.



**Figure 9.** Aggressive trajectory tracking under observation errors and external disturbances. (a) Trajectory of proposed controller. (b) Actual and desired position. (c) Actual and desired velocity. (d) Actual and desired acceleration. (e) Actual and desired attitude. (f) Actual and desired angular rate.

The MSE of position for tracking aggressive trajectory under observation errors and extreme external disturbances is shown in Table 7.

**Table 7.** The MSE in tracking aggressive trajectory under disturbances.

Algorithms/Time	0–8 s	8–12 s	12–20 s
Proposed	[0.5846, 0.0469, 0.3450]	[0.2929, 0.0951, 0.1592]	$[1.518 \times 10^{-4}, 1.1 \times 10^{-4}, 4.923 \times 10^{-5}]$

The results presented in Figure 9 and Table 7 demonstrated that the proposed controller exhibited strong disturbance rejection capabilities. Even in the presence of observation errors and substantial external disturbances, the controller was able to maintain robust control. It is worth mentioning that the replanning time of our controller in the position loop is less than 0.01 s, which endows our controller with online control capability.

## 5. Conclusions

In the context of waypoint tracking tasks for quadrotor UAVs, a robust controller is introduced in this paper. The LMPC algorithm is employed in the position loop and a nonlinear control law based on the  $SO(3)$  space is designed for the attitude control. The accurate tracking of the attitude loop enables the decoupling of the attitude control from the optimization within the MPC. The motor model constraints are taken into account for wide-time-domain prediction and feasible trajectory optimization. Furthermore, the nonlinear characteristics can be effectively compensated by the attitude controller, therefore allowing the controller to maintain its superior performance while reducing the computational expenses. Multiple aggressive trajectories and external disturbances were designed and simulations were conducted to test the controller. The proposed scheme showed high precision in tracking performance. Since the controller requires low computational resources, it is estimated to be implementable for real-time control for an actual quadrotor-UAV.

**Author Contributions:** Conceptualization, Z.X.; methodology, Z.X.; software, Z.X.; validation, Z.X. and W.Q.; formal analysis, Z.X., W.Q. and G.W.; investigation, Z.X.; resources, L.F.; data curation, Z.X.; writing—original draft preparation, Z.X.; writing—review and editing, G.W. and W.Q.; visualization, Z.X. and W.Q.; supervision, L.F., G.W., W.Q. and Y.H.; project administration, L.F., G.W., W.Q. and Y.H.; funding acquisition, L.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the Intelligent Aerospace System Team (2022R01003) of Zhejiang Provincial Leading Innovative Teams and the project of “Key Technology and Verification of Intelligent Piloted Electric Aircraft” (HYI21005) from Huzhou Institute of Zhejiang University.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Xu, H.; Yi, L.; Li, C.; Sun, Y.; Hou, L.; Bai, J.; Kong, F.; Han, X.; Lan, Y. Design and Experiment of Ecological Plant Protection UAV Based on Ozonated Water Spraying. *Drones* **2023**, *7*, 291. [[CrossRef](#)]
2. Liang, X.; Fang, Y.; Sun, N.; Lin, H. Nonlinear hierarchical control for unmanned quadrotor transportation systems. *IEEE Trans. Ind. Electron.* **2017**, *65*, 3395–3405. [[CrossRef](#)]
3. Xiong, T.; Liu, F.; Liu, H.; Ge, J.; Li, H.; Ding, K.; Li, Q. Multi-Drone Optimal Mission Assignment and 3D Path Planning for Disaster Rescue. *Drones* **2023**, *7*, 394. [[CrossRef](#)]
4. 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.
5. 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; Volume 3, pp. 2451–2456.
6. Greeff, M.; Schoellig, A.P. Flatness-based model predictive control for quadrotor trajectory tracking. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 6740–6745.
7. Lu, G.; Xu, W.; Zhang, F. On-manifold model predictive control for trajectory tracking on robotic systems. *IEEE Trans. Ind. Electron.* **2022**, *70*, 9192–9202. [[CrossRef](#)]
8. Romero, A.; Sun, S.; Foehn, P.; Scaramuzza, D. Model predictive contouring control for time-optimal quadrotor flight. *IEEE Trans. Robot.* **2022**, *38*, 3340–3356. [[CrossRef](#)]
9. Sun, S.; Romero, A.; Foehn, P.; Kaufmann, E.; Scaramuzza, D. A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Trans. Robot.* **2022**, *38*, 3357–3373. [[CrossRef](#)]
10. Zhao, W.; Liu, H.; Lewis, F.L. Robust formation control for cooperative underactuated quadrotors via reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4577–4587. [[CrossRef](#)]
11. Wang, Q.; Namiki, A.; Asignacion, A., Jr.; Li, Z.; Suzuki, S. Chattering Reduction of Sliding Mode Control for Quadrotor UAVs Based on Reinforcement Learning. *Drones* **2023**, *7*, 420. [[CrossRef](#)]

12. Kaufmann, E.; Loquercio, A.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Deep drone acrobatics. *arXiv* **2020**, arXiv:2006.05768.
13. Li, G.; Tunchez, A.; Loianno, G. Learning model predictive control for quadrotors. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5872–5878.
14. Han, H.; Cheng, J.; Xi, Z.; Yao, B. Cascade Flight Control of Quadrotors Based on Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11134–11141. [[CrossRef](#)]
15. Serrano, F.; Castillo, O.; Alassafi, M.; Alsaadi, F.; Ahmad, A. Terminal sliding mode attitude-position quaternion based control of quadrotor unmanned aerial vehicle. *Adv. Space Res.* **2023**, *71*, 3855–3867. [[CrossRef](#)]
16. Fresk, E.; Nikolakopoulos, G. Full quaternion based attitude control for a quadrotor. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 3864–3869.
17. Meradi, D.; Benselama, Z.A.; Hedjar, R.; Gabour, N.E.H. Quaternion-based Nonlinear MPC for Quadrotor’s Trajectory Tracking and Obstacles Avoidance. In Proceedings of the 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE), Constantine, Algeria, 29–31 October 2022; pp. 1–6.
18. Choutri, K.; Lagha, M.; Dala, L.; Lipatov, M. Quadrotors trajectory tracking using a differential flatness-quaternion based approach. In Proceedings of the 2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Sharjah, United Arab Emirates, 4–6 April 2017; pp. 1–5.
19. Lee, T. Global Exponential Attitude Tracking Controls on  $SO(3)$ . *IEEE Trans. Autom. Control* **2015**, *60*, 2837–2842. [[CrossRef](#)]
20. Yu, Y.; Yang, S.; Wang, M.; Li, C.; Li, Z. High performance full attitude control of a quadrotor on  $SO(3)$ . In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1698–1703.
21. Lee, T.; Leok, M.; McClamroch, N.H. Geometric tracking control of a quadrotor UAV on  $SE(3)$ . In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 5420–5425.
22. Lee, T. Robust Adaptive Attitude Tracking on  $SO(3)$  With an Application to a Quadrotor UAV. *IEEE Trans. Control Syst. Technol.* **2012**, *21*, 1924–1930.
23. Lian, S.; Meng, W.; Shao, K.; Zheng, J.; Zhu, S.; Li, H. Full Attitude Control of a Quadrotor Using Fast Nonsingular Terminal Sliding Mode With Angular Velocity Planning. *IEEE Trans. Ind. Electron.* **2022**, *70*, 3975–3984. [[CrossRef](#)]
24. Tian, B.; Liu, L.; Lu, H.; Zuo, Z.; Zong, Q.; Zhang, Y. Multivariable finite time attitude control for quadrotor UAV: Theory and experimentation. *IEEE Trans. Ind. Electron.* **2017**, *65*, 2567–2577. [[CrossRef](#)]
25. Islam, S.; Faraz, M.; Ashour, R.; Cai, G.; Dias, J.; Seneviratne, L. Adaptive sliding mode control design for quadrotor unmanned aerial vehicle. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 34–39.
26. 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.
27. Murray, R.M.; Li, Z.; Sastry, S.S.; Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 1994.
28. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **2020**, *6*, 478–485. [[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.