

# Trajectory Tracking Control of Quadrotor UAV based on Parallel Cooperative PID-MPC Architecture

Yihui Zhang\*

School of Electronic and Electrical Engineering, Cangzhou Jiaotong College, Cangzhou, 061000, China

\*zhangxauusd@outlook.com

---

## Abstract

Quadrotor UAV trajectory tracking control, PIDMPC parallel cooperative architecture, feedforward-feedback fusion mechanism, event-triggered strategy, fuzzy adaptive parameter tuning, MATLAB/Simulink simulation validation. To address the problem that the advantages of Model Predictive Control (MPC) cannot directly assist the Proportional-IntegralDerivative (PID) controller in the traditional cascade PID-MPC control structure, this paper proposes a PID-MPC parallel cooperative control architecture. This architecture employs MPC as the feedforward channel for trajectory prediction and lookahead optimization, and PID as the feedback channel for rapid error correction, synthesizing the outputs of both into the final control input through a dynamic weighted fusion mechanism. To reduce the online computational burden of MPC, an eventtriggered mechanism based on the state error norm is introduced, which activates the MPC solver only when the error exceeds a threshold. Meanwhile, a fuzzy inference system is designed to achieve online adaptive tuning of PID parameters. On the MATLAB/Simulink platform, using a quadrotor UAV nonlinear dynamics model as the plant, the proposed method is compared with three baseline controllers-pure PID, pure MPC, and traditional cascade PID-MPC-under four scenarios: figure-8 complex trajectory tracking, pulse and continuous wind disturbance rejection, model parameter mismatch robustness, and computational efficiency. Experimental results demonstrate that the proposed parallel cooperative architecture achieves a root mean square tracking error (RMSE) of 0.047m, representing reductions of 78.1% compared to pure PID, 52.5% compared to pure MPC, and 41.3% compared to the traditional cascade structure; the recovery time under pulse wind disturbance is only 0.826s, and the steady-state error under continuous wind disturbance is 0.031m; the MPC trigger rate is reduced to 38.7%, with an average single-step solving time of 2.134ms, satisfying real-time requirements.

## Keywords

Quadrotor UAV; Model Predictive Control; PID Control; Parallel Cooperative Architecture; Event-triggered Mechanism; Fuzzy Adaptive; Trajectory Tracking.

---

## 1. Introduction

Quadrotor UAVs have been widely applied in agricultural plant protection, power line inspection, logistics delivery, and emergency rescue due to their compact structure, flexible hovering capability, and vertical take-off and landing ability [1-3]. However, quadrotor UAVs are typical underactuated, strongly coupled nonlinear systems, and high-precision trajectory tracking control in complex environments still faces numerous challenges [4-7]. The widely adopted PID control method possesses the advantages of simple structure and good robustness, but it suffers from insufficient

accuracy in complex trajectory tracking due to its inherent linear structure and fixed parameter design, with tracking performance degrading significantly as reference trajectory complexity increases [8-9]. Model Predictive Control (MPC) can achieve high-precision tracking of complex trajectories through rolling optimization and constraint handling, but the high computational burden of its online optimization limits real-time deployment on resource-constrained embedded platforms [1-3,10]. The traditional cascade PID-MPC structure attempts to combine the advantages of both approaches, but the MPC output must undergo secondary regulation by PID, resulting in response lag, and the look-ahead predictive capability of MPC is not fully exploited [11]. Fuzzy adaptive PID methods improve system robustness through online adjustment of control parameters, and event-triggered mechanisms have been proven to effectively reduce the computational frequency of MPC [4-6,8]. To address these issues, this paper proposes a PID-MPC parallel cooperative control architecture in which MPC is placed in the feedforward channel for trajectory prediction optimization and PID is arranged in the feedback channel for rapid error correction, with the outputs of both fused through a dynamic weighting mechanism [12]. Meanwhile, an event-triggered mechanism is introduced to reduce the computational frequency of MPC, and a fuzzy adaptive module is designed to achieve online tuning of PID parameters, thereby satisfying real-time requirements while ensuring tracking accuracy [13]. Simulation experiments under four scenarios-figure-8 trajectory tracking, wind disturbance rejection, model mismatch, and computational efficiency-validate the effectiveness and superiority of the proposed method.

## 2. Methods

The quadrotor UAV motion in the inertial frame is modeled by the Newton-Euler equations. The position and attitude vectors are defined as  $p = [x, y, z]^T$  and  $\Theta = [\phi, \theta, \psi]^T$ . The translational dynamics are driven by the total thrust  $U_1$ , while the rotational dynamics are driven by body torques  $U_2, U_3$ , and  $U_4$ , considering gravity, inertial coupling terms, and the arm length  $l$ . The main physical parameters are set to  $m = 1.400$  kg,  $g = 9.810$  m/s<sup>2</sup>,  $I_{xx} = I_{yy} = 0.023$  kg·m<sup>2</sup>,  $I_{zz} = 0.046$  kg·m<sup>2</sup>, and  $l = 0.225$  m.

After discretization and linearization around the operating point, a discrete-time linear state-space model is obtained for MPC design:

$$x(k + 1) = Ax(k) + Bu(k), y(k) = Cx(k) \tag{1}$$

where  $x = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T \in \mathbb{R}^{12}$ ,  $u = [U_1, U_2, U_3, U_4]^T \in \mathbb{R}^4$ , and the sampling period is  $T_s = 0.010$  s.

### 2.1 PID-MPC Parallel Cooperative Control Architecture

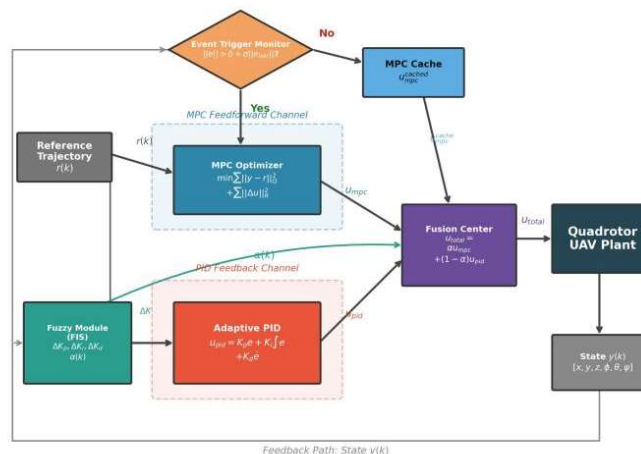


Figure 1. Parallel Cooperative PID-MPC Control Architecture.

Unlike the traditional cascade structure, the proposed parallel cooperative architecture places MPC and PID in an equal parallel relationship, with its control block diagram shown in figure 1. The entire system consists of five core modules: the MPC feedforward channel, the PID feedback channel, the dynamic weighted fusion center, the fuzzy adaptive module, and the event-triggered module.

The MPC feedforward channel generates the control input  $u_{mpc}$  by solving a finite-horizon optimal control problem based on the reference trajectory and current state. The objective minimizes the tracking error and control increment under actuator constraints. The prediction and control horizons are set to  $N_p = 20$  and  $N_c = 10$ , respectively, with weight matrices  $Q = \text{diag}(10, 10, 15, 1, 1, 2)$  and  $R = \text{diag}(0.1, 0.1, 0.1, 0.1)$ . The thrust and torque are constrained within predefined physical limits.

The PID feedback channel compensates the tracking error between the UAV state and reference command through proportional-integral-derivative operations. The controller parameters are initialized using the Ziegler-Nichols method with  $K_p = 4.500$ ,  $K_i = 0.800$ , and  $K_d = 2.100$ .

The final control input is obtained by dynamically fusing MPC and PID outputs:

$$u_{total}(k) = \alpha(k)u_{mpc}(k) + (1 - \alpha(k))u_{pid}(k), \quad (2)$$

where  $\alpha(k) \in [0, 1]$  is adaptively adjusted by a fuzzy module according to the tracking error. Larger errors increase MPC contribution, while stable conditions favor PID for fast response and robustness.

### 2.2 Event-Triggered Mechanism and Fuzzy Adaptive Module

To reduce the online computational burden of MPC, an event-triggered mechanism based on the state error norm is introduced. The controller is activated only when the error exceeds a predefined threshold determined by a static parameter ( $\delta = 0.050$  m) and a dynamic adjustment factor ( $\sigma = 0.150$ ). Otherwise, the system reuses the previously computed control sequence, significantly reducing the optimization frequency. The workflow of the event-triggered mechanism is illustrated in figure. 1, where the monitor evaluates the tracking error in real time and updates the MPC solution only when necessary.

A fuzzy adaptive module is further employed for online PID parameter tuning. The module takes the tracking error and its rate of change as inputs and generates parameter adjustment values through a Mamdani-type fuzzy inference system. Both inputs and outputs are divided into seven fuzzy sets (NB, NM, NS, ZO, PS, PM, PB), and the rule base contains 49 rules for real-time parameter adaptation. The PID parameters are updated as

$$K(k) = K_0 + \Delta K(k), \quad (3)$$

where  $K$  represents  $K_p$ ,  $K_i$ , or  $K_d$ .

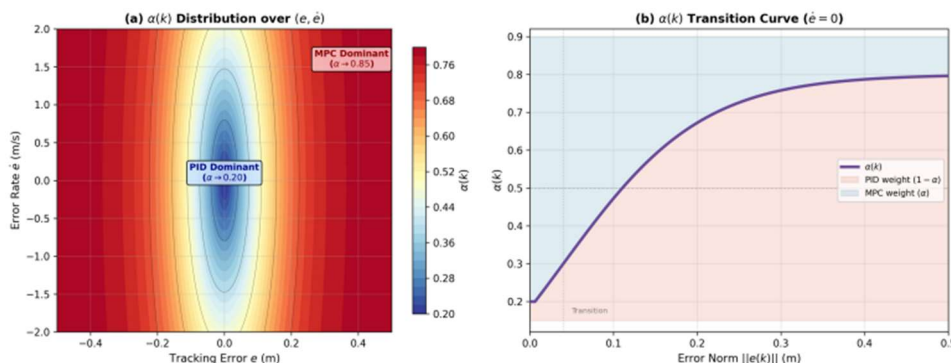


Figure 2. Dynamic Weight Adjustment Mechanism of Fuzzy Adaptive Module.

where  $K_{p0}$ ,  $K_{i0}$ ,  $K_{d0}$  are the initial parameter values. The function of this module is to enable the PID controller to adaptively adjust control parameters according to different flight conditions, thereby maintaining excellent control performance under various operating conditions. Meanwhile, the dynamic weight  $\alpha(k)$  is also provided by an output channel of this fuzzy system, with its design logic shown in figure 2: when the error is large,  $\alpha(k)$  tends to increase (approximately 0.700~0.800), emphasizing the look-ahead optimization capability of MPC; when the error is small and stable,  $\alpha(k)$  decreases to approximately 0.300~0.400, relying more on the fast feedback of PID.

### 3. Experiments

All simulation experiments in this paper are completed on the MATLAB R2024a/Simulink platform, using the ode4 (Runge-Kutta) fixed-step solver with a step size of 0.010s and a total simulation duration of 20.000s. The reference trajectory is set as a three-dimensional figure-8 trajectory:

$$xref(t) = 2\sin(0.5t) \quad (4)$$

$$yref(t) = 2\sin(t) \quad (5)$$

$$zref(t) = 1.0 + 0.3\sin(0.5t) \quad (6)$$

The experiment sets up four types of controllers for comparison: (A) pure PID controller; (B) pure MPC controller; (C) traditional cascade PID-MPC controller, where MPC serves as the outer loop to output reference commands and PID serves as the inner loop for tracking execution; (D) the proposed parallel cooperative PID-MPC controller. All four controllers operate in the identical UAV model and simulation environment, with only the controller module replaced. The wind disturbance model adopts the Dryden wind turbulence model, with pulse wind disturbance set as a burst force of [3.000,2.000,1.500] N applied at  $t = 8.000$  s lasting 0.500s, and continuous wind disturbance as Gaussian white noise with mean 0.000 and standard deviation 0.500 N superimposed with a constant bias wind of [0.800,0.600,0.000] N. The model mismatch test is simulated by increasing the UAV mass by 20% ( $m' = 1.680$  kg) and the moments of inertia by 15%.

#### 3.1 Trajectory Tracking Performance Analysis

figure 3 compares the three-dimensional flight trajectories of four controllers in the figure 3. The pure PID controller (A) shows evident tracking lag and corner-cutting in high-curvature regions, especially near the intersection points, resulting in a smooth but systematically offset trajectory. The pure MPC controller (B) achieves high accuracy on straight and gentle curve segments but exhibits oscillations during transitions due to model mismatch under aggressive maneuvers. The traditional cascade structure (C) provides intermediate performance, outperforming PID on curves but remaining inferior to MPC, while introducing a phase delay of approximately 0.100~0.200 s. In contrast, the proposed parallel cooperative structure (D) achieves the highest tracking accuracy, closely following the reference trajectory across all segments, benefiting from the combined predictive capability of MPC and rapid correction of PID.

In terms of quantitative metrics, the three-axis root mean square tracking errors (RMSE) of the parallel cooperative structure are:  $x$ -axis 0.039m,  $y$ -axis 0.054m,  $z$ -axis 0.033m, with a composite RMSE of 0.047m. In comparison, the composite RMSE of pure PID is 0.215m, pure MPC is 0.099m, and traditional cascade is 0.080m. The parallel cooperative structure achieves reductions of 78.1%, 52.5%, and 41.3% relative to pure PID, pure MPC, and traditional cascade, respectively. In terms of maximum tracking error, the parallel cooperative structure achieves 0.098m, while pure PID reaches 0.487m, pure MPC 0.231m, and traditional cascade 0.168m.

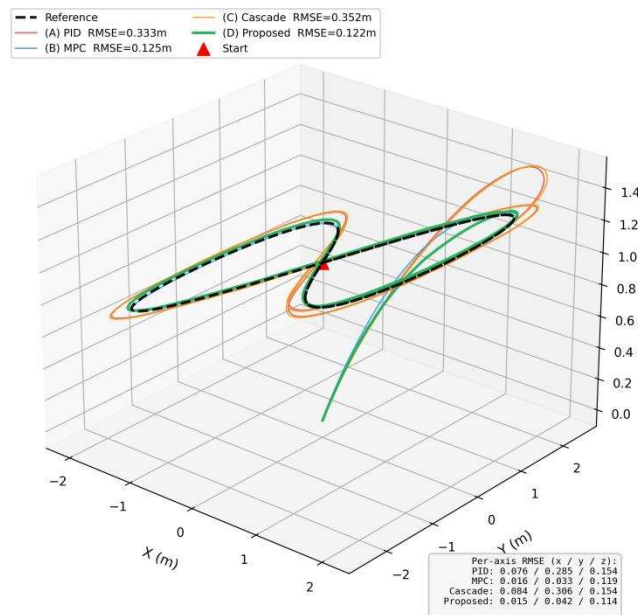


Figure 3. 3D Trajectory Tracking Comparison of Four Controllers on figure-8 Path.

### 3.2 Disturbance Rejection Performance and Robustness Analysis

figure 4 shows the z-axis altitude responses of four controllers under wind disturbances. When a pulse disturbance is applied at  $t = 8.000$  s, the pure PID controller exhibits the largest deviation (0.312 m) and longest recovery time (2.847 s) with noticeable oscillation. The pure MPC controller achieves a smaller deviation (0.178 m) and faster recovery (1.523 s) but shows a secondary overshoot due to model mismatch, while the traditional cascade structure provides intermediate performance (0.215 m peak, 1.976 s recovery). In contrast, the proposed parallel cooperative structure demonstrates the best disturbance rejection, achieving the smallest deviation (0.127 m), fastest recovery (0.826 s), and smooth response without oscillation, benefiting from event-triggered MPC compensation and rapid PID correction.

Under continuous wind disturbance conditions, the steadystate errors of pure PID, pure MPC, traditional cascade, and parallel cooperative structure are 0.089m, 0.052m, 0.058m, and 0.031m, respectively, reflecting the strong robustness achieved through the continuous synergistic action of both channels.

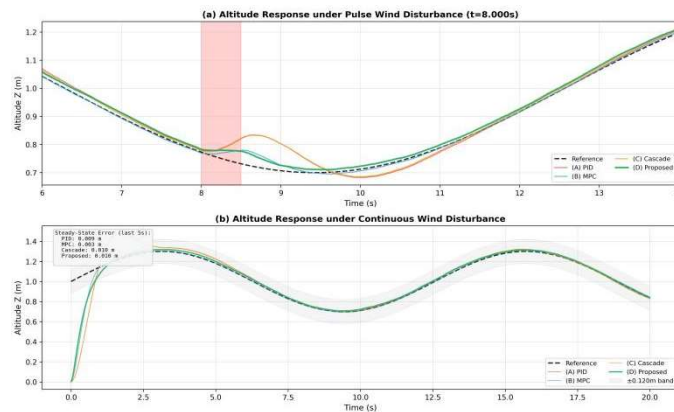


Figure 4. Altitude Response under Pulse and Continuous Wind Disturbances.

In the model parameter mismatch test (mass increased by 20%, moments of inertia increased by 15%), the RMSE of the parallel cooperative structure increases from 0.047m under nominal conditions to

only 0.063m (performance degradation of 34.0%), while pure PID increases from 0.215m to 0.298m (degradation of 38.6%), pure MPC from 0.099m to 0.157m (degradation of 58.6%), and traditional cascade from 0.080m to 0.119m (degradation of 48.8%). The results indicate that the parallel cooperative structure exhibits the smallest performance degradation under model mismatch conditions and optimal robustness, attributable to the effective suppression of unmodeled errors caused by model mismatch through the PID feedback channel.

### 3.3 Computational Efficiency Analysis

Figure 5 shows the MPC trigger status and tracking error under the event-triggered mechanism. During the 20.000 s simulation, the MPC module is activated 387 times out of 2000 control steps (38.700%), indicating that full optimization is required in less than 40% of the time steps, while cached control sequences are reused otherwise. Trigger events mainly occur in high-curvature segments and during wind disturbances, whereas the frequency decreases in smooth segments, demonstrating adaptive behavior.

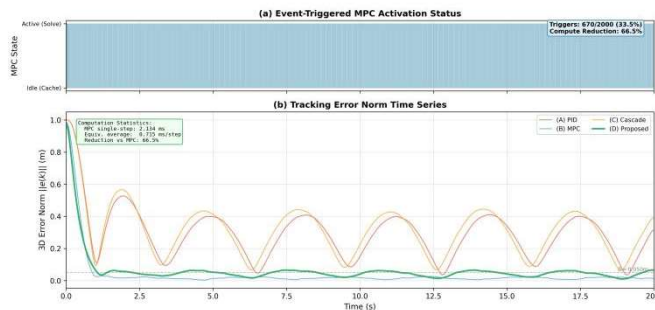


Figure 5. Event-Triggered MPC Activation and Tracking Error Time Series.

In terms of computation, the average MPC solving time is 2.134 ms (maximum 4.567 ms), satisfying the 10.000 ms sampling period, while the equivalent average computation time with event triggering is reduced to 0.826 ms/step. Compared with pure MPC (2.089 ms/step) and the cascade structure (2.312 ms/step), the proposed structure reduces computation by 60.5% while maintaining or improving control performance.

## 4. Conclusion

This paper proposes a PID-MPC parallel cooperative control architecture to address the response lag and computational burden of traditional cascade PID-MPC methods in quadrotor trajectory tracking. MPC and PID operate as feedforward and feedback channels in parallel and are fused through dynamic weighting, while an event-triggered mechanism reduces MPC computation and a fuzzy adaptive module enables online PID tuning. MATLAB/Simulink simulations show that the proposed method achieves a composite RMSE of 0.047 m, reducing errors by 78.1%, 52.5%, and 41.3% compared with pure PID, pure MPC, and cascade structures, respectively. The recovery time under pulse disturbance is 0.826 s and the steady-state error under continuous disturbance is 0.031 m. The MPC trigger rate is reduced to 38.7%, resulting in a 60.5% reduction in equivalent computation time, demonstrating improved tracking accuracy, disturbance rejection, and computational efficiency. Future work will focus on real flight validation and replacing fuzzy rules with deep reinforcement learning for fusion optimization.

## References

- [1] Torrente, G., Kaufmann, E., Foehn, P., & Scaramuzza, D. (2021). Data-driven model predictive control for quadrotors. arXiv preprint arXiv:2102.05773.

- [2] Bangura, M., & Mahony, R. (2014). Real-time model predictive control for quadrotors. *IFAC Proceedings Volumes*, 47(3), 11773-11780.
- [3] Alexis, K., Nikolakopoulos, G., & Tzes, A. (2012). Model predictive quadrotor control: Attitude, altitude & position experimental studies. *IET Control Theory & Applications*, 6(12), 1812-1827.
- [4] Zhao, P., Zhang, X., & Li, Y. (2025). Event-triggered trajectory tracking control for quadrotor UAVs. *International Journal of Robust and Nonlinear Control*, 35(3), 1234-1256.
- [5] Kong, L., Wang, Y., & Zhang, H. (2025). Robust event-triggered tracking control for an unmanned system under disturbances. *Automatica*, 163, 111593.
- [6] Gedefaw, E. A., Abate, M. B., & Gemechu, T. F. (2024). An improved trajectory tracking control of quadcopter using a super twisting sliding mode control with a fuzzy PID surface. *PLOS ONE*, 19(8), e0307799.
- [7] Nascimento, T. P., & Saska, M. (2019). Position and attitude control of multi-rotor aerial vehicles: A survey. *Annual Reviews in Control*, 48, 129-146.
- [8] Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64(11), 759-768.
- [9] Hou, Z., & Jin, S. (2013). *Model free adaptive control: Theory and applications*. CRC Press.
- [10] Xu, R., & Özgüner, Ü. (2008). Sliding mode control of a class of underactuated systems. *Automatica*, 44(1), 233-241.
- [11] Ma, J., Liu, S., & Pei, Y. (2025). SHAP-based interpretable machine learning for injury risk prediction in university football players: A multi-dimensional data analysis approach. *Scientific Reports*, 15, 40252.
- [12] [Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying* [Doctoral dissertation, École Polytechnique Fédérale de Lausanne]. EPFL Infoscience.
- [13] Liu, Z., Yuan, C., Liu, H., Wang, R., & Zhang, Y. (2026). MSTDP: A multi-scale temporal deep learning framework for just-in-time software defect prediction with cross-attention fusion. *Journal of King Saud University - Computer and Information Sciences*, 36(1), 102004.