

Advanced MPC Controller with Disturbance Suppression and Input Delay Compensation for Spherical Quadrotor UAV

Yuan Fang, Limin Ouyang, Yuyun Xia and Yinghong Tian

Abstract—The field of unmanned aerial vehicles (UAVs) has gained significant attention in recent years, with a particular focus on resilience to flight perturbations and timely control inputs, both of which are essential for reliable system operation. To address these challenges, this study proposes a disturbance suppression mechanism that enhances system robustness by incorporating state mutations into the algorithmic model. Additionally, an input delay compensation mechanism is introduced, which reconstructs the state equation using lagged control inputs to determine the optimal control input for the current moment. Simulation experiments demonstrate that the optimized model predictive control (MPC) algorithm surpasses traditional control methods in terms of attitude stabilization, disturbance resistance, and input responsiveness. To further evaluate the proposed approach, a real spherical quadrotor UAV was constructed. The spherical design offers substantial anti-collision benefits but also introduces sensitivity to disturbances. Results validate the effectiveness of the optimized algorithm within the context of this spherical quadrotor UAV, demonstrating improved performance and resilience.

Index Terms—MPC algorithm optimization, Disturbance suppression, Input delay compensation, Spherical quadrotor UAV, Flight stability

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) have demonstrated significant and varied applications in various fields due to their versatility and multi-purpose functions. Rotary-wing UAVs, in contrast to fixed-wing UAVs, offer vertical take-off and landing (VTOL) and hovering capabilities. Among these, single-rotor UAVs were among the first models developed.

Manuscript received Jul 7, 2024; revised Mar 18, 2025.

Yuan Fang is a graduate student in the school of Communication and Electronic Engineering, East China Normal University, Shanghai, China. (email: 51255904057@stu.ecnu.edu.cn)

Limin Ouyang is a graduate student in the school of Communication and Electronic Engineering, East China Normal University, Shanghai, China. (email: 51255904058@stu.ecnu.edu.cn)

Yuyun Xia is a graduate student in the school of Communication and Electronic Engineering, East China Normal University, Shanghai, China. (email: 51215904079@stu.ecnu.edu.cn)

Yinghong Tian is an associate professor in the school of Communication and Electronic Engineering, East China Normal University, Shanghai, China. (email: yhtian@cee.ecnu.edu.cn)

Evandro et al. developed a single-rotor UAV that operates without a swashplate mechanism, utilizing torque modulation for roll and pitch control, thereby eliminating the need for a complex swashplate system [1].

O. C. Carholt et al. designed a novel vertical take-off single-rotor UAV [2], which demonstrates higher flight efficiency and greater payload capacity compared with quadcopters. However, due to having fewer rotors, single-rotor UAVs often experience compromised flight stability. To overcome this limitation and enable multi-mode operations, Elena S. et al. proposed a terrain-adaptive circular-rotor micro aerial vehicle [3]. This aircraft operates efficiently in air, on land, and in water, seamlessly transitioning between different modes of movement. Nonetheless, a primary challenge in circular drone design lies in the significant complexity associated with both structural and functional implementations.

Quadrotor UAVs demonstrate significant advantages over single-rotor UAVs in terms of flight stability, position control, and maneuverability [4]. However, current quadrotor UAVs face specific challenges, including restricted flight duration, unstable flight conditions, and weak anti-interference capabilities [5]. To mitigate these challenges, researchers have focused on developing new control algorithms and improving existing UAV designs. Based on previous research, this study proposes a novel spherical-structured UAV.

The development of control algorithms in the field of quadrotors has undergone several stages. Ranging from traditional PID control algorithms to more complex control methods, the overall control performance and functional capabilities have been continuously improved [6].

The diversification of application requirements and optimization of functionality and performance have driven the evolution of traditional quadcopters to more innovative structural designs. The integration of novel control algorithms and continuous innovation offers UAVs new potential. These advancements enable them to adapt and perform in increasingly complex and varied mission scenarios. Lotufo A. M. proposed a control method based on Active Disturbance Rejection Control (ADRC) and Extended State Observer (ESO), leading to the design of a digital attitude control unit that estimates and compensates for disturbance uncertainties in real-time, thereby achieving effective attitude control for quadcopters [7]. Ahmad

F. and colleagues proposed a control design for quadcopter position and yaw based on the Linear Quadratic Regulator (LQR) algorithm, which enhanced control performance through comprehensive simulations and the application of vertical integral compensation [8].

The Model Predictive Control (MPC) algorithm is a control approach that improves precision by optimizing inputs within a prediction horizon. It is widely applied in autonomous systems and quadrotor UAV fields [9].

D. Hanover and colleagues introduced an Adaptive Nonlinear MPC algorithm to address model uncertainties, enhancing quadcopter performance and robustness under dynamic conditions and disturbances. This approach reduces control errors without manual gain adjustments [10].

Numerous researchers have sought to enhance quadcopter UAV control performance by optimizing the MPC algorithm or integrating it with complementary control strategies. S. Sun et al. conducted a comparative analysis between the MPC algorithm and the DFBC algorithm, finding that MPC provides substantial advantages in tracking dynamically infeasible trajectories. Experimental results further underscore the essential role of inner-loop controllers and aerodynamic models in achieving precise quadcopter flight control [11].

In practical applications, quadrotor UAVs often experience delays in control inputs due to external collisions and wireless data transmission, which disrupt their flight dynamics and degrade overall control performance [12], [13].

To address the challenges faced by quadcopter UAVs in flight control, this thesis proposes an innovative control algorithm and system design. The design aims to enhance the safety, stability, and efficiency of the UAV, particularly in adapting to complex environments. By optimizing the control strategy, it is expected to reduce the impact of external disturbances on flight, thereby achieving more reliable flight performance.

The main contributions of this article are as follows:

(1) To enhance interference resistance and ensure angular stability during collisions, a disturbance suppression mechanism is integrated into the Model Predictive Control (MPC) algorithm to address collision-related challenges. This mechanism effectively incorporates the rapid variations in the airframe's angular velocity into the algorithmic model.

(2) The MPC algorithm integrates an input delay compensation mechanism to mitigate signal loss and transmission delays in wireless communication, thereby enhancing system stability under diverse conditions.

(3) Through abundant simulation experiments, the study demonstrates that the optimized MPC algorithm exhibits excellent control performance in terms of attitude angle control, disturbance resistance, and control input timeliness. To further verify the effectiveness and feasibility of the algorithm, this study introduces a spherical quadrotor UAV. With its innovative design, this UAV supports a wide range of motion modes. The advanced MPC algorithm's reproducibility and reliability were validated on the experimental platform.

II. METHODOLOGY

A. Notation

The quadcopter model is illustrated in Fig. 1. The quadcopter system has two coordinate systems. Coordinate system 1 is the earth coordinate system with the origin at O_e , while coordinate system 2 is the airframe coordinate system with the origin at O_b . ω_i represents the propeller rotational speed of each motor. The arrows in the figure indicate the direction of the combined lift of the quadrotor propellers and the direction of the system gravity, respectively.

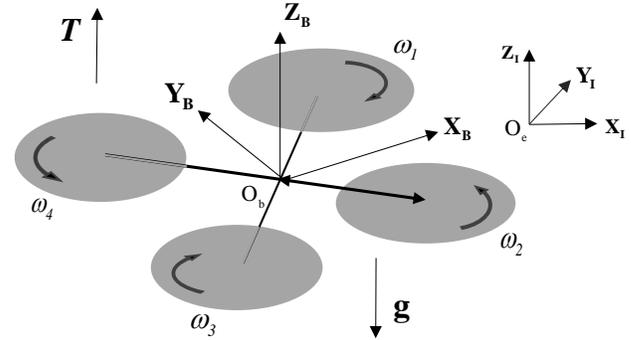


Fig. 1. Coordinate definitions and propeller numbering convention.

The three coordinate axes of the body frame $\mathbf{B} = (\mathbf{X}_B, \mathbf{Y}_B, \mathbf{Z}_B)$ are denoted as $\mathbf{X}_B, \mathbf{Y}_B, \mathbf{Z}_B$. The three coordinate axes represent the roll (ROL), pitch (PIT), and yaw (YAW) axes of the quadrotor system. $\mathbf{I} = (\mathbf{X}_I, \mathbf{Y}_I, \mathbf{Z}_I)$ represents the three-axis directions in the world coordinate system. $\boldsymbol{\alpha} = [\varphi, \theta, \psi]^T$ denotes the three-axis attitude angle.

B. Quadcopter dynamics model

A dynamic model for the quadrotor UAV is developed under conditions of sudden disturbances and air perturbations, drawing on experimental findings from [14], [15] regarding modeling techniques and key experimental parameter values. The following assumptions are made:

- The structural design of the UAV system is symmetric.
- The UAV's center of mass coincides with point O_b .
- The quadrotor propellers and UAV are assumed rigid.

The bolded letters represent the combined effect on the three axes of the quadrotor. The dynamic equations for the quadrotor UAV are derived using the Newton-Euler method, as outlined below:

$$\begin{cases} \dot{\mathbf{P}} = \mathbf{v} \\ m\dot{\mathbf{v}} = \mathbf{T} + \mathbf{F}_f - \mathbf{F}_g \\ \mathbf{J}\dot{\boldsymbol{\Omega}}_B = -\boldsymbol{\Omega}_B \wedge \mathbf{J}\boldsymbol{\Omega}_B + \boldsymbol{\Gamma}_f - \boldsymbol{\Gamma}_g \end{cases} \quad (1)$$

Where $\mathbf{P}, \mathbf{v}, m$ represent the position, velocity, and mass of the quadrotor system, respectively. $\boldsymbol{\Omega}_B = [\boldsymbol{\Omega}_{\text{rol}}, \boldsymbol{\Omega}_{\text{pit}}, \boldsymbol{\Omega}_{\text{yaw}}]^T$.

Ω_B represents the angular velocity of the three axes of the quadrotor UAV. $\mathbf{J} \in \mathbf{R}^{3 \times 3}$ is the body inertia matrix in the body coordinate system, $\mathbf{B} = (\mathbf{X}_B, \mathbf{Y}_B, \mathbf{Z}_B) \cdot \mathbf{J} = \mathbf{J}_x, \mathbf{J}_y, \mathbf{J}_z$ denotes the moments of inertia of the UAV body around the three axes of the coordinate system. \mathbf{F}_g represents the effect of gravity on the quadrotor system, which only affects the vertical direction.

$$\mathbf{F}_f = \begin{bmatrix} \cos \varphi \cos \psi \sin \theta + \sin \varphi \sin \psi \\ \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ \cos \varphi \cos \theta \end{bmatrix} \sum_{i=1}^4 F_i \quad (2)$$

$$F_i = C_T \omega_i^2 \quad (3)$$

where \mathbf{F}_f denotes the frictional force generated by the interaction between the quadrotor propellers and the air.

$$\mathbf{F}_g = \begin{bmatrix} d(F_3 - F_1) \\ d(F_4 - F_2) \\ C_M (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (4)$$

\mathbf{F}_g represents the torque generated by the quadrotor UAV body. F_i represents the thrust generated by each of the four rotors of the quadrotor UAV. d represents the arm length of the quadrotor.

$$\mathbf{F}_g = \sum_{i=1}^4 \Omega_B \wedge \mathbf{J}_r \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+1} \omega_i \end{bmatrix} \quad (5)$$

\mathbf{F}_g represents the torque generated by the gyroscopic effect of the propellers [16]. \mathbf{J}_r is the rotor inertia parameter [14].

Building on the above theoretical framework, the dynamic equations of the quadrotor UAV can be derived in the world coordinate system as follows:

$$\begin{cases} \ddot{Z} = \frac{1}{m} \{ (\cos \varphi \cos \theta) u_1 \} - g \\ \ddot{\varphi} = \frac{1}{J_x} \{ \dot{\theta} \dot{\psi} (J_y - J_z) - J_r \Omega_{tot} \dot{\theta} + du_2 \} \\ \ddot{\theta} = \frac{1}{J_y} \{ \dot{\varphi} \dot{\psi} (J_z - J_x) + J_r \Omega_{pit} \dot{\varphi} + du_3 \} \\ \ddot{\psi} = \frac{1}{J_z} \{ \dot{\theta} \dot{\varphi} (J_x - J_y) + C_M u_4 \} \end{cases} \quad (6)$$

u_1, u_2, u_3 and u_4 are the four inputs of the quadrotor UAV system. The control inputs satisfy the following equation:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{pmatrix} C_T & C_T & C_T & C_T \\ -C_T & 0 & C_T & 0 \\ 0 & -C_T & 0 & C_T \\ C_M & -C_M & C_M & -C_M \end{pmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (7)$$

C_T represents the thrust coefficient of the motor propeller, and C_M represents the torque coefficient. The appropriate values can be measured by the propeller pulling force coefficient and moment coefficient determination experiments [17], [18].

C. MPC algorithm optimization

This study optimizes the MPC algorithm by incorporating a disturbance suppression mechanism. In the model analysis process, the Z-axis velocity and the system's body angular velocity are processed through a threshold detector to generate a disturbance matrix.

The flowchart of the MPC algorithm process incorporating disturbance terms is shown as follows:

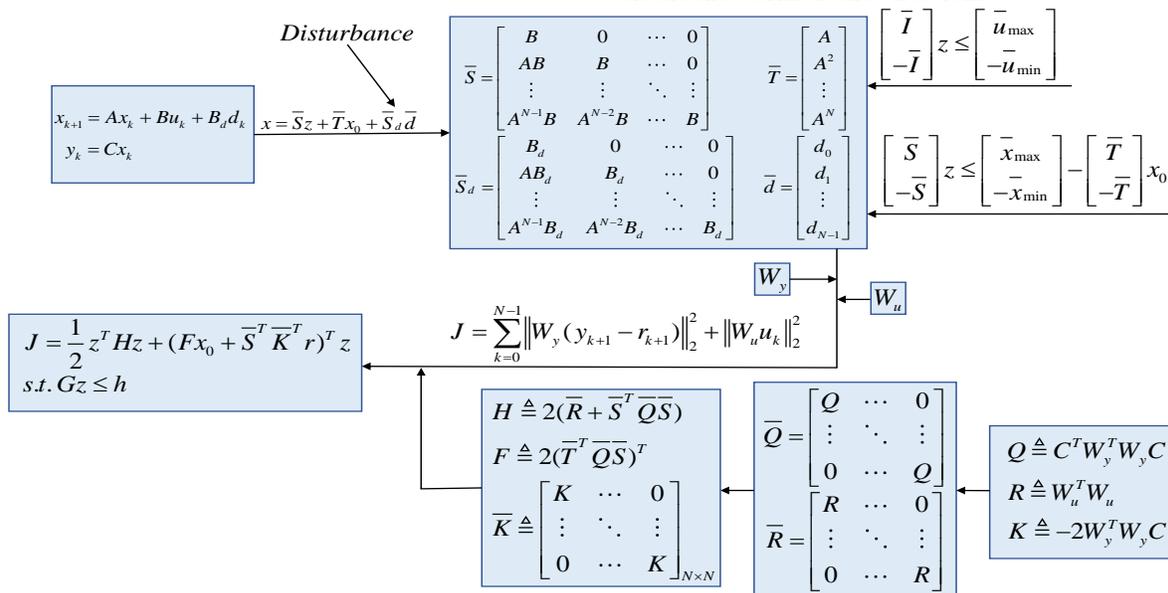


Fig. 2. The MPC algorithm block diagram

Unlike expected control inputs, disturbances typically affect the state variables. Current states are acquired through sensors, and state changes are processed using a differential detector. When the change exceeds a specified threshold, the angular velocity variation is introduced into the disturbance matrix, thereby enhancing the modeling of the optimization algorithm. The optimization of the disturbance algorithm can be found in lines 5 through 16 of **Algorithm 2**.

Differing from traditional controller output constraints, MPC optimization integrates constraints within the controller [19], thus achieving magnitude limitations internally. The quadratic programming problem is solved to yield the optimal solution z^* , with the first set of predicted inputs chosen as the system input, i.e., $u_0 = z_0^*$. The system modeling of the MPC algorithm is shown in **Algorithm 1**.

D. Input delay compensation

In this study, an optimization for handling system input delays in the MPC algorithm is also conducted. Assuming the system delay is denoted by τ , typically τ is a multiple of the control cycle. In experiments, τ is usually set to one or more control cycles. When input control loss or delayed input occurs, the system state equation is optimized as follows:

$$x_{k+1} = Ax_k + Bu_{k-\tau} \quad (8)$$

$$y_k = Cx_k \quad (9)$$

Given the equation $\bar{x}_k \triangleq x_{k+r}$, the selection of future states and current inputs leads to the following standard state-space equations:

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k \quad (10)$$

$$\bar{y}_k = C\bar{x}_k \quad (11)$$

Obtain the current optimal input u_k^* , using u_{k-r}^* as the system input. The algorithm optimization for handling input delays or losses has been completed. The implementation for handling input delays or losses can be found in lines 7 through 25 of **Algorithm 1**.

MPC Modeling and Delay Optimization Implementation:

Algorithm 1 : MPC & LATENCY OPTIMIZATION

```

1  z = mpc(A, B, C, Dd, Wq, Wr, N, x, rbar, ubar, ur, xr)
2  n = size(A), p = size(B), m = size(C)
3  pd = size(Dd), tau = size(ubar), Dvalue ← 1e - 1
4  Q = C' * (Wq' * Wq) * C
5  R = Wr' * Wr
6  K = -2 * (Wq' * Wq) * C
7  For i = 1 to N do
8      For j = 1 to N do
9          if i >= j do
10             Sbar = functionSbar(Ai-j * B)
11             Sdbar = functionSdbar(Ai-j * Dd)

```

```

12         else do
13             Sbar = Zeros(n, p)
14             Sdbar = Zeros(n, pd)
15         end
16     For i = 1 to N do
17         Qbar ← Q
18         Tbar ← Ai
19         Rbar ← R
20         Kbar ← K
21         dbar ← Dvalue
22     end
23     For index = 1 to tau do
24         x = A * x + B * ubar(:, index)
25     end
26     H = 2 * (Rbar + Sbar' * Qbar * Sbar)
27     F = 2 * (Tbar' * Qbar * Sbar)'
28     Fd = 2 * Sbar' * Qbar' * Sdbar
29     f = F * x + Sbar' * Kbar' * rbar + Fd * dbar
30     z = functionProg(H, f, ~)
31     return z

```

Disturbance Optimization Algorithm Implementation:

Algorithm 2 : DISTURBANCE OPTIMIZATION

```

1  Jx Jy Jz Wq Wr M N ur xr m dt Th ← Value
2  While True:
3      For iteration = 3 to M do
4          x, r ← Read(SerialObj)
5          A, B, C ← Matrixinit(x, dt)
6          if threshold comparator(x, Th) do
7              Dd ← Matrixdis(x, dt)
8              Dvalue ← 1e - 1
9              Dd ← Dd * Dvalue
10         else do
11             Dd ← Zerosdis()
12             Dvalue ← 0
13             Dd ← Dd * Dvalue
14             rbar = reshape(r, [N * m, 1])
15             ubar = u(:, iteration - τ: iteration - 1)
16             z = mpc(A, B, C, Dd, Wq, Wr, N, x, rbar, ubar, ur, xr)
17             u(:, iteration) = z(1:p)
18             Write(SerialObj, u(:, iteration))
19         end

```

The variable Th in line 6 of **Algorithm 2** represents the disturbance threshold. When the state change exceeds this threshold, the state variation is incorporated into the system model.

E. Compensating for supply voltage fluctuations

The relationship between the duty cycle and motor speed is influenced by battery voltage (V) [20]. The impact of voltage fluctuations on motor speed is mitigated by compensating for the PWM duty cycle of the motor. The specific implementation of this approach is discussed in detail below.

This study employs multivariate linear regression to model the relationship between duty cycle, voltage, and speed, thereby compensating for the discrepancies between the actual motor control inputs and the algorithm's predicted outputs in the

quadrotor system. Rotor experiments are conducted to precisely measure and record the variations in voltage, rotational speed, and duty cycle, thoroughly analyzing the interdependencies among these variables during motor operation.

Based on the ample experimental data, the table of estimated coefficients is presented as follows:

TABLE I
COEFFICIENTS OF MULTIPLE LINEAR REGRESSION VARIABLES

Variable	Estimate
Intercept	1065.1
ω_i (rad/s)	0.098375
Voltage(V)	-41.317

The mathematical relationship is as follows:

$$\eta = 1065.1 + 0.098375 * \omega_i - 41.317 * V \quad (12)$$

η represents the predicted duty cycle command, ω_i denotes motor speed, and V denotes the battery voltage.

The predicted duty cycle command, derived from the output motor speed command and the current voltage reading, is then applied to the system motors.

F. MPC-PID Controller

As shown in Fig. 3, the matrix T_m is the inverse matrix used to convert MPC algorithm torque outputs into desired body angular velocity outputs [21].

The converted desired body angular velocity, combined with state measurements from the IMU of the quadcopter UAV, forms a PID control loop to regulate the operation of the quadcopter system. The matrix T_m is:

$$T_m = \begin{bmatrix} f(u_1, u_1^\dagger) & 0 & 0 & 0 \\ 0 & f(u_2, u_2^\dagger) & 0 & 0 \\ 0 & 0 & f(u_3, u_3^\dagger) & 0 \\ 0 & 0 & 0 & f(u_4, u_4^\dagger) \end{bmatrix} \quad (13)$$

Where $f(u_i, u_i^\dagger)$ represents the mathematical relationship between control inputs u_1, u_2, u_3, u_4 and state variables $\dot{x}_5, \dot{x}_6, \dot{x}_7, \dot{x}_8$.

The performance metrics include the control time required for the system to reach the target attitude angles and the fluctuations in state data after stabilization [22].

The quadcopter's mathematical model is transformed into a state-space representation.

$$\begin{cases} \dot{x}_1 = x_5 \\ \dot{x}_2 = x_6 + x_7 \sin x_2 \tan x_3 + x_8 \cos x_2 \tan x_3 \\ \dot{x}_3 = x_7 \cos x_2 - x_8 \sin x_2 \\ \dot{x}_4 = \frac{\sin x_2 x_7 + \cos x_2 x_8}{\cos x_3} \\ \dot{x}_5 = \frac{u_1}{m} \cos x_3 \cos x_2 - g \\ \dot{x}_6 = \frac{u_2 - (J_y - J_z)x_7 x_8}{J_x} \\ \dot{x}_7 = \frac{u_3 - (J_z - J_x)x_6 x_8}{J_y} \\ \dot{x}_8 = \frac{u_4 - (J_x - J_y)x_6 x_7}{J_z} \end{cases} \quad (14)$$

Quadcopter UAV state-space representation:

$$\dot{X} = \hat{f}(X, U) \quad (15)$$

At the equilibrium point where $\dot{X} = 0$, we obtain:

$$\hat{f}(\bar{X}, \bar{U}) = 0 \quad (16)$$

$$A = \left. \frac{\partial \hat{f}(\bar{X}, \bar{U})}{\partial X} \right|_{X=\bar{X}} \quad (17)$$

$$B = \left. \frac{\partial \hat{f}(\bar{X}, \bar{U})}{\partial U} \right|_{X=\bar{X}} \quad (18)$$

$$C = [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \quad (19)$$

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8] \quad (20)$$

$$X = [z \ \phi \ \theta \ \psi \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}] \quad (21)$$

Matrices A, B, and C are used to solve for the optimal control input solution. The state variables represent system altitude, quadcopter attitude angles in three axes, altitude rate, and three-axis body angular velocity. The constraints consist of state constraints and system input constraints.

The system control block diagram is shown in Fig.3. The FOC block corresponds to the motor drive section.

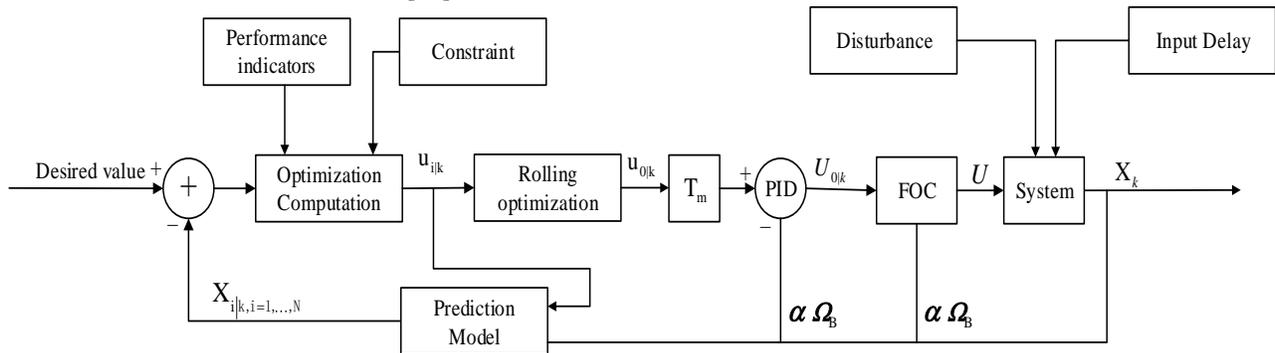


Fig. 3. System Control Block Diagram

III. IMPLEMENTATION DETAILS

Before delving into detailed simulations and practical experiments, we first outline the implementation details and experimental setup. The MPC flight controller is implemented in C++ and runs on a PC, which also handles data logging and network communication. Torque control commands are wirelessly transmitted to an embedded STM32F4 MCU via a wireless serial port interface. The STM32F4 platform hosts the robotic system, realizing the design of a spherical quadcopter robot. The MPC runs at a frequency limited to 100 Hz on the PC. The PID inner loop algorithm (200 Hz) and IMU measurements (400 Hz) operate on the STM32F4 platform. To ensure sensor data suitability for algorithm convergence, sensor outputs are filtered through a low-pass filter with a 10ms time constant before PID algorithm computation. In simulation experiments, a motor dynamics model is established. In real-world experiments, MCU acquires data from MT6701 magnetic encoders and IMUs, processes them, and sends them to the PC for MPC algorithm quadcopter system modeling. Fig. 4 displays the MT6701 module.



Fig. 4. Magnetic encoder module

The parameter settings and experimental conditions in the simulation experiments are consistent with those of real-world experiments.

IV. SIMULATION EXPERIMENTS

This study compares the PID algorithm with the optimized MPC-PID algorithm through simulation experiments. The PID algorithm implemented is a dual-loop series controller, while the MPC-PID algorithm includes an inner loop PID controller.

The experiments analyze the behavior of these algorithms in tracking angular expectations under varying experimental conditions, examining their effectiveness in suppressing perturbations, and observing the system response when there is a delay in the control input quantity.

Prior to starting the simulation experiments, we specify the parameters utilized in the robot experiments as listed in TABLE II and TABLE III.

TABLE II
QUADROTOR CONFIGURATIONS

Parameter(s)	Unit(s)	Value(s)
m	[kg]	1.277
d	[m]	0.110
C_T	[kg · m/rad ²]	1.5784e-6
C_M	[kg · m ² /rad ²]	1.4280e-8
(u_{min}, u_{max})	[N · m]	(0,9.5)
J_x	[kg · m ²]	8.810e-2
J_y	[kg · m ²]	2.203e-2
J_z	[kg · m ²]	4.227e-2

The PID algorithm tuning parameters are first reduced by a factor of 1000 in the software to simplify adjustment, and then increased by 1000 for practical tuning [13].

TABLE III
CONTROLLER GAINS AND PARAMETERS

MPC-PID		PID
W_q	diag(90,2,2,2)	Attitude Loop (PID)
W_r	diag(5,0.5,0.5,0.5)	K_{rol} (3500,5,6000)
dt	20 ms	K_{pit} (1500,2,100)
N	20	K_{yaw} (3000,1,100)
K_{rol}	(2500, 1, 4000)	Angular Velocity Loop (PID)
K_{pit}	(2000, 5, 3500)	K_{rol} (3000,1, 4000)
K_{yaw}	(3500, 3, 100)	K_{pit} (2000,10,350)
		K_{yaw} (3500,5, 150)

W_q is the error weighting matrix, and W_r is the input weighting matrix. The table displays the necessary configuration parameters for the simulation experiments.

A. State convergence simulation

As illustrated in Fig. 5(a), when the desired system angle was set to 30°, the MPC-PID algorithm achieved the desired angle more rapidly and with smoother performance than the PID algorithm. As shown in Fig. 5(b), the angular velocity exhibits a more gradual change under the optimized MPC, without overshoot or repeated regulation seen in the PID algorithm. The spherical quadrotor system, integrated with the MPC-PID algorithm, improves the accuracy of angular state control.

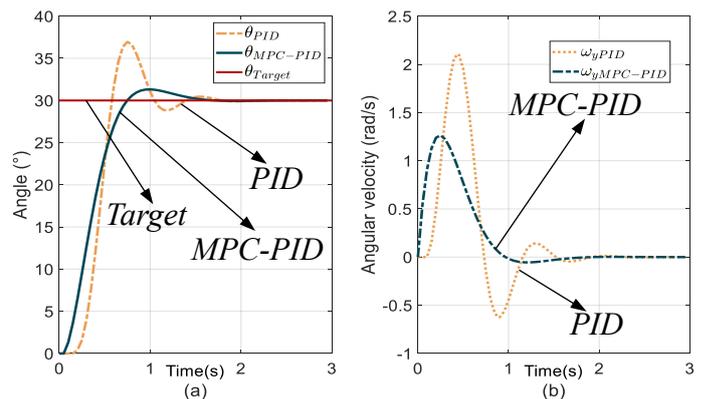


Fig. 5. Comparison of attitude angle variations

B. Disturbance simulation

In the simulation experiment, a consistent desired angle is set for the system. Program instructions are then used to apply a perturbation at a specific time to alter the angular state and simulate the effect of the perturbation [23].

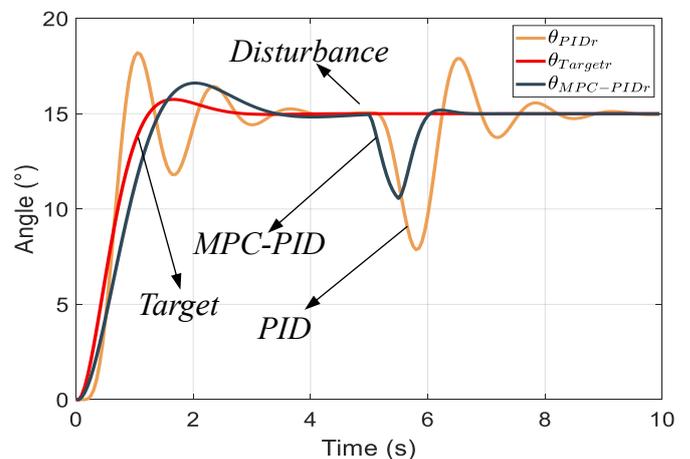


Fig. 6. Disturbance simulation

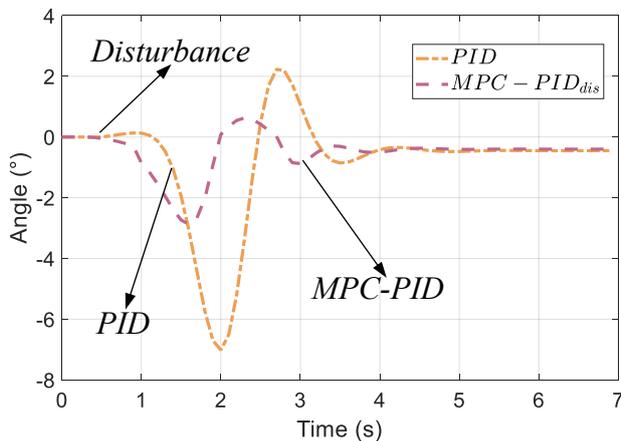


Fig. 7. Disturbance details

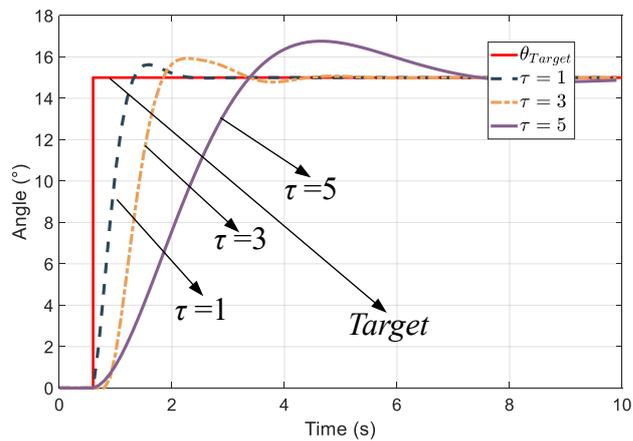


Fig. 9. MPC-PID Input delay state variation

The accurate system modeling emphasized in this study helps to reduce the disturbance-induced state fluctuations and enhance the overall robustness of the system.

As shown in Fig. 6, the MPC-PID algorithm exhibits excellent performance in achieving the target attitude angle, ensuring a more streamlined process and minimizing system state fluctuations. In comparison to the PID algorithm, which has limited disturbance rejection and is relatively fragile, the MPC-PID system is more resilient and experiences fewer state changes. Additionally, as shown in Fig. 7, the algorithm improves its efficiency when dealing with disturbances by optimizing the disturbance suppression mechanism, highlighting its superior anti-disturbance capability.

C. Input delay compensation simulation

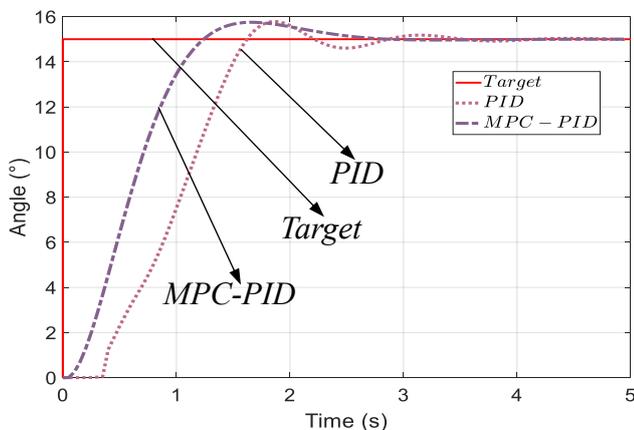


Fig. 8. Comparison of State Dynamics

Under the PID algorithm, as demonstrated in Fig. 8, the state remains unchanged initially due to the delayed application of control inputs. In contrast, the MPC-PID algorithm incorporates an input delay handling mechanism that effectively controls the state change during these initial moments. It is evident that the presence of input delays in the PID algorithm causes a less seamless transition of changing the angular state to the desired angle in comparison to the MPC-PID algorithm.

Fig. 9 depicts the performance of the advanced MPC-PID algorithm under various input delays, denoted as τ . The system state consistently reaches the desired angle across different delays, demonstrating stable convergence. Notably, the algorithm effectively restricts the overshoot to within 1 degree upon reaching the target state, indicative of its strong control performance.

V. REAL-WORLD EXPERIMENTS

In this study, angle tracking, interference mutation, and input delay experiments were conducted on the spherical quadcopter UAV using two different algorithms: advanced MPC-PID and PID. External disturbances were applied to the spherical structure of the quadrotor system to observe internal angular state changes. Additionally, we observed the overall motion control of the spherical robot at a macroscopic level. Experiments were conducted in various ground environments such as grass, concrete, and wood surfaces to observe the overall motion control of the spherical robot.

In the structural design, carbon fiber materials are chosen to reduce system weight and enhance structural rigidity [12]. This design enhances the ground collision and cushioning performance of the system. Fig. 10 is the structural simulation and design diagram of the spherical system. The spherical quadcopter structure is shown in Fig. 11. The dimensions of the spherical quadrotor system are expressed in meters. m denotes the mass of the spherical quadrotor system.

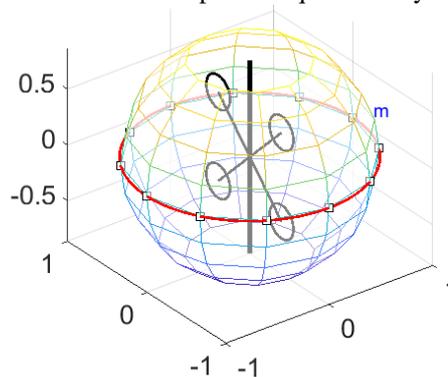


Fig. 10. Structural simulation design diagram

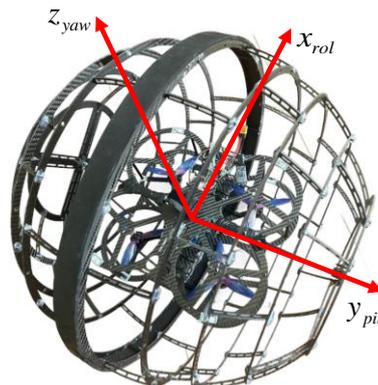


Fig. 11. The spherical quadcopter structure

A. State convergence experiment

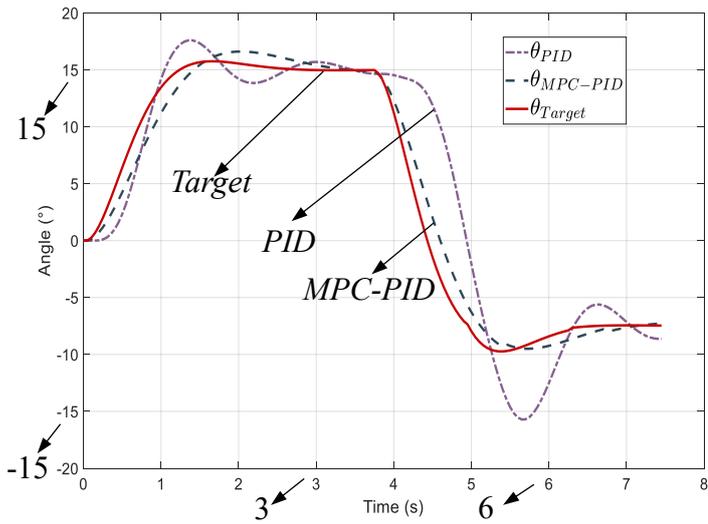


Fig. 12. Angle variation curves in real-world experiments

The angular tracking performance of the PID algorithm is illustrated by curve θ_{PID} in Fig.12, while curve $\theta_{MPC-PID}$ represents the angular tracking performance of the MPC-PID algorithm. Curve θ_{Target} represents the desired attitude angle change. The angular tracking performance of the MPC-PID algorithm outperforms that of the PID algorithm, indicating a higher tracking accuracy for the desired angle. Conversely, the PID algorithm, lacking compensation for the inherent nonlinear effects of the spherical structure, exhibits angle tracking deviations during ground motion. This suboptimal performance is evident in the substantial tracking errors at the curve's endpoint, in contrast to the improved performance of the optimized MPC-PID algorithm.

B. Disturbance experiment

The spherical robot's performance is evaluated through perturbation experiments, comparing the system's attitude angle response under identical perturbations using both the PID and optimized MPC-PID algorithms.

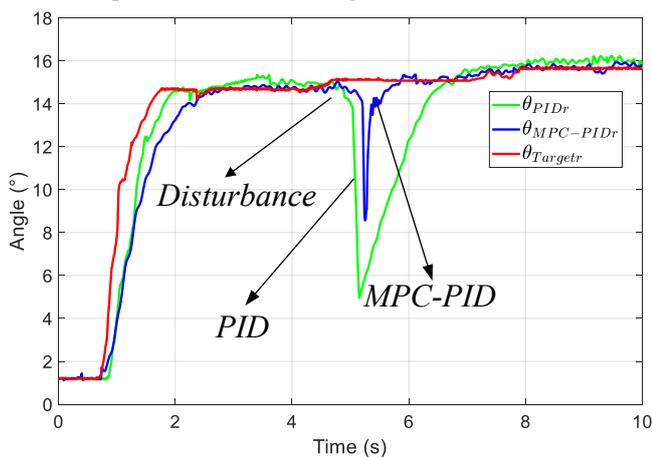


Fig. 13. Comparison of Disturbance Suppression Effects

As shown in Fig. 13, the advanced MPC-PID demonstrates greater effectiveness in suppressing disturbances affecting the system's state, with enhanced disturbance rejection capabilities. In contrast, the traditional PID algorithm does not consider the inherent nonlinear characteristics of the spherical quadrotor, leading to lower robustness and a decreased ability to recover after disturbances.

Moreover, the optimized MPC-PID algorithm exhibits faster recovery to the desired angle, underscoring its superior disturbance suppression mechanisms and enhanced system resilience.

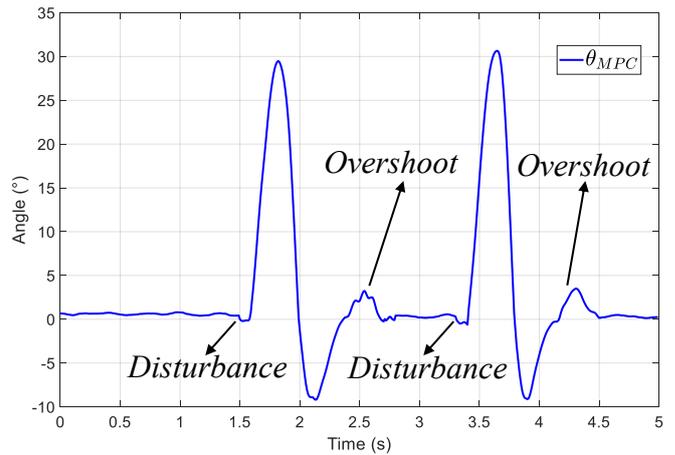


Fig. 14. State Variation Under Large Disturbances

To further assess the efficacy of the MPC-PID algorithm, we intentionally increased the disturbance level to observe its influence on the system state. As shown in Fig.14, when external disturbances induce significant changes in state (30°), the MPC-PID algorithm remains convergent, demonstrating its robustness against disturbance variations. The results indicate that overshooting occurs after convergence due to excessive disturbances, suggesting further optimization of the parameter matrix to mitigate overshoot issues.

These disturbances are mathematically modeled as pulse variations through matrix operations, with the disturbance suppression mechanisms providing a damping effect on these pulses, thereby enhancing system stability.

C. Control input delay compensation experiment

In practical experiments, we programmed the delay to be fixed at τ control cycles, where τ is typically an integer multiple of the control period. This study compares the system state performance of PID and the optimized MPC-PID algorithms in real-world experiments.

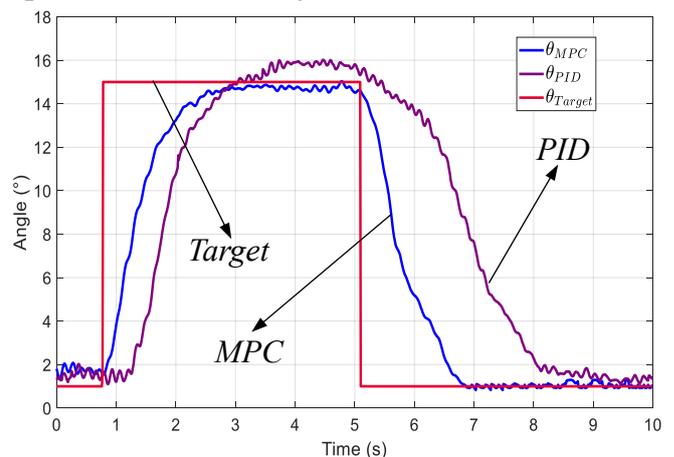


Fig. 15. Comparison of the Effects of Different Algorithms

The optimized MPC-PID algorithm with a delay handling mechanism, as illustrated in Fig.15, exhibits superior tracking accuracy and timeliness in the presence of control lags. In contrast, the PID algorithm, which disregards control delays, demonstrates poor tracking performance concerning input delays in the control system. A notable reduction of 1.5 seconds in the time required to revert to the initial angle is

observed with the optimized MPC-PID algorithm compared to the conventional PID algorithm.

The tracking lag of the PID algorithm becomes more prominent in the latter part of the curve, resulting in a significant decrease in tracking effectiveness.

Furthermore, experiments examined the impact of varying delays on the system state using the MPC-PID algorithm, showing the fluctuations in state through resultant curves.

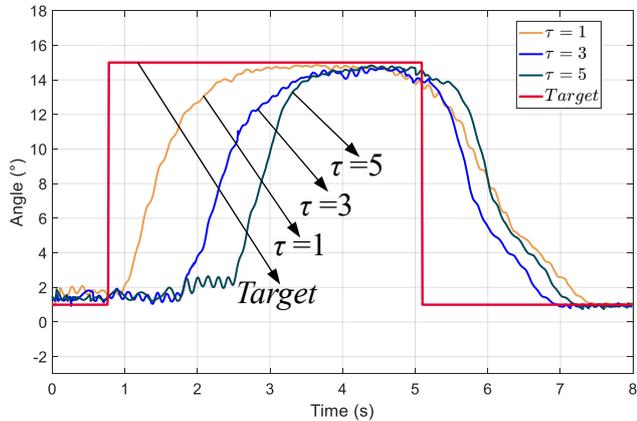


Fig. 16. Comparison of Different Delay Effects

Fig. 16 shows that the system has good convergence of the angular states under different delay conditions. Initially, the delays cause a lag in the state change during control.

However, the optimization of the input delay handling mechanism allows the system to return to the initial position quickly under different delay conditions. Meanwhile, further adjustment of parameters to improve the control effect of the algorithm can optimize the state change lag that exists in the pre-control period.

D. Multi-scenario motion analysis

To validate the stability and robustness of the spherical robot designed in this study across multiple motion scenarios, the robot was placed in three distinct environments. We applied specific angle targets and observed the robot's ability to track these targets and its variations in motion state.

Experimental data are illustrated in Fig. 17.

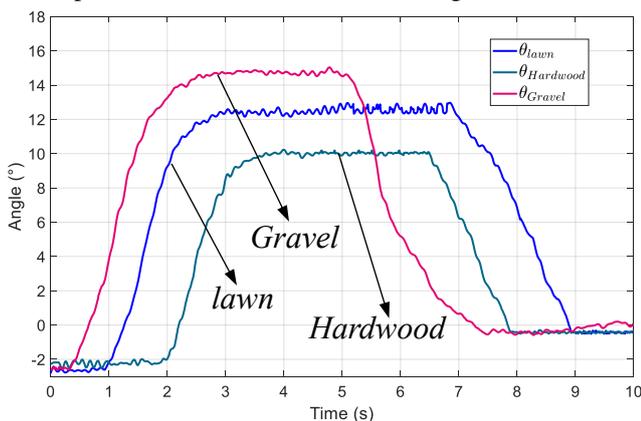


Fig. 17. Angle Variations in Diverse Motion Scenarios

In different ground conditions, the system exhibits varying state changes: the desired angles are 12° on the lawn, 10° on hardwood flooring, and 15° on gravel. As shown in Fig. 17, the curves indicate that the system maintains good attitude angle tracking performance in different scenarios. In addition, the system exhibits good stability for state changes under various environmental motions, which highlights the ability

of the spherical quadrotor system to operate effectively in different motion scenarios.

E. ARPI Performance Comparison

The Angle Response Performance Index (ARPI) is a vital metric for assessing the performance of quadrotor UAVs. It quantitatively evaluates the UAV's overall capability by analyzing the speed and precision of its angular adjustments in response to external commands.

To evaluate the performance of the advanced MPC-PID algorithm against the conventional PID algorithm, approximately 50 experiments were conducted. The ARPI values were calculated and recorded for each trial, followed by a detailed comparative analysis of the results.

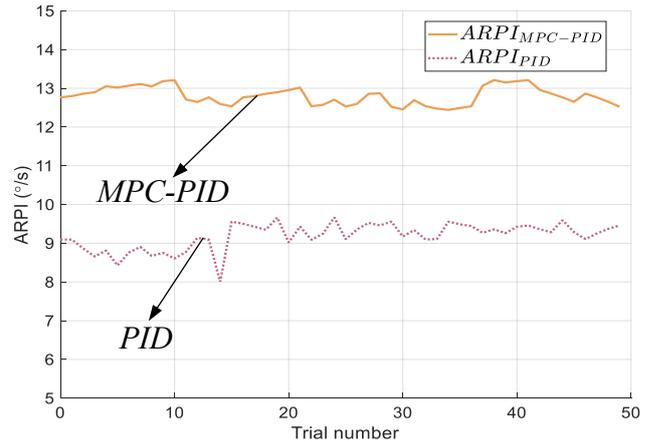


Fig. 18. ARPI Comparison Between Algorithms

As shown in Fig.18, the ARPI values obtained from experiments on the spherical quadrotor UAV using the advanced MPC-PID algorithm are significantly higher, consistently averaging approximately 13. In contrast, when the conventional PID algorithm is used, the ARPI values are around 9.

These findings demonstrate that the advanced MPC-PID algorithm significantly enhances the UAV's dynamic responsiveness and control command sensitivity, thereby improving overall flight stability.

F. Analysis of Model Uncertainty

In practical experiments, we systematically varied model parameters under different conditions and recorded the attitude angle variation data of the quadrotor system [11]. The resulting table was derived from data computation.

CONDITIONS	Angle RMSE [deg] (mean ± SD)	
	MPC-PID	PID
BASELINE	1.01 ± 0.58	2.75 ± 1.02
+50% Drag	2.61 ± 1.24	3.94 ± 1.69
+100% Drag	4.87 ± 3.56	6.11 ± 2.32
-30% m	0.81 ± 0.61	1.76 ± 1.10
+30% m	1.49 ± 0.64	3.57 ± 1.12
10ms Latency	2.15 ± 0.26	3.09 ± 0.11
30ms Latency	2.89 ± 0.16	3.98 ± 0.89
50ms Latency	3.09 ± 0.69	4.87 ± 1.66

Each number represents the mean and standard deviation (SD) across 20 angle tracking experiments. The bold values show that their values are smaller than their counterparts in the same row.

As illustrated in Table IV, introducing drag and control delays increases the instability of the system's attitude-angle state.

The state stability of a quadrotor system with an advanced controller is crucial, as emphasized in article [24].

However, the optimized MPC-PID algorithm outperforms the PID algorithm under different experimental conditions, which can be attributed to the accurate modeling as well as the optimization of the algorithm. Furthermore, reducing system mass improves attitude angle tracking performance and stability, as a lower mass enhances coordination during system motion.

VI. DISCUSSION

The spherical quadrotor robot has a robust and tamper-resistant mechanical structure with good ground motion.

Compared with the traditional PID method, the optimized MPC-PID algorithm solves the discrepancy between the simulation of the spherical quadrotor system and the actual experiments, and ensures the convergence of the attitude angle control in the actual motion scenario.

Moreover, MPC-PID introduces mechanisms for rejecting disturbances, allowing rapid adjustment of system states during significant disruptions. By leveraging precise quadrotor dynamics modeling, the algorithm enables swift and accurate state adjustments even in the presence of general disturbances, surpassing PID in responsiveness and precision.

The delayed handling mechanism of MPC-PID resolves latency issues caused by spatial delays or data loss during control data transmission. Experimental findings illustrate improved convergence of system states with integrated delay handling, demonstrating superior performance over the PID algorithm. Furthermore, experimental data across multiple loss scenarios indicate consistent robust convergence of system states.

The spherical quadrotor robot exhibits enhanced state regulation and multi-scene locomotion capabilities when utilizing the optimized MPC-PID algorithm, which demonstrates superior performance compared to traditional PID algorithm.

VII. CONCLUSION

This study emphasizes the superior attitude-angle control capability, disturbance suppression capability, resistance to control hysteresis, and multi-scenario motion capability of the spherical quadrotor robot through a systematic comparison between the optimized MPC-PID algorithm and the traditional PID algorithm. Simulation and practical experimental data confirm that the optimized MPC-PID algorithm demonstrates higher control accuracy compared to the traditional PID algorithm. Moreover, the spherical quadrotor robot equipped with the advanced MPC-PID algorithm showcases robust motion stability in various scenarios.

REFERENCES

- [1] Evandro B, Frederic B, and Stephane V, "Modelling, control and simulation of a single rotor UAV with swashplateless torque modulation," *Aerospace Science and Technology*, vol.140, pp108433, 2023.
- [2] O. C. Carholt, E. Fresk, G. Andrikopoulos and G. Nikolakopoulos, "Design, modelling and control of a Single Rotor UAV," 2016 24th Mediterranean Conference on Control and Automation (MED), 21-24 June, 2016, Athens, Greece, pp840-845.
- [3] Elena S, Brian D, Vikram H, and Inderjit C, "All-Terrain Cyclocopter Capable of Aerial, Terrestrial, and Aquatic Modes," *JOURNAL OF THE AMERICAN HELICOPTER SOCIETY*, vol.66, 2021.
- [4] Moad I, Mohammad S, and Fawaz A, "A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms," *Journal of Intelligent Robotic Systems*, vol.104, 2022.
- [5] González Á M, Bregua S P and Pierce J G, "Unmanned Aerial Vehicles(UAVs) in Marine Mammal Research: A Review of Current Applications and Challenges," *Drones*, vol.7, pp667, 2023.
- [6] Ivan L, and Javier M, "PID control of quadrotor UAVs: A survey," *Annual Reviews in Control*, vol.56, pp100900, 2023.
- [7] Lotufo A M, Colangelo L, P-Montenegro C, Canuto E, and Novara C, "UAV quadrotor attitude control: An ADRC-EMC combined approach," *Control Engineering Practice*, vol.84, pp13-22, 2019.
- [8] Ahmad F, Kumar P, Bhandari A, and P. Patil P, "Simulation of the Quadcopter Dynamics with LQR based Control," *Materials Today: Proceedings*, vol.24, pp326-332, 2020.
- [9] Renbo Qing, Xiaoming Tang, Hua Huang and Yongzhen Cao, "An MPC Method for Trajectory Tracking of Unmanned Vehicle with LMI-Constrained Unscented Kalman Filter," *IAENG International Journal of Computer Science*, vol. 51, no. 4, pp447-462, 2024.
- [10] D. Hanover, P. Foehn, S. Sun, E. Kaufmann and D. Scaramuzza, "Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp690-697, 2022.
- [11] S. Sun, A. Romero, P. Foehn, E. Kaufmann and D. Scaramuzza, "A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp3357-3373, 2022.
- [12] Goh, Guo Dong, William Toh, Yee Ling Yap, T. Y. Ng and Wai Yee Yeong, "Additively manufactured continuous carbon fiber-reinforced thermoplastic for topology optimized unmanned aerial vehicle structures" *Composites Part B: Engineering*, vol.216, pp108840, 2021.
- [13] Dong, and Jiaying, "Performance comparison and analysis of traditional PID and fuzzy PID control applied to UAV," *Journal of Physics: Conference Series*, vol. 2649, pp012001, November 2023.
- [14] J. Ye, J. Wang, T. Song, Z. Wu and P. Tang, "Nonlinear Modeling the Quadcopter Considering the Aerodynamic Interaction," *IEEE Access*, vol. 9, pp134716-134732, 2021.
- [15] Bristeau, P. Martin, E. Salaün and N. Petit, "The role of propeller aerodynamics in the model of a quadrotor UAV," 2009 European Control Conference (ECC), 23-26 August, 2009, Budapest, Hungary, pp683-688.
- [16] Raffo, Guilherme V, Manuel G. Ortega and F. R. Rubio, "An integral predictive / nonlinear H^∞ control structure for a quadrotorhelicopter," *Automatica*, vol. 46, pp29-39, 2010.
- [17] D. Six, S. Briot, J. Erskine and A. Chriette, "Identification of the Propeller Coefficients and Dynamic Parameters of a Hovering Quadrotor From Flight Data," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp1063-1070, 2020.
- [18] G. Andria and A. Di Nisio et al., "Design and Performance Evaluation of Drone Propellers," 2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), 20-22 June, 2018, Rome, Italy, pp407-412.
- [19] G. Torrente, E. Kaufmann, P. Föhn and D. Scaramuzza, "Data-Driven MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp3769-3776, 2021.
- [20] Liang Yan-hu, "Modeling and testing of power system based on voltage compensation for quadrotors," *Systems engineering and electronics*, vol. 36, pp934-939, 2014.
- [21] N. Seyed Gogani, V. Behnamgol, S. M. Hakimi and G. Derakhshan, "Finite Time Back Stepping Supper Twisting Controller Design for a Quadrotor," *Engineering Letters*, vol. 30, no. 2, pp674-680, 2022.
- [22] Mete Özbaltan, and Serkan Çaşka, "Altitude control of quadcopter with symbolic limited optimal discrete control," *International Journal of Dynamics and Control*, vol. 12, no. 5, pp1533-1540, 2024.
- [23] Zexin Wang, Jiang Zhao, Zhihao Cai, Yingxun Wang and Ningjun Liu, "Onboard actuator model-based Incremental Nonlinear Dynamic Inversion for quadrotor attitude control: Method and application," *Chinese Journal of Aeronautics*, vol. 34, no. 11, pp216-227, 2021.
- [24] Almido Haryanto Ginting, Oyas Wahyunggoro, and Adha Imam Cahyadi, "Robust Proportional-Derivative Control on SO(3) to Compensate The Unknown Center of Gravity of Quadrotor," *Engineering Letters*, vol. 28, no. 2, pp359-370, 2020.