applied sciences

MDPI

Article

Design and Simulation Verification of Model Predictive Attitude Control Based on Feedback Linearization for Quadrotor UAV

Xingyu Yuan [1], Jinfa Xu [1,*] and Shengwei Li [1,2]

[1] National Key Laboratory of Rotorcraft Aeromechanics, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China; yuanxy@nuaa.edu.cn (X.Y.); lsw_2001@163.com (S.L.)
[2] Aircraft Design and Research Institute, Harbin Aircraft Industry Group Co., Ltd., Harbin 150066, China
* Correspondence: xjfae@nuaa.edu.cn

Abstract: The flight dynamics of quadrotor UAVs are characterized by significant nonlinearity and inter-axis coupling, posing challenges for the direct application of linear control theory in flight control system design. This paper proposes a feedback linearization transformation method to reduce the computational burden of MPC while addressing the nonlinear coupled flight dynamics in attitude control. The quadrotor UAV's attitude motion is transformed via feedback linearization into a linear decoupled model, consisting of three independent second-order systems. MPC theory is then employed to design the attitude control system. By solving the constrained optimal control problem within the MPC framework, control inputs for each second-order system are derived, enabling precise attitude tracking through receding horizon control theory. Simulation verifications were subsequently conducted, with the experiment comparing FLMPC, LMPC and AMPC. The results indicate that FLMPC outperforms both the other controllers in terms of control precision, computational efficiency and robustness to disturbances, suggesting its effectiveness for real-time UAV operations where both performance and computational resource constraints are critical.

Keywords: quadrotor; attitude control; feedback linearization; model predictive control

check for updates

Academic Editors: Rosario Pecora, Yutaka Ishibashi, Piotr Lichota and Mariusz Jacewicz

Received: 18 November 2024
Revised: 16 April 2025
Accepted: 6 May 2025
Published: 7 May 2025

Citation: Yuan, X.; Xu, J.; Li, S. Design and Simulation Verification of Model Predictive Attitude Control Based on Feedback Linearization for Quadrotor UAV. *Appl. Sci.* **2025**, *15*, 5218. https://doi.org/10.3390/app15095218

Copyright: © 2025 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

A quadrotor Unmanned Aerial Vehicle (UAV) is an aircraft with four rotors capable of vertical takeoff, landing and hovering. With its simple mechanical structure and excellent maneuverability, it is increasingly used in applications like military reconnaissance, cargo transport, emergency rescue and terrain mapping [1–3]. To ensure a safe flight and the completion of complex tasks, attitude control is crucial for the quadrotor UAV flight control system. A quadrotor UAV is a multi-input, multi-output (MIMO) nonlinear system with strong inter-axis coupling, making attitude control challenging, as adjustments on one axis often affect others. This challenge is further compounded by the system's under-actuated nature, with only four inputs available to control six outputs. Additionally, its lightweight design makes it highly sensitive to disturbances such as wind, requiring robust control strategies to maintain stability [4,5]. Consequently, numerous engineers and researchers have conducted in-depth studies on these control challenges [6,7].

Proportional-Integral-Derivative (PID) control is a classic control design method known for its simple structure and ease of implementation [8–10]. Active Disturbance Rejection Control (ADRC) offers robust control by estimating and compensating for both external and internal disturbances in real time, while its nonlinear state error feedback ensures high-quality control performance [11]. The Sliding Mode Control (SMC) method is

a control technique capable of handling nonlinear problems and exhibits strong robustness against system parameter variations and external disturbances [12,13]. However, each method has its limitations and achieving optimal control performance remains challenging.

Model Predictive Control (MPC) is a receding horizon control method known for its strong disturbance rejection, and robustness. Reference [14] demonstrated that Linear MPC (LMPC) outperforms both Proportional-Derivative (PD) and Linear Quadratic Regulator (LQR) in terms of control effectiveness and disturbance rejection for quadrotor UAV trajectory tracking. Despite this, LMPC struggles to effectively address the inherent nonlinear characteristics of the quadrotor UAV. In the most MPC-based methods, the linearized dynamics of the system is employed to avoid the complexity that results from nonlinear optimization problems [15,16]. The scholars in [17] designed an adaptive MPC method using Laguerre polynomials for quadrotor attitude stabilization and validated its feasibility. And in [18], Nonlinear MPC (NLMPC) provides better performance in trajectory tracking and system stability compared to LMPC, but this comes with a high computational cost. Although parameterized NMPC can reduce computational complexity, they may compromise control system performance to some extent [19]. In real-world flight scenarios, most MPC based methods rely on linearized system dynamics to simplify the control problem and avoid the computational complexity associated with solving nonlinear optimization problems. However, the inherently nonlinear dynamics of quadrotor UAVs often demand nonlinear control methods to achieve precise and stable flight performance. Although these nonlinear approaches can offer improved control accuracy and robustness, they are typically more challenging to design and implement, as they require advanced modeling and considerable computational resources. This trade-off highlights the ongoing need for efficient, real-time nonlinear control strategies in practical quadrotor applications.

Feedback Linearization (FL) can address the problem by transforming the system with coordinate adjustments and nonlinear state feedback, effectively bypassing intrinsic nonlinearities to enable the use of linear control methods [20,21]. Unlike conventional linearization, which only approximates system dynamics around a specific operating point and neglects higher-order dynamics, FL provides a globally valid linearization [22,23]. Reference [24] presents an adaptive tracking controller based on output feedback linearization to compensate for dynamical changes in a quadrotor's center of gravity, successfully managing the uncertain gravity parameter and stabilizing the quadrotor in real time, as demonstrated through simulation results. Rahmat M F proposes an adaptive feedback linearization (AFBL) control approach, combined with a PID controller, to stabilize a quadrotor's attitude and altitude in the presence of wind disturbances and parameter uncertainties [25]. While simulations demonstrate significant performance improvements over conventional methods, the approach neglects the nonlinear relationship between Euler angle rates and angular velocities in the body frame, making it unsuitable for scenarios involving rapid UAV maneuvers. In another study, Cai Z designs a feedback linearization model predictive control algorithm with a disturbance observer for quadcopter trajectory tracking, demonstrating effective tracking performance under disturbances through numerical simulations [26]. However, it assumes slight roll and pitch angle rotations and does not address the constraining effect on state variables in the model predictive control design.

This paper presents a feedback linearization model based predictive control method (FLMPC) approach for attitude control of quadrotor UAV, targeting precise and rapid tracking performance under system constraints. The core idea is to incorporate the nonlinear mapping between Euler angular velocities and body-frame angular velocities into the feedback linearization process, allowing the inherently nonlinear and coupled dynamics of the quadrotor to be transformed into a fully decoupled linear system. This transformation significantly reduces the computational complexity while enabling the determination of

optimal control inputs within predefined constraints, a process in which the constraint on the original system state variables has been modified. The rest of the paper is organized as follows: Section 2 establishes the flight dynamics model of the quadrotor UAV and performs feedback linearization transformation to create a controlled object model, serving as the foundation for controller design and simulation. Section 3 details the design of the quadrotor UAV attitude controller based on FLMPC, including key design principles and constraints. Section 4 conducts simulation experiments, where the proposed method's performance is evaluated through various test scenarios, demonstrating its effectiveness in achieving stable and responsive control. Section 5 provides a comprehensive summary of the findings, discusses the implications of the work, and suggests directions for future research to further refine and expand upon this control strategy. The Appendix A presents the process of designing an attitude controller using MPC.

2. Dynamics Model of Quadrotor UAV

2.1. Modeling of Attitude Motion Characteristics

The quadrotor's flight is sustained by the forces generated by each of its four rotors, with each rotor producing thrust that contributes to the overall lift, stability, and control of the vehicle [27,28]. An inertial frame $o\ x_e y_e z_e$ and a body-fixed frame $o\ x_b y_b z_b$, with the origin at the center of mass of the quadrotor as shown in Figure 1, are considered.

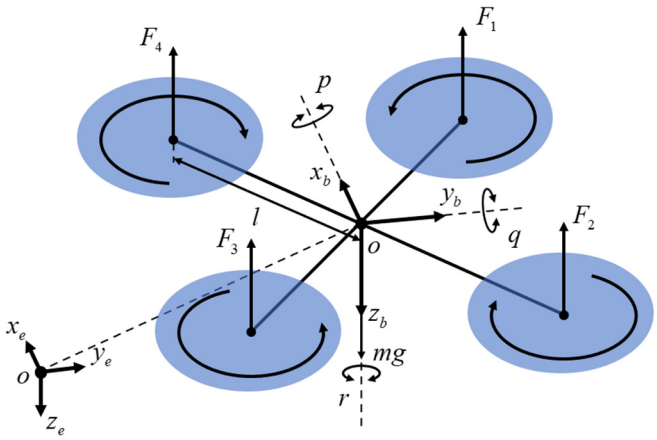


Figure 1. Configuration, inertial and body fixed frame of the quadrotor.

Based on the principles of rigid body dynamics, the rotational dynamics of a quadrotor can be expressed using the Newton–Euler method in the body fixed frame with the inertia matrix $\boldsymbol{I} = \mathrm{diag}(I_x, I_y, I_z)$, assuming the neglect of gyroscopic torque and air resistance [29,30]:

$$\boldsymbol{I} \cdot \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (\boldsymbol{I} \cdot \boldsymbol{\omega}) + \boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{\omega} = (p, q, r)^{\mathrm{T}}$ is the angular velocity vector and $\dot{\boldsymbol{\omega}}$ is its derivative with respect to time; $\boldsymbol{\omega} \times (\boldsymbol{I} \cdot \boldsymbol{\omega})$ represents the gyroscopic effect due to the rotational motion of the quadrotor; and $\boldsymbol{\tau} = (\tau_x, \tau_y, \tau_z)^{\mathrm{T}}$ is the external torque vector acting on the quadrotor.

To facilitate control-oriented modeling, the external torque vector $\boldsymbol{\tau}$ is represented by the pitch, roll, and yaw control inputs U_1, U_2 and U_3. Accordingly, the rotational dynamics of the quadrotor can be derived from (1):

$$\begin{cases} \dot{p} = \frac{I_y - I_z}{I_x} qr + \frac{1}{I_x} U_1 \\ \dot{q} = \frac{I_z - I_x}{I_y} rp + \frac{1}{I_y} U_2 \\ \dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{1}{I_z} U_3 \end{cases}, \tag{2}$$

The attitude of the quadrotor is given by the three Euler angles: yaw angle ϕ, pitch angle θ and roll angle ψ that together form the vector $\mathbf{\Omega} = (\phi, \theta, \psi)^{\mathrm{T}}$. The conversion relationship between Euler angle velocities $\dot{\mathbf{\Omega}}$ and angular velocities in the body frame $\boldsymbol{\omega}$ is as follows:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} + \boldsymbol{R}_x\boldsymbol{R}_y \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \boldsymbol{R}_x \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}, \quad (3)$$

where

$$\boldsymbol{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}, \boldsymbol{R}_y = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}, \boldsymbol{R}_z = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The above equation can be written as:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad (4)$$

In summary, from (2) and (4), the nonlinear dynamics model of the quadrotor attitude can be obtained as:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} p + q\tan\theta\sin\phi + r\tan\theta\cos\phi \\ q\cos\phi - r\sin\phi \\ q\sin\phi\sec\theta + r\cos\phi\sec\theta \\ \frac{I_y - I_z}{I_x}qr \\ \frac{I_z - I_x}{I_y}rp \\ \frac{I_x - I_y}{I_z}pq \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix}, \quad (5)$$

It can be seen from the equation that the attitude motion of the quadrotor UAV is nonlinear and exhibit inter-axis coupling. For attitude control design, this nonlinear coupling cannot be neglected.

2.2. Model Linear Transformation

The feedback linearization of the quadrotor UAV can be achieved through an intuitive transformation of the state variables and an input redefinition.

Since the goal of feedback linearization is to decouple (5) into three second-order linear systems, define the target system as:

$$\begin{cases} \ddot{\phi} = -\phi + k_1\dot{\phi} \\ \ddot{\theta} = -\theta + k_2\dot{\theta} \\ \ddot{\psi} = -\psi + k_3\dot{\psi} \end{cases}, \quad (6)$$

where k_1, k_2 and k_3 are constant parameters whose values are yet to be determined, and they play a crucial role in defining the system's stability and dynamic performance.

The second derivatives $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$ can be expressed in (7) based on the model (5) by setting $u_1 = U_1/I_x$, $u_2 = U_2/I_y$ and $u_3 = U_3/I_z$:

$$\begin{cases} \ddot{\phi} = \left(\dot{\phi}\tan\theta + \dot{\psi}\sec\theta\right)\dot{\theta} + (\sigma_1 + u_1) + (\sigma_2 + u_2)\tan\theta\sin\phi + (\sigma_3 + u_3)\tan\theta\cos\phi \\ \ddot{\theta} = -\dot{\phi}\dot{\psi}\cos\theta + (\sigma_2 + u_2)\cos\phi - (\sigma_3 + u_3)\sin\phi \\ \ddot{\psi} = \left(\dot{\phi}\sec\theta + \dot{\psi}\tan\theta\right)\dot{\theta} + (\sigma_2 + u_2)\sin\phi\sec\theta + (\sigma_3 + u_3)\cos\phi\sec\theta \end{cases}, \tag{7}$$

where $\sigma_1 = \left(I_y - I_z\right)/I_x \cdot q \cdot r$, $\sigma_2 = \left(I_z - I_x\right)/I_y \cdot r \cdot p$, $\sigma_3 = \left(I_x - I_y\right)/I_z \cdot p \cdot q$.

Defining the state variable vector $\boldsymbol{\varkappa}^{\mathrm{T}} = (z_1, z_2, z_3, z_4, z_5, z_6) = \left(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right)$, the above equations (6) and (7) can be reformulated and expressed in a state–space representation as $\dot{\boldsymbol{\varkappa}} = f(\boldsymbol{\varkappa}, \boldsymbol{u})$, where:

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{pmatrix} = \begin{pmatrix} z_2 \\ -z_1 + k_1 z_2 \\ z_4 \\ -z_3 + k_2 z_4 \\ z_6 \\ -z_5 + k_3 z_6 \end{pmatrix} + \begin{pmatrix} 0 \\ z_1 - k_1 z_2 + f_1(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + u_1 + u_2\tan z_3 \sin z_1 + u_3 \tan z_3 \cos z_1 \\ 0 \\ z_3 - k_2 z_4 + f_2(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + u_2 \cos z_1 - u_3 \sin z_1 \\ 0 \\ z_5 - k_3 z_6 + f_3(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + u_2 \sin z_1 \sec z_3 + u_3 \cos z_1 \sec z_3 \end{pmatrix}. \tag{8}$$

Here

$$f_1(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) = (z_2\tan z_3 + z_6 \sec z_3)z_4 + \sigma_1 + \sigma_2 \tan z_3 \sin z_1 + \sigma_3 \tan z_3 \cos z_1,$$

$$f_2(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) = -z_2 z_6 \cos z_3 + \sigma_2 \cos z_1 - \sigma_3 \sin z_1,$$

$$f_3(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) = (z_2 \sec z_3 + z_6 \tan z_3)z_4 + \sigma_2 \sin z_1 \sec z_3 + \sigma_3 \cos z_1 \sec z_3.$$

Define a virtual control vector:

$$\boldsymbol{v} = \boldsymbol{N}(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\varkappa})\boldsymbol{u} \tag{9}$$

where

$$\boldsymbol{N}(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) = \begin{pmatrix} f_1(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + z_1 - k_1 z_2 \\ f_2(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + z_3 - k_2 z_4 \\ f_3(\boldsymbol{\varkappa}, \boldsymbol{\sigma}) + z_5 - k_3 z_6 \end{pmatrix} \tag{10}$$

$$\boldsymbol{M}(\boldsymbol{\varkappa}) = \begin{pmatrix} 1 & \tan z_3 \sin z_1 & \tan z_3 \cos z_1 \\ 0 & \cos z_1 & -\sin z_1 \\ 0 & \sin z_1 \sec z_3 & \cos z_1 \sec z_3 \end{pmatrix} \tag{11}$$

and substitute it for the second term on the right side of (8). The system can be reformulated into a linearized model with decoupled attitude dynamics, expressed as:

$$\dot{\boldsymbol{\varkappa}} = \boldsymbol{A}\boldsymbol{\varkappa} + \boldsymbol{B}\boldsymbol{v} \tag{12}$$

where

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 & & \\ & \boldsymbol{A}_2 & \\ & & \boldsymbol{A}_3 \end{pmatrix}, \boldsymbol{A}_i = \begin{pmatrix} 0 & 1 \\ -1 & k_i \end{pmatrix}, i = 1, 2, 3,$$

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1 & & \\ & \boldsymbol{B}_2 & \\ & & \boldsymbol{B}_3 \end{pmatrix}, \boldsymbol{B}_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, i = 1, 2, 3.$$

Here A_1, A_2 and A_3 represent the characteristic matrices of second-order linear systems for roll, pitch and yaw motions, respectively. The eigenvalues of these matrices and the damping ratios are:

$$\lambda_{i_{1,2}} = \frac{k_i \pm \sqrt{k_i^2 - 4}}{2} \qquad (i = 1, 2, 3), \tag{13}$$

$$\xi_i = -\frac{k_i}{2} \qquad (i = 1, 2, 3). \tag{14}$$

To stabilize these second-order systems and achieve a damping ratio of 0.707, set $k_1 = k_2 = k_3 = -1.414$.

The inputs of the controlled system of the quadrotor UAV can be calculated as:

$$\boldsymbol{u} = \boldsymbol{M}(\varkappa)^{-1}(\boldsymbol{v} - \boldsymbol{N}(\varkappa, \sigma)), \tag{15}$$

where $\boldsymbol{N}(\varkappa, \sigma)$ and $\boldsymbol{M}(\varkappa)$ are transformation matrices for decoupling and linearizing the controlled system model. The matrix $\boldsymbol{M}(\varkappa)$ is non-singular and the equation has a solution if and only if $|\boldsymbol{M}(\varkappa)| \neq 0$. Therefore, the pitch angle range of the quadrotor UAV is limited in $(-90°, 90°)$.

3. Attitude Controller Design

This section presents the design of a model predictive controller (MPC) for the attitude control of the quadrotor UAV, based on the feedback-linearized model from the previous section. The structure of the attitude control system is illustrated in Figure 2, while the detailed design process of the attitude controller is provided in the Appendix A.

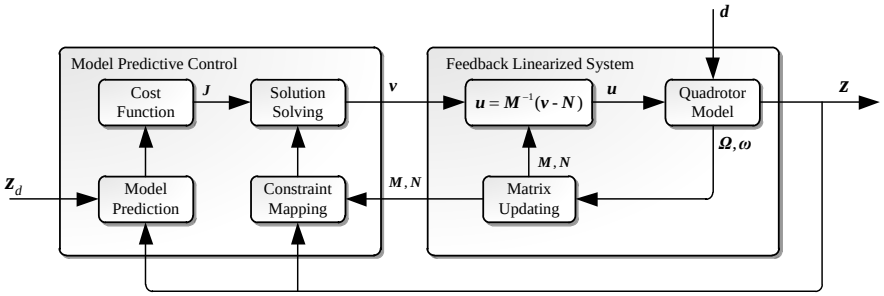


Figure 2. The attitude control system structure.

3.1. Prediction Equation for the Feedback Linearized Model

The discrete-time state–space equation of the feedback linearized model (12) can be obtained using the zero-order hold method:

$$\varkappa_{[k+1]} = \boldsymbol{A}_{\mathrm{d}}\varkappa_{[k]} + \boldsymbol{B}_{\mathrm{d}}\boldsymbol{v}_{[k]}, \tag{16}$$

where $\boldsymbol{A}_{\mathrm{d}} = \mathrm{e}^{\boldsymbol{A}T_s}$, $\boldsymbol{B}_{\mathrm{d}} = \boldsymbol{A}^{-1}(\mathrm{e}^{\boldsymbol{A}T_s} - \boldsymbol{I})\boldsymbol{B}$.

According to the model predictive control theory, (16) can be rewritten into an incremental form:

$$\varkappa_{[k+1]} = \boldsymbol{A}_{\mathrm{d}}\delta\varkappa_{[k]} + \boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k]} + \varkappa_{[k]}, \tag{17}$$

where $\delta\boldsymbol{v}_{[k]} = \boldsymbol{v}_{[k]} - \boldsymbol{v}_{[k-1]}$, $\delta\varkappa_{[k]} = \varkappa_{[k]} - \varkappa_{[k-1]}$.

Based on the state vector $\varkappa_{[k|k]}$ at step k, the state vector at step $k+1$ can be predicted as:

$$\varkappa_{[k+1|k]} = \boldsymbol{A}_{\mathrm{d}}\delta\varkappa_{[k|k]} + \boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k|k]} + \varkappa_{[k|k]}, \tag{18}$$

the state vector at step $k + 2$ can be predicted as:

$$\boldsymbol{\varkappa}_{[k+2|k]} = (\boldsymbol{A}_{\mathrm{d}}{}^2 + \boldsymbol{A}_{\mathrm{d}})\delta\boldsymbol{\varkappa}_{[k|k]} + (\boldsymbol{A}_{\mathrm{d}}\boldsymbol{B}_{\mathrm{d}} + \boldsymbol{B}_{\mathrm{d}})\delta\boldsymbol{v}_{[k|k]} + \boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k+1|k]} + \boldsymbol{\varkappa}_{[k|k]}. \tag{19}$$

where $\delta\boldsymbol{v}_{[k+i|k]}$ is the control input increment at step $k + i$ predicted at step k; $\boldsymbol{\varkappa}_{[k+i|k]}$ is the system state vector predicted at step $k + i$ from step k.

Similarly, the state vectors at step $k + N_{\mathrm{c}}$ and at step $k + N_{\mathrm{p}}$ are given by:

$$\boldsymbol{\varkappa}_{[k+N_{\mathrm{c}}|k]} = \sum_{i=1}^{N_{\mathrm{c}}} \boldsymbol{A}_{\mathrm{d}}{}^i\delta\boldsymbol{\varkappa}_{[k|k]} + \sum_{i=0}^{N_{\mathrm{c}}-1} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k|k]} + \cdots + \boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k+N_{\mathrm{c}}-1|k]} + \boldsymbol{\varkappa}_{[k|k]}. \tag{20}$$

$$\boldsymbol{\varkappa}_{[k+N_{\mathrm{p}}|k]} = \sum_{i=1}^{N_{\mathrm{p}}} \boldsymbol{A}_{\mathrm{d}}{}^i\delta\boldsymbol{\varkappa}_{[k|k]} + \sum_{i=0}^{N_{\mathrm{p}}-1} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k|k]} + \cdots + \sum_{i=0}^{N_{\mathrm{p}}-N_{\mathrm{c}}} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}}\delta\boldsymbol{v}_{[k+N_{\mathrm{c}}-1|k]} + \boldsymbol{\varkappa}_{[k|k]}. \tag{21}$$

where N_{c} is the control horizon and N_{p} is the prediction horizon. To reduce the computational burden, the control inputs will be held constant at the value from the previous time step once the time exceeds the control horizon.

By combining equations (18) through (21) and defining the control input increment vector $\delta\boldsymbol{V}_{[k]}^{\mathrm{T}} = \begin{pmatrix}\delta\boldsymbol{v}_{[k|k]} & \delta\boldsymbol{v}_{[k+1|k]} & \cdots & \delta\boldsymbol{v}_{[k+N_{\mathrm{c}}|k]}\end{pmatrix}$ and state vector $\boldsymbol{Z}_{[k]}^{\mathrm{T}} = \begin{pmatrix}\boldsymbol{\varkappa}_{[k+1|k]} & \boldsymbol{\varkappa}_{[k+2|k]} & \cdots & \boldsymbol{\varkappa}_{[k+N_{\mathrm{p}}|k]}\end{pmatrix}$, the predicted states from step $k + 1$ to $k + N_{\mathrm{p}}$ can be written as:

$$\boldsymbol{Z}_{[k]} = \boldsymbol{P}\delta\boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{I}\boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{H}\delta\boldsymbol{V}_{[k]}, \tag{22}$$

where

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{A}_{\mathrm{d}} \\ \sum_{i=1}^{2} \boldsymbol{A}_{\mathrm{d}}{}^i \\ \vdots \\ \sum_{i=1}^{N_{\mathrm{p}}} \boldsymbol{A}_{\mathrm{d}}{}^i \end{pmatrix}_{N_{\mathrm{p}}\times 1}, \boldsymbol{I} = \begin{pmatrix} \boldsymbol{I}_{n\times n} \\ \boldsymbol{I}_{n\times n} \\ \vdots \\ \boldsymbol{I}_{n\times n} \end{pmatrix}_{N_{\mathrm{p}}\times 1}, \boldsymbol{H} = \begin{pmatrix} \boldsymbol{B}_{\mathrm{d}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \sum_{i=0}^{1} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} & \boldsymbol{B}_{\mathrm{d}} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{N_{\mathrm{c}}-1} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} & \sum_{i=0}^{N_{\mathrm{c}}-2} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} & \cdots & \boldsymbol{B}_{\mathrm{d}} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{N_{\mathrm{p}}-1} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} & \sum_{i=0}^{N_{\mathrm{p}}-2} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} & \cdots & \sum_{i=0}^{N_{\mathrm{p}}-N_{\mathrm{c}}} \boldsymbol{A}_{\mathrm{d}}{}^i\boldsymbol{B}_{\mathrm{d}} \end{pmatrix}_{N_{\mathrm{p}}\times N_{\mathrm{c}}}.$$

3.2. Model Predictive Control Performance

For a given control objective $\boldsymbol{\varkappa}_{d[k+i]}$, the performance index for the control within the prediction horizon N_{p} at step k is defined as:

$$J = \left\|\boldsymbol{\varkappa}_{[k+N_{\mathrm{p}}|k]} - \boldsymbol{\varkappa}_{d[k+N_{\mathrm{p}}]}\right\|_{\boldsymbol{S}}^2 + \sum_{i=1}^{N_{\mathrm{p}}-1} \left\|\boldsymbol{\varkappa}_{[k+i|k]} - \boldsymbol{\varkappa}_{d[k+i]}\right\|_{\boldsymbol{Q}}^2 + \sum_{i=0}^{N_{\mathrm{c}}-1} \left\|\delta\boldsymbol{v}_{[k+i|k]}\right\|_{\boldsymbol{R}}^2, \tag{23}$$

where $\boldsymbol{S}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weighting matrices for the terminal state, process state and input variables, respectively, and all are positive definite matrices. Defining the target vector $\boldsymbol{Z}_{d[k]} = \begin{pmatrix}\boldsymbol{\varkappa}_{d[k+1]} & \boldsymbol{\varkappa}_{d[k+2]} & \cdots & \boldsymbol{\varkappa}_{d[k+N_{\mathrm{p}}]}\end{pmatrix}^{\mathrm{T}}$ and the equation above can then be rewritten in matrix vector form as shown in (24):

$$J = \left\|\boldsymbol{Z}_{[k]} - \boldsymbol{Z}_{d[k]}\right\|_{\overline{\boldsymbol{Q}}}^2 + \left\|\delta\boldsymbol{V}_{[k]}\right\|_{\overline{\boldsymbol{R}}}^2, \tag{24}$$

where $\overline{\boldsymbol{Q}} = \mathrm{diag}\begin{pmatrix}\boldsymbol{Q} & \cdots & \boldsymbol{Q} & \boldsymbol{S}\end{pmatrix}$ and $\overline{\boldsymbol{R}} = \mathrm{diag}\begin{pmatrix}\boldsymbol{R} & \cdots & \boldsymbol{R}\end{pmatrix}$.

By substituting (22) into (24) and defining:

$$\boldsymbol{E}_{[k]} = \boldsymbol{Z}_{d[k]} - \boldsymbol{P}\delta\boldsymbol{\varkappa}_{[k|k]} - \boldsymbol{I}\boldsymbol{\varkappa}_{[k|k]}, \tag{25}$$

a standard quadratic performance index in terms of $\delta\boldsymbol{V}_{[k]}$ is obtained:

$$J = \frac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H} + \overline{\boldsymbol{R}})\delta\boldsymbol{V}_{[k]} - (\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]})^{\mathrm{T}}\delta\boldsymbol{V}_{[k]}. \tag{26}$$

3.3. Constrained Virtual Control Solution

For the predicted control inputs $\boldsymbol{u}_{[k+i]}$ where $i = 1, 2, \ldots, N_{\mathrm{p}}$, there are upper and lower boundary constraints:

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u}_{[k+i]} \leq \boldsymbol{u}_{\max}. \tag{27}$$

Utilizing the feedback linearization control law from (15), it follows that:

$$\boldsymbol{u}_{\min} \leq \boldsymbol{M}_{[k+i-1]}^{-1}(\boldsymbol{v}_{[k+i]} - \boldsymbol{N}_{[k+i-1]}) \leq \boldsymbol{u}_{\max}. \tag{28}$$

Therefore, the inequality constraints for the predicted virtual control input $\boldsymbol{v}_{[k+i]}$ at step k are:

$$\begin{pmatrix} \boldsymbol{I}_{3\times3} \\ -\boldsymbol{I}_{3\times3} \end{pmatrix}\boldsymbol{v}_{[k+i]} \leq \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\max} + \boldsymbol{N}_{[k+i-1]} \\ -(\boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\min} + \boldsymbol{N}_{[k+i-1]}) \end{pmatrix}. \tag{29}$$

Given that $\boldsymbol{v}_{[k+i]} = \boldsymbol{v}_{[k+i-1]} + \delta\boldsymbol{v}_{[k+i]}$, it can be concluded that:

$$\begin{pmatrix} \boldsymbol{I}_{3\times3} \\ -\boldsymbol{I}_{3\times3} \end{pmatrix}\delta\boldsymbol{v}_{[k+i]} \leq \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\max} + \boldsymbol{N}_{[k+i-1]} - \boldsymbol{v}_{[k+i-1]} \\ -(\boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\min} + \boldsymbol{N}_{[k+i-1]} - \boldsymbol{v}_{[k+i-1]}) \end{pmatrix}. \tag{30}$$

Similarly, the upper and lower constraint vectors for the predicted angular rate $\boldsymbol{\omega}_{[k+i]}$ at step k are denoted as $\boldsymbol{\omega}_{\max}$ and $\boldsymbol{\omega}_{\min}$. And from (4), the inequality constraints for $\dot{\boldsymbol{\Omega}}_{[k+i]}$ can be expressed as:

$$\begin{pmatrix} \boldsymbol{I}_{3\times3} \\ -\boldsymbol{I}_{3\times3} \end{pmatrix}\dot{\boldsymbol{\Omega}}_{[k+i]} \leq \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{\omega}_{\max} \\ -\boldsymbol{M}_{[k+i-1]}\boldsymbol{\omega}_{\min} \end{pmatrix}. \tag{31}$$

According to the theory of constrained model predictive control, the simulation tool "quadprog" can be utilized to obtain a set of optimal predicted control increment vectors for incremental control law (22), subject to the constraints (30) and (31), as well as the quadratic performance criterion (26). The first element of the predicted control increment vector, as shown in (32), corresponds to the virtual control input vector base on model predictive control theory [31,32]. This predictive control process is repeated to achieve MPC receding horizon control theory.

$$\boldsymbol{v}_{[k]} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \cdots & \boldsymbol{0}_{3\times3} \end{bmatrix}\delta\boldsymbol{V}_{[k]}^{*} + \boldsymbol{v}_{[k-1]} \tag{32}$$

Based on the design principles outlined above, the attitude control of the quadrotor UAV is achieved through feedback linearization model predictive control, with the control law computation given by (15).

4. Simulation and Results Analysis

4.1. Methods

To verify the effectiveness of the proposed controller, simulations were performed using the quadrotor parameters listed in Table 1. The simulations were conducted on

a computer with an Intel i5-9400F processor (Intel, Santa Clara, CA, USA) and 16 GB of memory.

Table 1. Parameters of quadrotor.

Parameters	Values	Unit
m	0.8	kg
I_x	9.7×10^{-3}	kg·m^2
I_y	9.7×10^{-3}	kg·m^2
I_z	1.7×10^{-2}	kg·m^2
l	0.06	m
g	9.81	m/s^2

The control parameters are Ts = 0.01 s, with $N_p = N_c = 10$. The range for the three-axis angular velocities are [−1.5, 1.5], and the constraints on the three control inputs are [−1, 1]. The gains of the FLMPC controller are set as $\boldsymbol{S}$ = diag(1, 1, 1, 1, 1, 1), $\boldsymbol{Q}$ = diag(100, 1, 100, 1, 25, 1), $\boldsymbol{R}$ = diag(0.001, 0.001, 0.001). A comparison is made with LMPC [15] and adaptive MPC (AMPC) [18], where the gains for both controllers are $\boldsymbol{S}$ = diag(1, 1, 1, 1, 1, 1), $\boldsymbol{Q}$ = diag(100, 100, 25, 1, 1, 1), $\boldsymbol{R}$ = diag(10, 10, 10). The parameters of the three controllers remained constant throughout the simulations to ensure fair comparison.

The simulation starts with the quadrotor UAV in an initial hover state, with attitude angles $\phi = \theta = \psi = 0$. A target pitch angle of 0.2 rad and a sinusoidal roll angle with an amplitude of 0.5 rad were applied to simulate the quadrotor's flight with oscillatory motion, while maintaining the yaw angle at 0 rad.

The second case aims to evaluate the designed controller's ability in handling sudden changes and assess their step response performance in maintaining stability and accurately reaching the specified target angles. Attitude targets of 0.8 rad for pitch, 0.2 rad for roll and 1.3 rad for yaw are applied while the quadrotor UAV is in the hover state.

To further validate the robustness and disturbance rejection capability of the designed controller, a step input with an amplitude of 1 rad was applied individually to each attitude angle and each angle was subjected to external white noise disturbances with a standard deviation of 0.01 rad, to simulate the uncertain disturbances encountered in real flight.

4.2. Results

The results are illustrated in the following figures. In Figures 3 and 4, (a) shows the time response of the quadrotor's attitude angles, (b) shows the angular velocities, (c) demonstrates the variation of the three control inputs over time and (d) displays the computation time of each controller. Figure 5 presents the white noise disturbances applied during the simulation. Figure 6 exhibits the attitude tracking curves in step inputs with disturbances.

As shown in Figure 3a, all controllers effectively track the desired input in pitch control with an approximate 270 ms delay behind the command curve. In terms of roll control, all controllers demonstrate comparable performance, following the target command accurately within 650 ms. For yaw control, the FLMPC controller maintains a steady angle close to 0. In contrast, the LMPC controller exhibits slight fluctuations, while the AMPC controller experiences more noticeable fluctuations. Integrating these observations with the computational time in Figure 3d, it is evident that the FLMPC controller achieves superior control performance with relatively lower computational cost, requiring only one-tenth of the computational time of the AMPC.

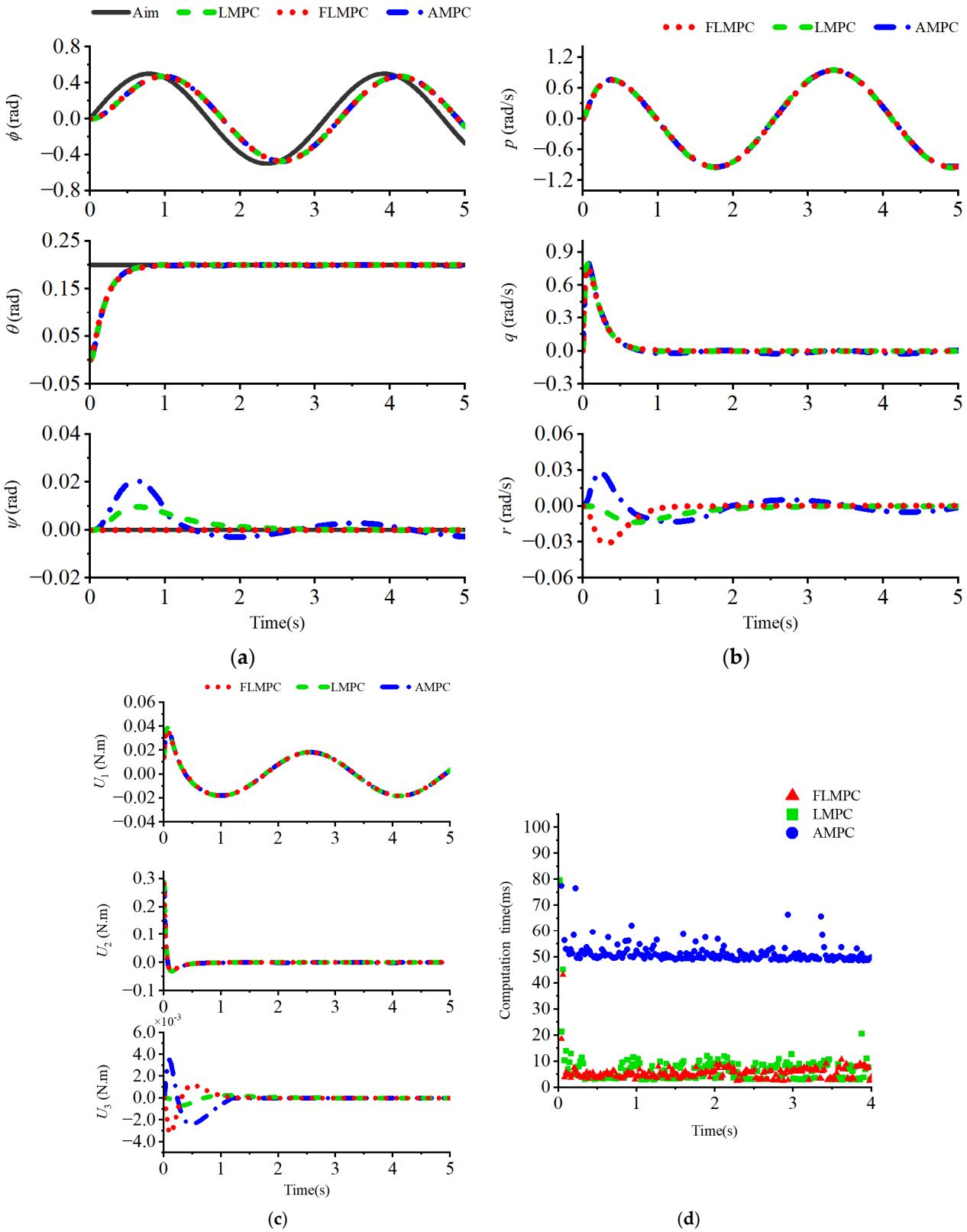


Figure 3. Controller performance in roll oscillations and pitch step. (**a**) Angular attitudes curves. (**b**) Angular velocities curves. (**c**) Control inputs curves. (**d**) Computation time.

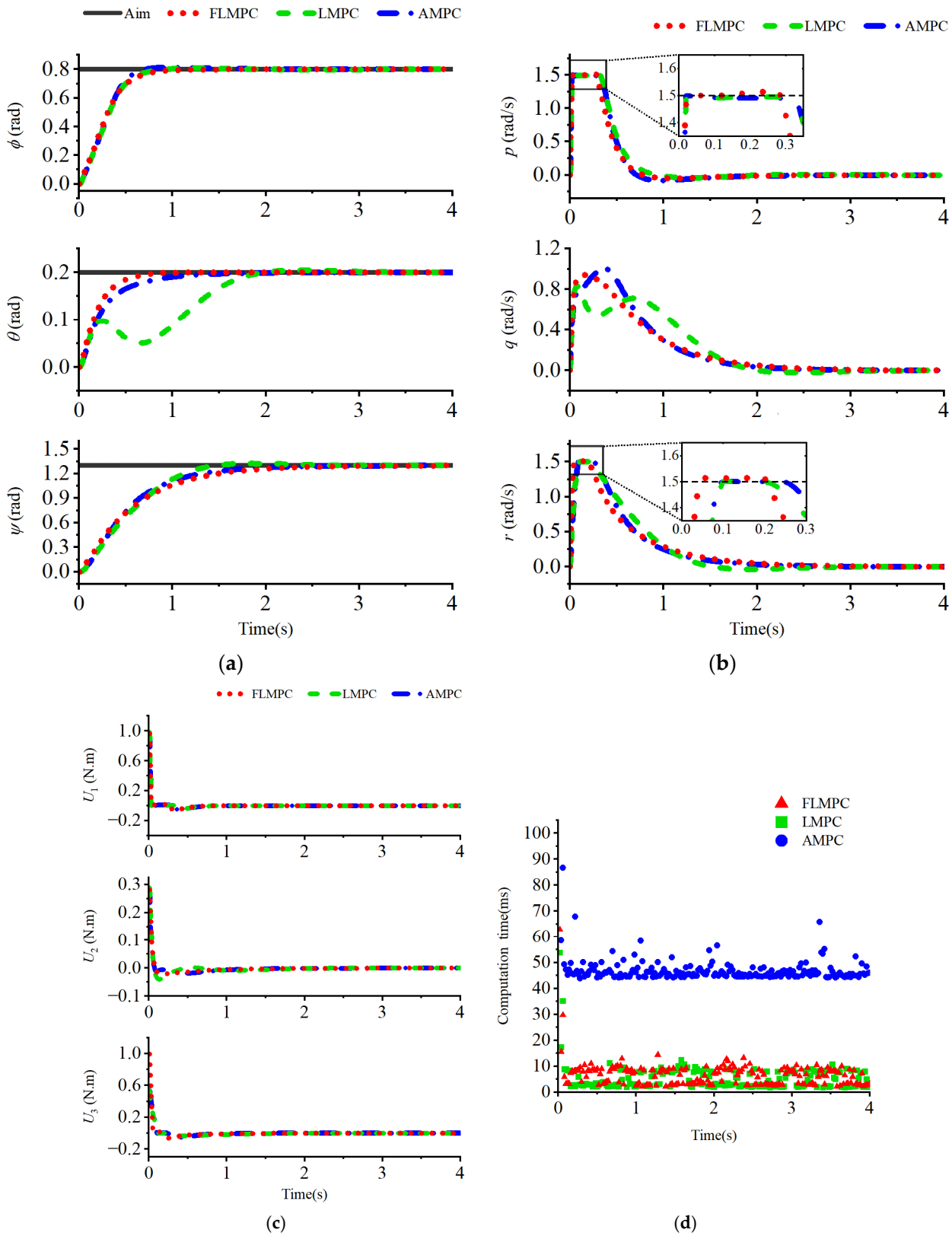


Figure 4. Controller performance in roll, pitch and yaw step inputs. (**a**) Angular attitudes curves. (**b**) Angular velocities curves. (**c**) Control inputs curves. (**d**) Computation time.

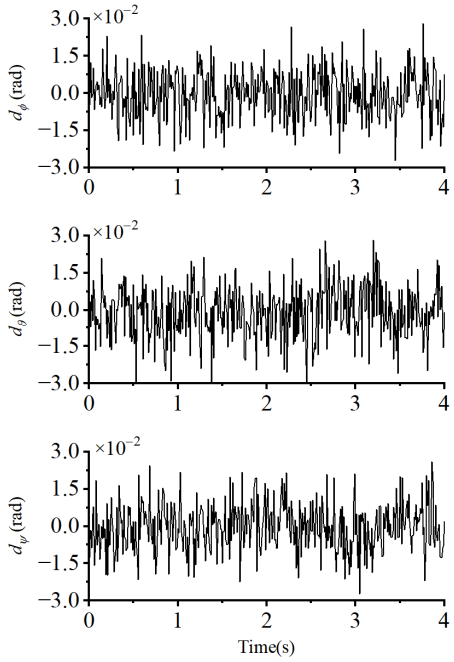


Figure 5. Disturbances applied to each attitude angle.

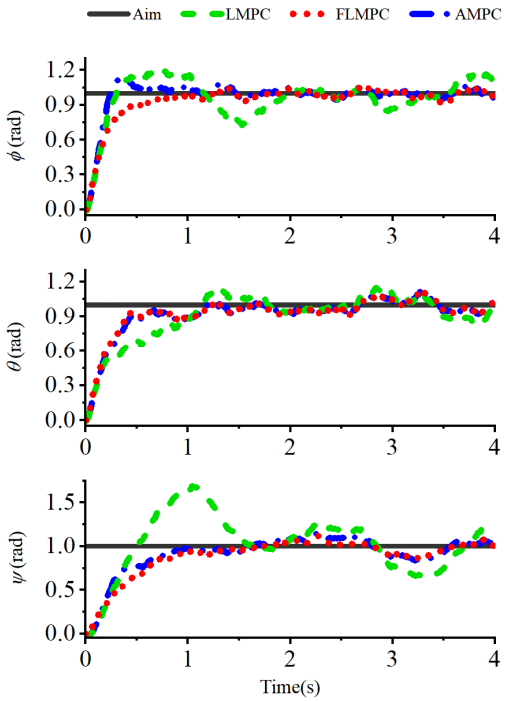


Figure 6. Attitude tracking curves in step inputs with disturbances.

As observed in Figure 4a, all MPC controllers effectively reach and stabilize at their target values with minimal overshoot. FLMPC provides the fastest response for the pitch angle, stabilizing in approximately 0.95 s, followed by AMPC at around 1.65 s, and LMPC, which is influenced by channel coupling, taking the longest at about 2.7 s. For the roll and yaw angles, all controllers demonstrate similar settling times, though LMPC shows greater fluctuation due to coupling effects. Figure 4b shows that the variations in p and r remain well within their upper constraints, while Figure 4c illustrates that the maximum control inputs do not exceed their specified boundaries. Table 2 provides a summary of key performance metrics, including response time, overshoot and mean computation time for three controllers.

Table 2. Key performance metrics for the three controllers.

Performance Metrics	Response Time (s)			Overshoot (%)			Mean Computation Time (ms)
	ϕ	θ	ψ	ϕ	θ	ψ	
FLMPC	0.83	0.95	2.07	0.46	0.00	0.01	6.03
LMPC	0.76	2.7	2.25	1.07	2.17	1.88	5.23
AMPC	0.63	1.65	1.90	1.67	0.00	0.00	47.59

The FLMPC and AMPC controllers demonstrated superior performance in Figure 6. Both of them showing fluctuations under disturbances of less than 10%, while LMPC exhibited notable fluctuations, with a maximum overshoot of 50%, indicating a lower robustness to disturbances. These results confirm that FLMPC provides robust and stable control across all attitude angles, mitigating the impact of external disturbances effectively.

4.3. Discussion

The simulation results indicate that for systems with nonlinear coupling characteristics, all three controllers (FLMPC, LMPC and AMPC) are able to achieve target control under constraints and disturbances. However, the LMPC method shows significant shortcomings in handling inter-axis coupling, particularly in rapidly changing environments where it fails to meet high precision requirements and performs poorly when disturbances are encountered. On the other hand, while AMPC is capable of handling nonlinearities effectively, its computational time per iteration is relatively long, posing challenges for real-time applications, especially in cases with limited computational resources.

In contrast to the two methods mentioned above, the model predictive method combined with feedback linearization in this study offers distinct advantages in dealing with system nonlinearities and disturbances, along with outstanding computational efficiency. Due to its lower computational burden, FLMPC can update control inputs in real time, maintaining high-precision control performance. Furthermore, FLMPC exhibits excellent disturbance rejection capabilities, effectively mitigating the impact of external disturbances and ensuring system stability and accuracy. Thus, FLMPC is particularly well-suited for engineering applications that require high real-time performance when dealing with complex nonlinear systems.

Overall, FLMPC strikes a balance between control performance and computational efficiency. It not only outperforms the other methods in simulations but also holds strong potential for practical applications. Particularly in high-dynamic environments, FLMPC offers significant advantages over traditional control methods, ensuring superior control performance while reducing computational overhead and enhancing system reliability.

5. Conclusions

In this work, a feedback linearization model predictive control (FLMPC) approach is proposed to address the nonlinear and coupled characteristics of quadrotor UAV attitude dynamics. By explicitly incorporating the nonlinear mapping between Euler angular velocities and body-frame angular velocities into the feedback linearization process, the proposed method achieves effective linearization and decoupling of roll, pitch and yaw channels. This allows the design of an MPC controller on a linearized model with appropriately transformed constraints, thereby ensuring both theoretical soundness and practical feasibility.

Simulation results demonstrate the effectiveness of the proposed FLMPC method under various constraint and disturbance conditions. Comparative analyses with LMPC and AMPC show that, while all controllers can achieve the desired tracking performance, FLMPC consistently offers more stable and precise tracking, stronger disturbance rejection,

and lower computational cost compared to AMPC. Unlike LMPC, which suffers from axis coupling, FLMPC successfully mitigates this issue through dynamic decoupling. These advantages make FLMPC a promising candidate for real-time attitude control of quadrotors in nonlinear and constrained environments.

However, it is also acknowledged that despite its improved computational efficiency, the optimization process in FLMPC still imposes a notable computational burden for embedded processors. As such, future work will focus on the development of faster and more efficient optimization algorithms, or approximation-based solvers, to further enhance the real-time applicability of the proposed method in practical embedded systems.

Author Contributions: Methodology, X.Y.; Validation, X.Y.; Writing—review & editing, J.X.; Funding acquisition, J.X.; Investigation, S.L. All authors contributed extensively to the current study. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Laboratory of Rotorcraft Aeromechanics Fund Program of China (9140C400504130C4148); supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Author Shengwei Li was employed by the company Harbin Aircraft Industry Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Attitude Controller Design Based on MPC

(1) Discretization

For the state–space equation of a continuous linear time-invariant system (12), the solution of the system is:

$$\boldsymbol{\varkappa}_{(t)} = \mathrm{e}^{\boldsymbol{A}(t-t_0)}\boldsymbol{\varkappa}_{(t_0)} + \mathrm{e}^{\boldsymbol{A}t}\int_{t_0}^{t}\mathrm{e}^{-\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{v}_{(\tau)}\mathrm{d}\tau. \tag{A1}$$

Considering the sampling time from t_k to t_{k+1}, it can be concluded that:

$$\boldsymbol{\varkappa}_{(t_{k+1})} = \mathrm{e}^{\boldsymbol{A}(t_{k+1}-t_k)}\boldsymbol{\varkappa}_{(t_k)} + \mathrm{e}^{\boldsymbol{A}t_{k+1}}\int_{t_k}^{t_{k+1}}\mathrm{e}^{-\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{v}_{(\tau)}\mathrm{d}\tau. \tag{A2}$$

Using the zero-order hold method and setting $\boldsymbol{v}_{(t_k)} = \boldsymbol{v}_{(t_{k+1})}$, Equation (A2) can be written as:

$$\boldsymbol{\varkappa}_{(t_{k+1})} = \mathrm{e}^{\boldsymbol{A}(t_{k+1}-t_k)}\boldsymbol{\varkappa}_{(t_k)} + \int_{t_k}^{t_{k+1}}\mathrm{e}^{\boldsymbol{A}(t_{k+1}-\tau)}\mathrm{d}\tau\boldsymbol{B}\boldsymbol{v}_{(t_k)}. \tag{A3}$$

Letting $\boldsymbol{A}_\mathrm{d} = \mathrm{e}^{\boldsymbol{A}(t_{k+1}-t_k)} = \mathrm{e}^{\boldsymbol{A}T_s}$, $\boldsymbol{B}_\mathrm{d} = \int_{t_k}^{t_{k+1}}\mathrm{e}^{\boldsymbol{A}(t_{k+1}-\tau)}\mathrm{d}\tau\boldsymbol{B} = -\boldsymbol{A}^{-1}\mathrm{e}^{\boldsymbol{A}(t_{k+1}-\tau)}\big|_{t_k}^{t_{k+1}}\boldsymbol{B} = \boldsymbol{A}^{-1}(\mathrm{e}^{\boldsymbol{A}T_s}-\boldsymbol{I})\boldsymbol{B}$, the discrete-time state–space equation can be derived:

$$\boldsymbol{\varkappa}_{[k+1]} = \boldsymbol{A}_\mathrm{d}\boldsymbol{\varkappa}_{[k]} + \boldsymbol{B}_\mathrm{d}\boldsymbol{v}_{[k]}. \tag{A4}$$

(2) Transformation of the MIMO system

To achieve control of a non-zero target and eliminate static error, rewrite Equation (A4) into an incremental form:

$$\boldsymbol{\varkappa}_{[k+1]} = \boldsymbol{A}_\mathrm{d}\delta\boldsymbol{\varkappa}_{[k]} + \boldsymbol{B}_\mathrm{d}\delta\boldsymbol{v}_{[k]} + \boldsymbol{\varkappa}_{[k]}. \tag{A5}$$

where $\delta \boldsymbol{v}_{[k]} = \boldsymbol{v}_{[k]} - \boldsymbol{v}_{[k-1]}$, $\delta \boldsymbol{\varkappa}_{[k]} = \boldsymbol{\varkappa}_{[k]} - \boldsymbol{\varkappa}_{[k-1]}$.

Based on the state vector $\boldsymbol{\varkappa}_{[k|k]}$ at step k, the state vector at step $k+1$ can be predicted as:

$$\begin{aligned} \boldsymbol{\varkappa}_{[k+1|k]} &= \delta \boldsymbol{\varkappa}_{[k+1|k]} + \boldsymbol{\varkappa}_{[k|k]} \\ &= \boldsymbol{A}_\mathrm{d} \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{\varkappa}_{[k|k]} \end{aligned} \tag{A6}$$

The state vector at step $k+2$ can be predicted as:

$$\begin{aligned} \boldsymbol{\varkappa}_{[k+2|k]} &= \delta \boldsymbol{\varkappa}_{[k+2|k]} + \boldsymbol{\varkappa}_{[k+1|k]} \\ &= \boldsymbol{A}_\mathrm{d} \delta \boldsymbol{\varkappa}_{[k+1|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+1|k]} + \boldsymbol{\varkappa}_{[k+1|k]} \\ &= \boldsymbol{A}_\mathrm{d}{}^2 \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{A}_\mathrm{d} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+1|k]} + \boldsymbol{A}_\mathrm{d} \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{\varkappa}_{[k|k]} \\ &= (\boldsymbol{A}_\mathrm{d}{}^2 + \boldsymbol{A}_\mathrm{d}) \delta \boldsymbol{\varkappa}_{[k|k]} + (\boldsymbol{A}_\mathrm{d} \boldsymbol{B}_\mathrm{d} + \boldsymbol{B}_\mathrm{d}) \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+1|k]} + \boldsymbol{\varkappa}_{[k|k]} \end{aligned} \tag{A7}$$

where $\delta \boldsymbol{v}_{[k+i|k]}$ is the control input increment at step $k+i$ predicted at step k; $\boldsymbol{\varkappa}_{[k+i|k]}$ is the system state vector predicted at step $k+i$ from step k.

Similarly, the state vector at step $k+N_\mathrm{c}$ and at step $k+N_\mathrm{p}$ are given by:

$$\begin{aligned} \boldsymbol{\varkappa}_{[k+N_\mathrm{c}|k]} &= \delta \boldsymbol{\varkappa}_{[k+N_\mathrm{c}|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{c}-1|k]} \\ &= \boldsymbol{A}_\mathrm{d} \delta \boldsymbol{\varkappa}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{c}-1|k]} \\ &= \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{c}} \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{c}-1} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{c}-2} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+1|k]} + \cdots + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{c}-1|k]} \\ &= \sum_{i=1}^{N_\mathrm{c}} \boldsymbol{A}_\mathrm{d}{}^i \delta \boldsymbol{\varkappa}_{[k|k]} + \sum_{i=0}^{N_\mathrm{c}-1} \boldsymbol{A}_\mathrm{d}{}^i \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \cdots + \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{\varkappa}_{[k|k]} \end{aligned} \tag{A8}$$

$$\begin{aligned} \boldsymbol{\varkappa}_{[k+N_\mathrm{p}|k]} &= \delta \boldsymbol{\varkappa}_{[k+N_\mathrm{p}|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{p}-1|k]} \\ &= \boldsymbol{A}_\mathrm{d} \delta \boldsymbol{\varkappa}_{[k+N_\mathrm{p}-1|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{p}-1|k]} \\ &= \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{p}} \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{p}-1} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{p}-2} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+1|k]} + \cdots + \boldsymbol{A}_\mathrm{d}{}^{N_\mathrm{p}-N_\mathrm{c}} \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{\varkappa}_{[k+N_\mathrm{p}-1|k]} \\ &= \sum_{i=1}^{N_\mathrm{p}} \boldsymbol{A}_\mathrm{d}{}^i \delta \boldsymbol{\varkappa}_{[k|k]} + \sum_{i=0}^{N_\mathrm{p}-1} \boldsymbol{A}_\mathrm{d}{}^i \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k|k]} + \cdots + \sum_{i=0}^{N_\mathrm{p}-N_\mathrm{c}} \boldsymbol{A}_\mathrm{d}{}^i \boldsymbol{B}_\mathrm{d} \delta \boldsymbol{v}_{[k+N_\mathrm{c}-1|k]} + \boldsymbol{\varkappa}_{[k|k]} \end{aligned} \tag{A9}$$

where N_c is the control horizon and N_p is the prediction horizon. To reduce the computational burden, the control inputs will be held constant at the value from the previous time step once the time exceeds the control horizon.

By combining Equations (A6) through (A9), the control input increment vector $\delta \boldsymbol{V}$ and state vector $\boldsymbol{Z}$ can be defined as:

$$\delta \boldsymbol{V}_{[k]} = \begin{pmatrix} \delta \boldsymbol{v}_{[k|k]} & \delta \boldsymbol{v}_{[k+1|k]} & \cdots & \delta \boldsymbol{v}_{[k+N_\mathrm{c}|k]} \end{pmatrix}^\mathrm{T}, \tag{A10}$$

$$\boldsymbol{Z}_{[k]} = \begin{pmatrix} \boldsymbol{\varkappa}_{[k+1|k]} & \boldsymbol{\varkappa}_{[k+2|k]} & \cdots & \boldsymbol{\varkappa}_{[k+N_\mathrm{p}|k]} \end{pmatrix}^\mathrm{T}. \tag{A11}$$

Then, the predicted states from step $k+1$ to $k+N_\mathrm{p}$ can then be written as:

$$\boldsymbol{Z}_{[k]} = \boldsymbol{P} \delta \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{I} \boldsymbol{\varkappa}_{[k|k]} + \boldsymbol{H} \delta \boldsymbol{V}_{[k]}, \tag{A12}$$

where

$$\boldsymbol{P}=\begin{pmatrix}\boldsymbol{A}_{\mathrm{d}}\\ \sum_{i=1}^{2}\boldsymbol{A}_{\mathrm{d}}{}^{i}\\ \vdots\\ \sum_{i=1}^{N_{\mathrm{p}}}\boldsymbol{A}_{\mathrm{d}}{}^{i}\end{pmatrix}_{N_{\mathrm{p}}\times 1},\ \boldsymbol{I}=\begin{pmatrix}\boldsymbol{I}_{n\times n}\\ \boldsymbol{I}_{n\times n}\\ \vdots\\ \boldsymbol{I}_{n\times n}\end{pmatrix}_{N_{\mathrm{p}}\times 1},\ \boldsymbol{H}=\begin{pmatrix}\boldsymbol{B}_{\mathrm{d}} & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \sum_{i=0}^{1}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}} & \boldsymbol{B}_{\mathrm{d}} & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{i=0}^{N_{\mathrm{c}}-1}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}} & \sum_{i=0}^{N_{\mathrm{c}}-2}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}} & \cdots & \boldsymbol{B}_{\mathrm{d}}\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{i=0}^{N_{\mathrm{p}}-1}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}} & \sum_{i=0}^{N_{\mathrm{p}}-2}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}} & \cdots & \sum_{i=0}^{N_{\mathrm{p}}-N_{\mathrm{c}}}\boldsymbol{A}_{\mathrm{d}}{}^{i}\boldsymbol{B}_{\mathrm{d}}\end{pmatrix}_{N_{\mathrm{p}}\times N_{\mathrm{c}}}.$$

(3) Cost Function

For a given target $\boldsymbol{\varkappa}_{d[k+i]}$, the cost function that reflects the system performance requirements at time k can be defined as:

$$J=\left\|\boldsymbol{\varkappa}_{[k+N_{\mathrm{p}}|k]}-\boldsymbol{\varkappa}_{d[k+N_{\mathrm{p}}]}\right\|_{\boldsymbol{S}}^{2}+\sum_{i=1}^{N_{\mathrm{p}}-1}\left\|\boldsymbol{\varkappa}_{[k+i|k]}-\boldsymbol{\varkappa}_{d[k+i]}\right\|_{\boldsymbol{Q}}^{2}+\sum_{i=0}^{N_{\mathrm{c}}-1}\left\|\delta\boldsymbol{v}_{[k+i|k]}\right\|_{\boldsymbol{R}}^{2}\tag{A13}$$

In the above equation, the first term represents the terminal cost, which helps eliminate static errors. The second term corresponds to the running cost, reflecting the deviation between the state variables and the desired target values. The third term represents the input cost, capturing the magnitude of the control effort exerted on the system. The relative importance of each objective is determined by the weight matrices $\boldsymbol{S}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$, which are positive definite matrices.

Define the target vector $\boldsymbol{Z}_{d[k]}=\begin{pmatrix}\boldsymbol{\varkappa}_{d[k+1]} & \boldsymbol{\varkappa}_{d[k+2]} & \cdots & \boldsymbol{\varkappa}_{d[k+N_{\mathrm{p}}]}\end{pmatrix}^{\mathrm{T}}$. The above equation can be written in matrix–vector form as:

$$J=\left\|\boldsymbol{Z}_{[k]}-\boldsymbol{Z}_{d[k]}\right\|_{\overline{\boldsymbol{Q}}}^{2}+\left\|\delta\boldsymbol{V}_{[k]}\right\|_{\overline{\boldsymbol{R}}}^{2},\tag{A14}$$

where $\overline{\boldsymbol{Q}}=\mathrm{diag}\begin{pmatrix}\boldsymbol{Q} & \cdots & \boldsymbol{Q} & \boldsymbol{S}\end{pmatrix}$, $\overline{\boldsymbol{R}}=\mathrm{diag}\begin{pmatrix}\boldsymbol{R} & \cdots & \boldsymbol{R}\end{pmatrix}$.

Substituting the state prediction Equation (A12) into the cost function (A14), and defining

$$\boldsymbol{E}_{[k]}=\boldsymbol{Z}_{d[k]}-\boldsymbol{P}\delta\boldsymbol{\varkappa}_{[k|k]}-\boldsymbol{I}\boldsymbol{\varkappa}_{[k|k]},\tag{A15}$$

the following is obtained:

$$\begin{aligned}J &= \left\|\boldsymbol{H}\delta\boldsymbol{V}_{[k]}-\boldsymbol{E}_{[k]}\right\|_{\overline{\boldsymbol{Q}}}^{2}+\left\|\delta\boldsymbol{V}_{[k]}\right\|_{\overline{\boldsymbol{R}}}^{2}\\ &= \tfrac{1}{2}\left(\boldsymbol{H}\delta\boldsymbol{V}_{[k]}-\boldsymbol{E}_{[k]}\right)^{\mathrm{T}}\overline{\boldsymbol{Q}}\left(\boldsymbol{H}\delta\boldsymbol{V}_{[k]}-\boldsymbol{E}_{[k]}\right)+\tfrac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{R}}\delta\boldsymbol{V}_{[k]}\\ &= \tfrac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H}\delta\boldsymbol{V}_{[k]}-\tfrac{1}{2}\boldsymbol{E}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H}\delta\boldsymbol{V}_{[k]}-\tfrac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]}\\ &\quad+\tfrac{1}{2}\boldsymbol{E}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]}+\tfrac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{R}}\delta\boldsymbol{V}_{[k]}\end{aligned}.\tag{A16}$$

Since $\frac{1}{2}\boldsymbol{E}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H}\delta\boldsymbol{V}_{[k]}$ and $\frac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]}$ are transposes of each other and both are scalars, Equation (A16) can be written as:

$$J=\frac{1}{2}\delta\boldsymbol{V}_{[k]}^{\mathrm{T}}\left(\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H}+\overline{\boldsymbol{R}}\right)\delta\boldsymbol{V}_{[k]}-\boldsymbol{E}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H}\delta\boldsymbol{V}_{[k]}+\frac{1}{2}\boldsymbol{E}_{[k]}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]}.\tag{A17}$$

Furthermore, since the last term in the above equation is determined by the initial state values $\boldsymbol{\varkappa}_{[k|k]}$ and $\delta\boldsymbol{\varkappa}_{[k|k]}$, and is unaffected by $\delta\boldsymbol{V}_{[k]}$, it can be disregarded during optimization.

Therefore, a standard quadratic performance index in terms of $\delta \boldsymbol{V}_{[k]}$ can be formulated as:

$$J = \frac{1}{2}\delta \boldsymbol{V}_{[k]}^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{H} + \overline{\boldsymbol{R}})\delta \boldsymbol{V}_{[k]} - (\boldsymbol{H}^{\mathrm{T}}\overline{\boldsymbol{Q}}\boldsymbol{E}_{[k]})^{\mathrm{T}}\delta \boldsymbol{V}_{[k]}. \tag{A18}$$

(4) System Constraints

Feedback linearization introduces new state variables and virtual control quantities. The constraints on the angular velocity rate $\boldsymbol{\omega}$ and the control input $\boldsymbol{u}$ of the quadrotor cannot be directly applied to the feedback linearized system. Instead, an indirect constraint method is employed, where the constraints from the original system are mapped to the new system at each time step.

For the control input $\boldsymbol{u}_{[k+i]}$, where $i = 1, 2, \ldots, N_{\mathrm{p}}$, there are upper and lower boundary vectors $\boldsymbol{u}_{\max}$ and $\boldsymbol{u}_{\min}$:

$$\boldsymbol{u}_{\min} \le \boldsymbol{u}_{[k+i]} \le \boldsymbol{u}_{\max}. \tag{A19}$$

Substituting the feedback linearization control law $\boldsymbol{u} = \boldsymbol{M}(\boldsymbol{\tau}, \boldsymbol{\omega})^{-1}(\boldsymbol{v} - \boldsymbol{N}(\boldsymbol{\tau}, \boldsymbol{\omega}, \varkappa))$ into the above equation, it can be derived that:

$$\boldsymbol{u}_{\min} \le \boldsymbol{M}_{[k+i-1]}^{-1}(\boldsymbol{v}_{[k+i]} - \boldsymbol{N}_{[k+i-1]}) \le \boldsymbol{u}_{\max}. \tag{A20}$$

Then, the boundaries of $\boldsymbol{u}_{[k+i]}$ can be expressed as inequality constraints on $\boldsymbol{v}_{[k+i]}$:

$$\begin{pmatrix} \boldsymbol{I}_3 \\ -\boldsymbol{I}_3 \end{pmatrix} \boldsymbol{v}_{[k+i]} \le \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\max} + \boldsymbol{N}_{[k+i-1]} \\ -(\boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\min} + \boldsymbol{N}_{[k+i-1]}) \end{pmatrix}. \tag{A21}$$

Since $\boldsymbol{v}_{[k+i]} = \boldsymbol{v}_{[k+i-1]} + \delta \boldsymbol{v}_{[k+i]}$, the Equation (A21) can be written as:

$$\begin{pmatrix} \boldsymbol{I}_3 \\ -\boldsymbol{I}_3 \end{pmatrix} \delta \boldsymbol{v}_{[k+i]} \le \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\max} + \boldsymbol{N}_{[k+i-1]} - \boldsymbol{v}_{[k+i-1]} \\ -(\boldsymbol{M}_{[k+i-1]}\boldsymbol{u}_{\min} + \boldsymbol{N}_{[k+i-1]} - \boldsymbol{v}_{[k+i-1]}) \end{pmatrix}. \tag{A22}$$

Similarly, the upper and lower boundary vectors of the angular velocity rate $\boldsymbol{\omega}_{[k+i]}$ can be expressed in the following form:

$$\boldsymbol{\omega}_{\min} \le \boldsymbol{M}_{[k+i-1]}^{-1}\dot{\boldsymbol{\Omega}}_{[k+i]} \le \boldsymbol{\omega}_{\max}. \tag{A23}$$

Since $\dot{\boldsymbol{\Omega}}_{[k+i]} = \boldsymbol{M}_{[k+i-1]}\boldsymbol{\omega}_{[k+i]}$, the inequality constraints on the $\dot{\boldsymbol{\Omega}}_{[k+i]}$ can be derived as:

$$\begin{pmatrix} \boldsymbol{I}_3 \\ -\boldsymbol{I}_3 \end{pmatrix} \dot{\boldsymbol{\Omega}}_{[k+i]} \le \begin{pmatrix} \boldsymbol{M}_{[k+i-1]}\boldsymbol{\omega}_{\max} \\ -\boldsymbol{M}_{[k+i-1]}\boldsymbol{\omega}_{\min} \end{pmatrix}. \tag{A24}$$

For quadratic programming problems with inequality constraints, analytical solutions are often difficult to obtain, so numerical methods are used. Substituting Equations (A18), (A22) and (A24) into the quadprog function, the virtual control increment sequence $\delta \boldsymbol{V}_{[k]}^{*}$ can be solved. The feedback linearized system's virtual control vector is

$$\boldsymbol{v}_{[k]} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \cdots & \boldsymbol{0}_3 \end{bmatrix} \delta \boldsymbol{V}_{[k]}^{*} + \boldsymbol{v}_{[k-1]} \tag{A25}$$

References

1. Li, B.; Wang, Y. An Enhanced Model Predictive Controller for Quadrotor Attitude Quick Adjustment with Input Constraints and Disturbances. *Int. J. Control Autom. Syst.* **2022**, *20*, 648–659. [CrossRef]
2. Thusoo, R.; Jain, S.; Bangia, S. Quadrotors in the Present Era: A Review. *Inf. Technol. Ind.* **2021**, *9*, 164–178.
3. Lei, B.; Wang, N.; Xu, P.; Song, G. New crack detection method for bridge inspection using UAV incorporating image processing. *J. Aerosp. Eng.* **2018**, *31*, 04018058. [CrossRef]

4. Idrissi, M.; Annaz, F. Dynamic modelling and analysis of a quadrotor based on selected physical parameters. *Int. J. Mech. Eng. Robot. Res. (IJMERR)* **2020**, *9*, 784–790. [CrossRef]
5. Li, C.; Wang, Y.; Yang, X. Adaptive fuzzy control of a quadrotor using disturbance observer. *Aerosp. Sci. Technol.* **2022**, *128*, 107784. [CrossRef]
6. Roy, R.; Islam, M.; Sadman, N.; Mahmud, M.P.; Gupta, K.D.; Ahsan, M.M. A Review on Comparative Remarks, Performance Evaluation and Improvement Strategies of Quadrotor Controllers. *Technologies* **2021**, *9*, 37. [CrossRef]
7. Hassan, S.; Ali, A.; Rafic, Y. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerosp. Electron. Syst. Mag.* **2018**, *33*, 14–33.
8. Sheta, A.; Braik, M.; Maddi, D.R.; Mahdy, A.; Aljahdali, S.; Turabieh, H. Optimization of PID Controller to Stabilize Quadcopter Movements Using Meta-Heuristic Search Algorithms. *Appl. Sci.* **2021**, *11*, 6492. [CrossRef]
9. Al Tahtawi, A.R.; Yusuf, M. Low-cost quadrotor hardware design with PID control system as flight controller. *Telecommun. Comput. Electron. Control* **2019**, *17*, 1923–1930. [CrossRef]
10. Mirtaba, M.; Jeddi, M.; Nikoofard, A.; Shirmohammadi, Z. Design and implementation of a low-complexity flight controller for a quadrotor UAV. *Int. J. Dyn. Control* **2023**, *11*, 689–700. [CrossRef]
11. Qiao, Z.; Zhuang, K.; Zhao, T.; Xue, J.; Zhang, M.; Cui, S.; Gao, Y. Simulation of a Quadrotor under Linear Active Disturbance Rejection. *Appl. Sci.* **2022**, *12*, 12455. [CrossRef]
12. Labbadi, M.; Cherkaoui, M. Robust Integral Terminal Sliding Mode Control for Quadrotor UAV with External Disturbances. *Int. J. Aerosp. Eng.* **2019**, *2019*, 2016416. [CrossRef]
13. Pan, J.; Shao, B.; Xiong, J.; Zhang, Q. Attitude Control of Quadrotor UAVs Based on Adaptive Sliding Mode. *Int. J. Control Autom. Syst.* **2023**, *21*, 2698–2707. [CrossRef]
14. Islam, M.; Okasha, M. A Comparative Study of PD, LQR and MPC on Quadrotor Using Quaternion Approach. In Proceedings of the 2019 7th International Conference on Mechatronics Engineering (ICOM), Putrajaya, Malaysia, 30–31 October 2019; pp. 1–6.
15. Lopes, R.V.; Santana, P.H.R.Q.A.; Borges, G.; Ishihara, J. Model predictive control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft. In Proceedings of the 21st International Congress of Mechanical Engineering, Natal, Brazil, 24–28 October 2011; pp. 176–185.
16. Chikasha, P.N.; Dube, C. Adaptive Model Predictive Control of a Quadrotor. *Ifac Pap.* **2017**, *50*, 157–162. [CrossRef]
17. Villarreal, O.J.G.; Rossiter, J.A.; Shin, H. Laguerre-based adaptive MPC for attitude stabilization of quad-rotor. In Proceedings of the 2018 UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, 5–7 September 2018; pp. 360–365.
18. Benotsmane, R.; Vásárhelyi, J. Towards Optimization of Energy Consumption of Tello Quad-Rotor with Mpc Model Implementation. *Energies* **2022**, *15*, 9207. [CrossRef]
19. Murilo, A.; Lopes, R.V. Unified NMPC framework for attitude and position control for a VTOL UAV. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2019**, *233*, 889–903. [CrossRef]
20. Kuantama, E.; Tarca, I.; Tarca, R. Feedback linearization LQR control for quadcopter position tracking. In Proceedings of the 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, Greece, 10–13 April 2018; pp. 204–209.
21. Kong, X.; Abdelbaky, M.A.; Liu, X.; Lee, K.Y. Stable feedback linearization-based economic MPC scheme for thermal power plant. *Energy* **2023**, *268*, 126658. [CrossRef]
22. Te Braake, H.A.B.; Ayala Botto, M.; Sá da Costa, J.; van Can, H.J.L.; Verbruggen, H.B. Linear predictive control based on approximate input-output feedback linearisation. *IEE Proc.-Control Theory Appl.* **1999**, *146*, 295–300. [CrossRef]
23. Ermeydan, A.; Kaba, A. Feedback linearization control of a quadrotor. In Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 21–23 October 2021; pp. 287–290.
24. Palunko, I.; Fierro, R. Adaptive control of a quadrotor with dynamic changes in the center of gravity. *IFAC Proc. Vol.* **2011**, *44*, 2626–2631. [CrossRef]
25. Rahmat, M.F.; Eltayeb, A.; Basri, M.A.M. Adaptive feedback linearization controller for stabilization of quadrotor UAV. *Int. J. Integr. Eng.* **2020**, *12*, 1–17.
26. Cai, Z.; Zhang, S.; Jing, X. Model predictive controller for quadcopter trajectory tracking based on feedback linearization. *IEEE Access* **2021**, *9*, 162909–162918. [CrossRef]
27. Baek, J.; Jung, J. A Model-Free Control Scheme for Attitude Stabilization of Quadrotor Systems. *Electronics* **2020**, *9*, 1586. [CrossRef]
28. Xiong, J.J.; Zheng, E.H. Position and attitude tracking control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 725–731. [CrossRef]
29. Shen, S.; Xu, J. Attitude active disturbance rejection control of the quadrotor and its parameter tuning. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8876177. [CrossRef]
30. Rinaldi, M.; Primatesta, S.; Guglieri, G. A Comparative Study for Control of Quadrotor UAVs. *Appl. Sci.* **2023**, *13*, 3464. [CrossRef]

31. Morato, M.M.; Normey-Rico, J.E.; Sename, O. Model predictive control design for linear parameter varying systems: A survey. *Annu. Rev. Control* **2020**, *49*, 64–80. [CrossRef]
32. Okasha, M.; Kralev, J.; Islam, M. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. *Aerospace* **2022**, *9*, 298. [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.