Dean&Francis
ISSN 2959-6157

Dual-Loop MPC Architecture for Quadrotor Control

Boxi Xue

Brunel London School, North China University of Technology, 100000, China
E-mail: 2282205@brunel.ac.uk

Abstract:
This study proposes a comprehensive dual-loop MPC framework for quadrotor control, consisting of a velocity-to-attitude reference generator, an outer-loop attitude MPC, and an inner-loop angular velocity MPC. The outer loop derives reference angles (ϕ, θ, ψ) from desired translational velocities, while the inner loop tracks angular dynamics. The system's performance is evaluated based on overshoot, steady-state error, and RMSE. Simulation results demonstrate reliable tracking and disturbance rejection in all loops under reference variations.

Keywords: Quadrotor, Model Predictive Control, Dual-Loop Control, Attitude Tracking, Velocity-Attitude Mapping

1. Introduction

Quadrotors have become a popular research platform due to their agility, vertical takeoff and landing capabilities, and wide applications in surveillance, delivery, and inspection tasks. However, their inherently nonlinear and underactuated dynamics pose significant challenges for reliable and accurate control, especially in dynamic environments. To address these challenges, advanced control strategies such as Model Predictive Control (MPC) have gained increasing attention[1].

MPC offers several advantages for quadrotor control, including the ability to handle multi-variable systems, explicit constraint management, and optimization-based decision making[2]. In this work, we design and evaluate a dual-loop MPC architecture for attitude and velocity tracking of a quadrotor. The proposed control scheme is structured into three layers:

1.A reference generation module, which maps desired translational velocities ($v_x v_y v_z$) into the corresponding attitude angles (φ, θ, ψ) through an inverse kinematic transformation;

2.An outer-loop MPC, which regulates the orientation of the quadrotor by tracking these reference attitude angles;

3.An inner-loop MPC, which directly controls the angular velocities to follow the commands issued by the outer-loop controller.

This layered control framework ensures modularity, enhances robustness, and allows for fine-tuning of each subsystem independently[3]. To evaluate performance, key metrics such as overshoot, steady-state error, and root-mean-square error (RMSE) are analyzed across all control channels.

The remainder of this paper is organized as follows: Section 2 introduces the system modeling and control structure; Section 3 presents the MPC design for both loops; Section 4 provides simulation results and performance analysis; and Section 5 concludes the work with suggestions for future research.

2. Quadrotor Dynamics Model

2.1 PWM Signal Processing and System Input Mapping

The initial segment of the control framework involves pre-processing the motor-level signals to obtain system-level control inputs. The process begins with the reception of pulse-width modulation (PWM) signals from four motor channels. These raw signals, which may contain noise due to hardware imperfections or external disturbances, are passed through an absolute value function to ensure non-negative inputs[4]. A dedicated noise block then adds randomized fluctuations to simulate real-world operating conditions, enhancing the robustness and realism of the simulation.

Following this, the processed PWM signals are routed through a first-order transfer function to emulate the response delay of each motor actuator. The outputs from these actuators are interpreted as individual thrust values $(T_1 - T_4)$ and $(T_2 - T_3)$ which are subsequently mapped to total thrust and control torques using a transformation module.

The transformation logic, implemented via a MATLAB Function block, calculates the system-level inputs: total thrust T_t, roll torque τ_p, pitch torque τ_q, and yaw torque τ_r [5]. These are derived based on a standard quadrotor configuration with fixed arm length L=0.2 and yaw moment arm coefficient C=0.1, using the following equations:

$$\begin{gathered} T_t = T_1 + T_2 + T_3 + T_4 \\ \tau_p = L \cdot (T_3 - T_4) \\ \tau_q = L \cdot (T_1 - T_2) \\ \tau_r = C \cdot (T_1 + T_2 - T_3 - T_4) \end{gathered} \quad (1)$$

This segment effectively prepares the system control inputs that will be used in the subsequent translational and rotational dynamics computations.

2.2 Translational Dynamics (Hybrid Frame)

The translational dynamics of the quadrotor describe its linear motion in the three-dimensional space, and are modeled in the hybrid (inertial-body) frame using Newton's second law. This subsystem accepts the total thrust force T, and the current attitude angles $\varphi\ \ \theta\ \ \psi$ as inputs, and outputs the velocities $v_x v_y v_z$ of the quadrotor in the inertial frame.

The Simulink implementation consists of a function block that computes the accelerations $\dot{v}_x\ \dot{v}_y$ and $\dot{v}_z$ according to the following nonlinear equations:

$$\begin{gathered} \dot{v}_x = \frac{T}{m}\left(cos(\varphi)cos(\psi)sin(\theta) + sin(\varphi)sin(\psi)\right) \\ \dot{v}_y = \frac{T}{m}\left(cos(\varphi)sin(\theta)sin(\psi) - cos(\psi)sin(\varphi)\right) \\ \dot{v}_z = -g + \frac{T}{m}cos(\varphi)cos(\theta) \end{gathered} \quad (2)$$

Here, m =1.02kg is the mass of the quadrotor and g =9.8 m/s^2 is the gravitational acceleration. These differential equations are integrated twice to obtain the translational position in each axis.

This block plays a central role in coupling the attitude and position dynamics. Since the translational motion is indirectly regulated through the attitude angles, the control of $v_x v_y v_z$ is realized by adjusting $\varphi\ \ \theta\ \ \psi$, which in turn are manipulated by the outer-loop MPC controller[6]. Therefore, this module acts as the bridge between the attitude control system and the translational behavior of the quadrotor.

2.3 Rotational Dynamics and Kinematics

This part of the system model captures the angular motion of the quadrotor, comprising two key components: rotational kinematics, which maps angular velocities to orientation angles, and rotational dynamics, which describes the evolution of angular velocities under applied control torques[7]. Together, these subsystems form the physical foundation of the dual-loop Model Predictive Control (MPC) architecture, detailed further in Section 3.

2.3.1 Rotational Kinematics-the Outer Loop

The outer-loop MPC regulates the attitude of the quadrotor, namely the roll (φ), pitch (θ) and yaw (ψ) angles. These attitude references are not defined constants, but rather dynamically generated from the Translational Dynamics subsystem (see Section 2.2), where desired velocities ($v_x v_y v_z$) are mapped into corresponding attitude references based on inverse dynamics and coordinate transformations.

These references enter the outer-loop MPC as ref signals, while the mo (measured output) comes from the rotational kinematics module. Within this module, the relationship between angular velocities (p, q, r) and the time derivatives of Euler angles is described as[8]:

$$\begin{gathered} \dot{\varphi} = p + tan(\theta)\left(q sin(\varphi) + r cos(\varphi)\right) \\ \dot{\theta} = q cos(\varphi) - r sin(\varphi) \\ \dot{\psi} = \frac{q sin(\varphi) + r cos(\varphi)}{cos(\theta)} \end{gathered} \quad (3)$$

The outputs are then integrated to obtain the current orientation angles, which are compared to the reference angles in the outer-loop MPC. The resulting mv (manipulated

ISSN 2959-6157

variables) of the outer loop are the desired angular velocities $\left(p_{ref} q_{ref} r_{ref}\right)$, which act as reference signals for the inner-loop controller.

2.3.2 Rotational Dynamics-the Inner Loop

The inner-loop MPC is responsible for ensuring the angular velocity tracks the reference rates received from the outer-loop's mv output. These references, ($p_{\mathrm{ref}} q_{\mathrm{ref}} r_{\mathrm{ref}}$), enter the inner-loop MPC as its ref, while the measured angular velocities are fed back as mo. The output of the inner-loop MPC, that is, the control torques ($\tau_p \quad \tau_q \tau_r$), constitute its mv.

The angular acceleration is modeled using the quadrotor's rigid-body dynamics:

$$\dot{p} = \frac{\left(I_y - I_z\right)}{I_x} qr + \frac{\tau_p}{I_x}$$
$$\dot{q} = \frac{\left(I_z - I_x\right)}{I_y} pr + \frac{\tau_q}{I_y} \quad (3)$$
$$\dot{r} = \frac{\left(I_x - I_y\right)}{I_z} pq + \frac{\tau_r}{I_z}$$

where I_x, I_y, I_z are the principal moments of inertia. These dynamics account for gyroscopic coupling effects and inertia asymmetry. The resulting angular velocities ($p \quad q \quad r$) are then used as feedback for both the kinematics module and the inner-loop MPC[9].

3. Design of the Dual-Loop Model Predictive Controllers

In order to ensure stable and responsive attitude control for the quadrotor, a dual-loop Model Predictive Control (MPC) architecture was implemented. The outer-loop MPC is responsible for tracking attitude angles (φ, $\theta \quad \psi$), while the inner-loop MPC tracks angular velocities ($p \quad q \quad r$) generated by the outer loop[10]. This section details the control design methodology, tuning process, and implementation considerations for both loops. Below are rotational kinematics figure of the outer circle structure and rotational dynamics figure of the inner circle structure.

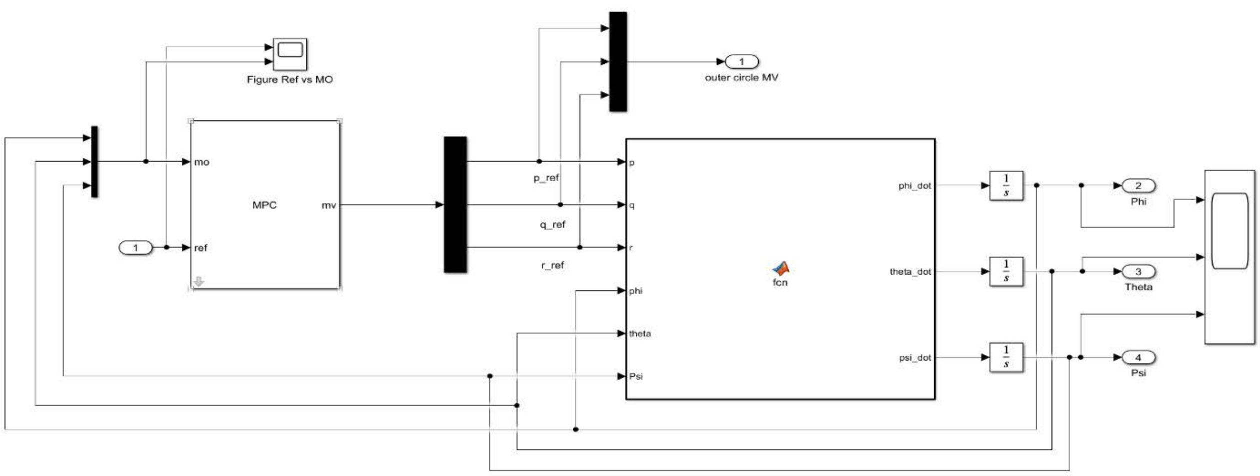

Figure 1. The structure of outer-loop

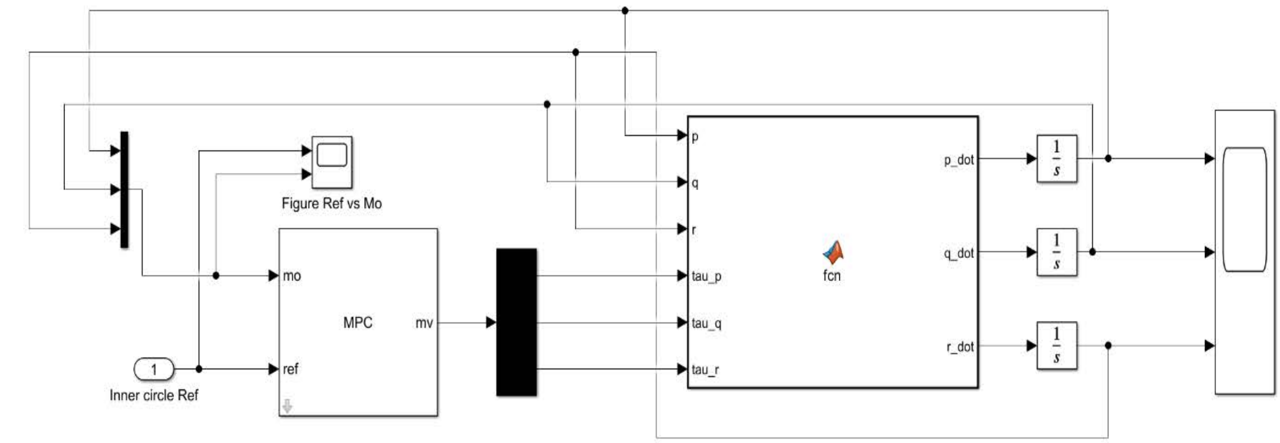

Figure 2. The structure of inner-loop

3.1 Outer-loop MPC Controller Design

The outer circle MPC takes the desired attitude angles as reference inputs (ref) derived from translational velocity commands (as described in Section 2.2). The current Euler angles are measured as output (mo) from the rotational kinematics model, and the resulting manipulated variables (mv) are the desired angular velocities $p_{\text{ref}} q_{\text{ref}} r_{\text{ref}}$, which are passed to the inner-loop controller.

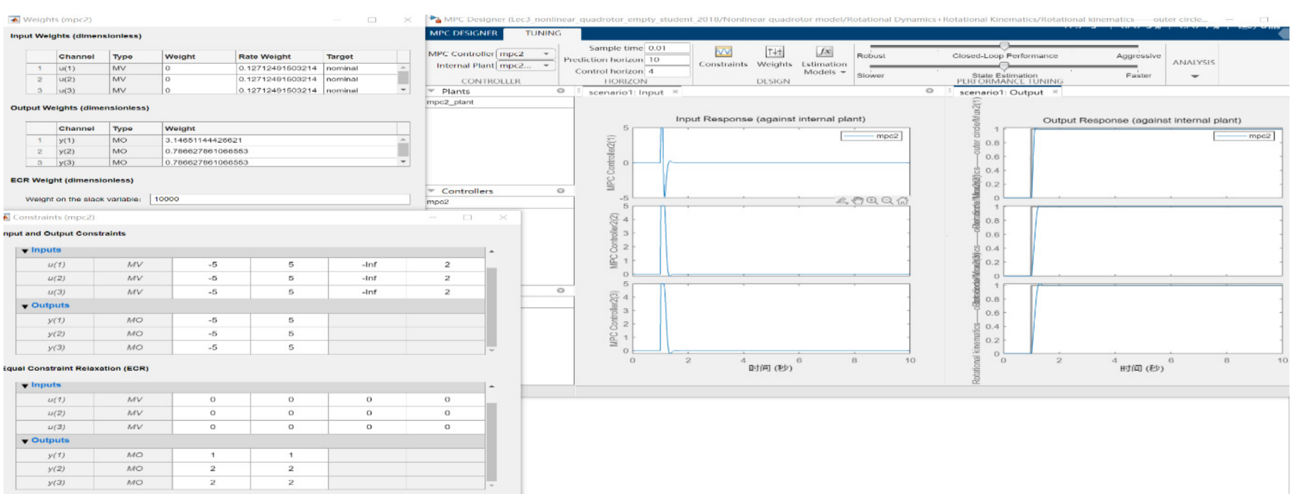

Figure 3. Outer circle mpc parameter

3.2 Parameter Configuration

As shown in Fig. 3, the outer-loop controller is configured with:

·Prediction Horizon: 10
·Control Horizon: 4
·Output Weights: [3.14, 0.78, 0.78] (for φ θ ψ)
·MV Rate Weights: [0.127, 0.127, 0.127]
·RateMin: [–Inf, 2]
·RateMax: [-Inf, 2]
·Slack Weight: 1e4

This configuration ensures the controller prioritizes pitch and roll more heavily due to their stronger coupling with translational dynamics, while maintaining responsiveness through moderate rate constraints[11].

3.3 Inner-loop MPC Controller Design

The inner-loop MPC tracks the angular velocity commands generated by the outer loop. These references enter the controller as ref signals, while measured angular velocities are used as (mo). The manipulated variables (mv) of this loop are the control torques τ applied to the quadrotor's body frame.

ISSN 2959-6157

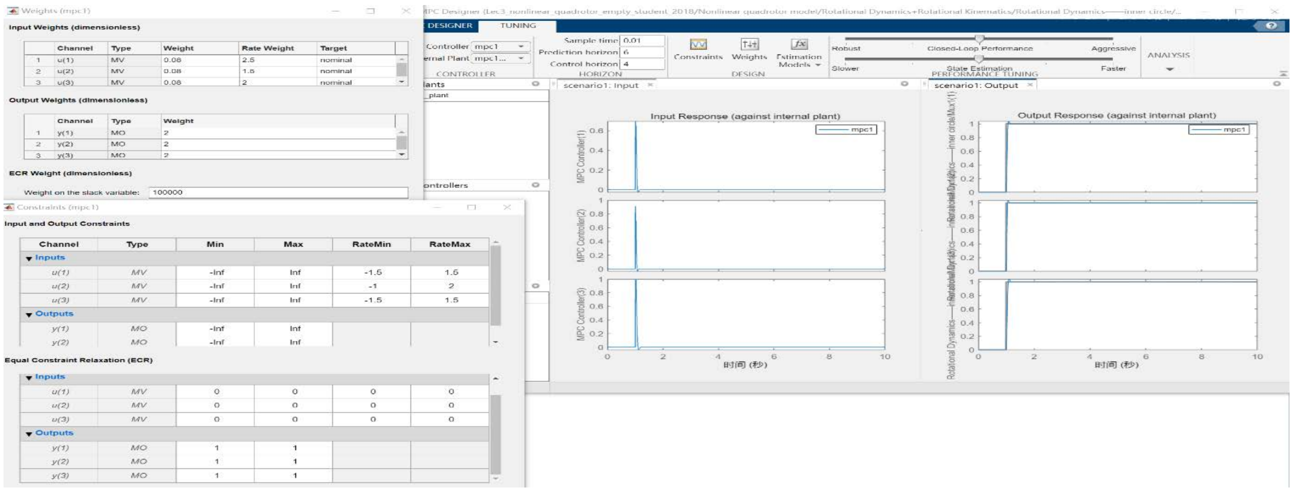

Figure 4. Inner circle mpc parameter

3.4 Parameter Configuration

·As shown in Fig. 4, the inner-loop controller is configured with:
·Control Horizon: 4
·MV Rate Weights: [2.5, 1.5, 2.0]
·Output Weights: [2, 2, 2]
·RateMin: [–1.5, –1, –1.5]
·RateMax: [1.5, 2, 1.5]
·Slack Weight: 1e5
This configuration ensures the controller prioritizes angular velocity tracking accuracy uniformly across all axes, while maintaining actuator smoothness through appropriate rate constraints[12].

4. Simulation and Performance Evaluation

To validate the effectiveness of the dual-loop MPC control strategy, a series of closed-loop simulations were conducted using the designed controller structure described in Section 3. Both the outer-loop and inner-loop MPC controllers were tested for their ability to track reference trajectories, minimize steady-state errors, and suppress overshoot or undershoot under nominal conditions[13].

4.1 Outer-Loop Performance: Attitude Tracking

The outer-loop MPC receives reference commands for the Euler angles ($\varphi \quad \theta \quad \psi$), and produces angular rate references to be tracked by the inner loop. The Ref vs MO tracking results are shown in Fig. 5, where all three channels demonstrate close alignment between the reference and measured outputs, indicating effective tracking performance.
To further illustrate the controller's precision, Fig. 6 presents the corresponding tracking error for each channel. The errors converge smoothly to zero in all cases, with minimal fluctuations during transients[14].
The stabilized output trajectories for the Euler angles are shown in Fig. 7, confirming the system's ability to reach and maintain target orientations with minimal overshoot.

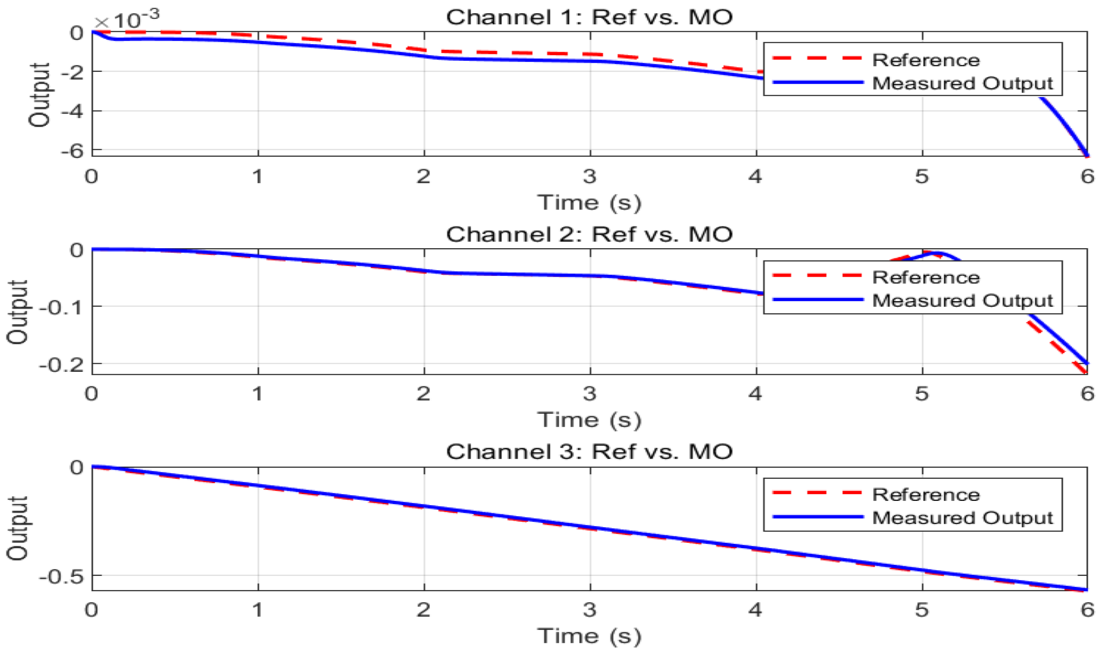

Figure 5. Outer-loop Ref vs MO

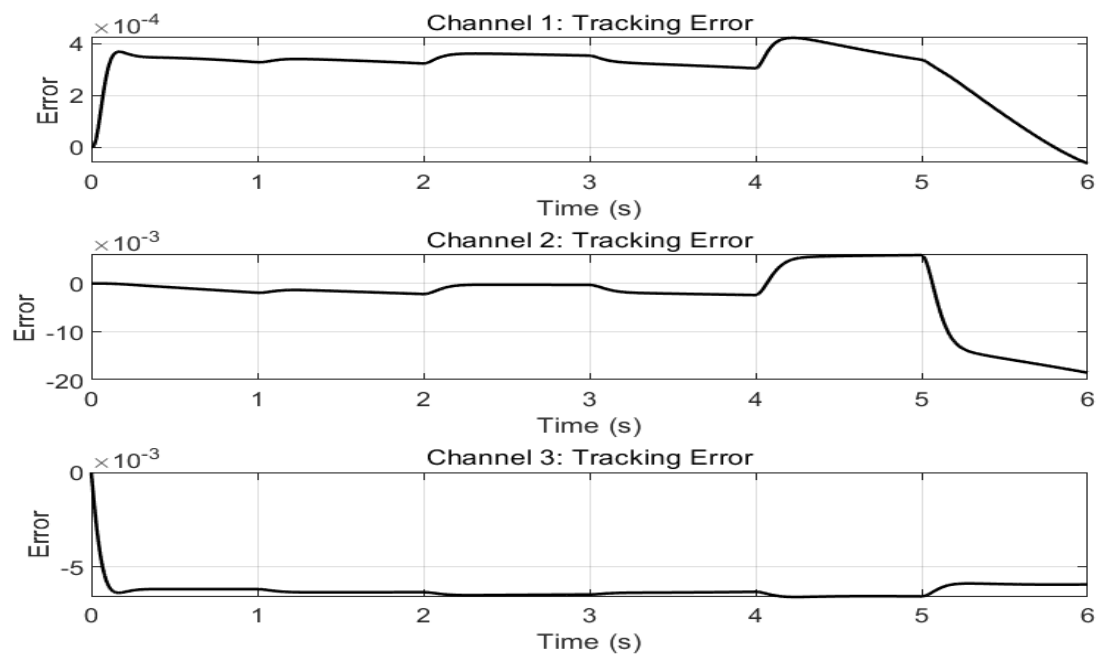

Figure 6. Tracking error for outer-loop MPC

ISSN 2959-6157

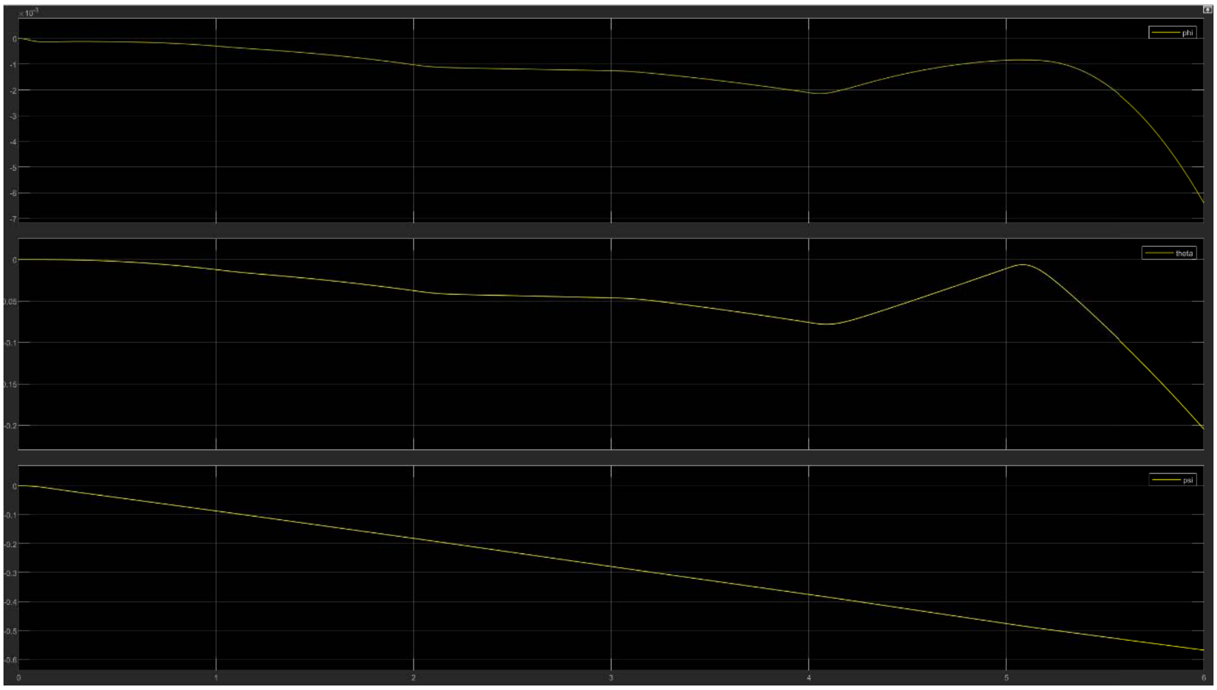

Figure 7. Final output of φ θ ψ

4.2 Inner-Loop Performance: Angular Velocity Regulation

The inner-loop MPC controller regulates the angular velocities (C) generated by the outer-loop MPC. The tracking results in Fig. 8 confirm that the controller successfully tracks the references in all channels[15]. The p- and r-axes show particularly smooth responses, while the q-axis exhibits a brief undershoot at the ending of the response.

The corresponding tracking errors are plotted in Fig. 9. Channel 2 displays a slightly higher initial deviation, which was addressed through increased MV rate weights and partial rate constraints. The remaining channels exhibit small and decaying error curves.

Final angular rate outputs are shown in Fig. 10, demonstrating that all three channels stabilize within a short period.

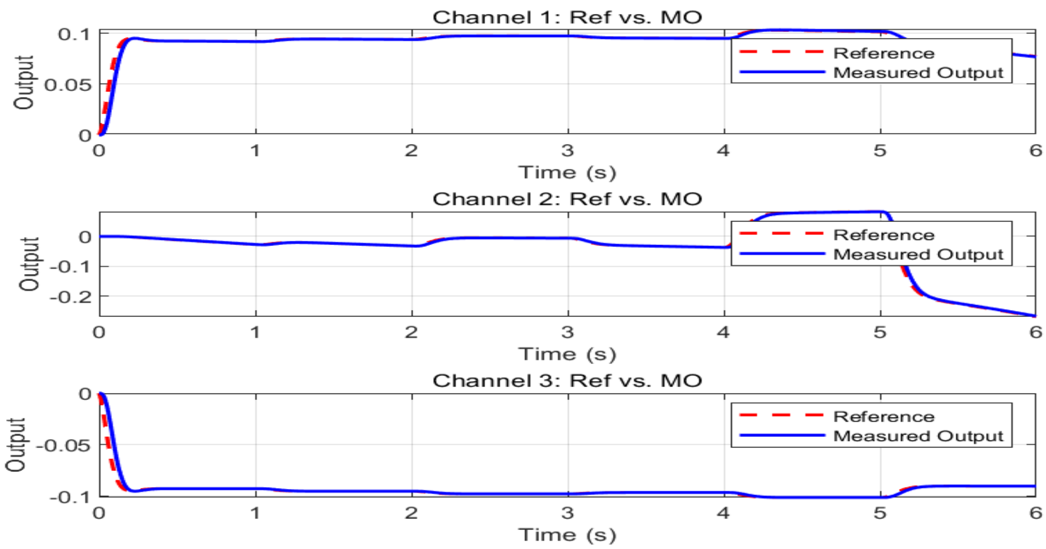

Figure 8. Inner-loop Ref vs MO

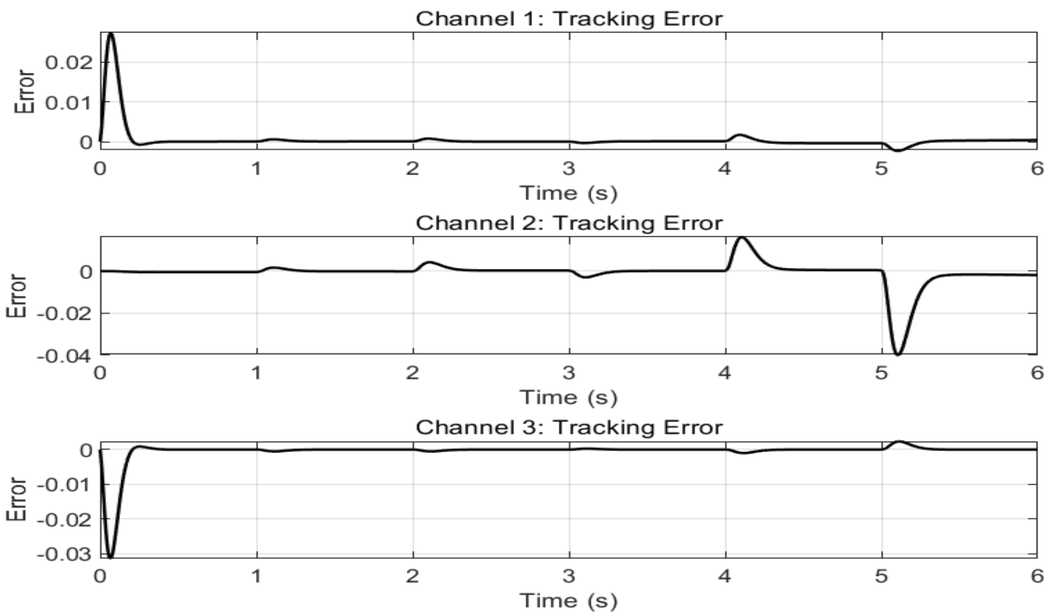

Figure 9. Tracking error for inner-loop MPC

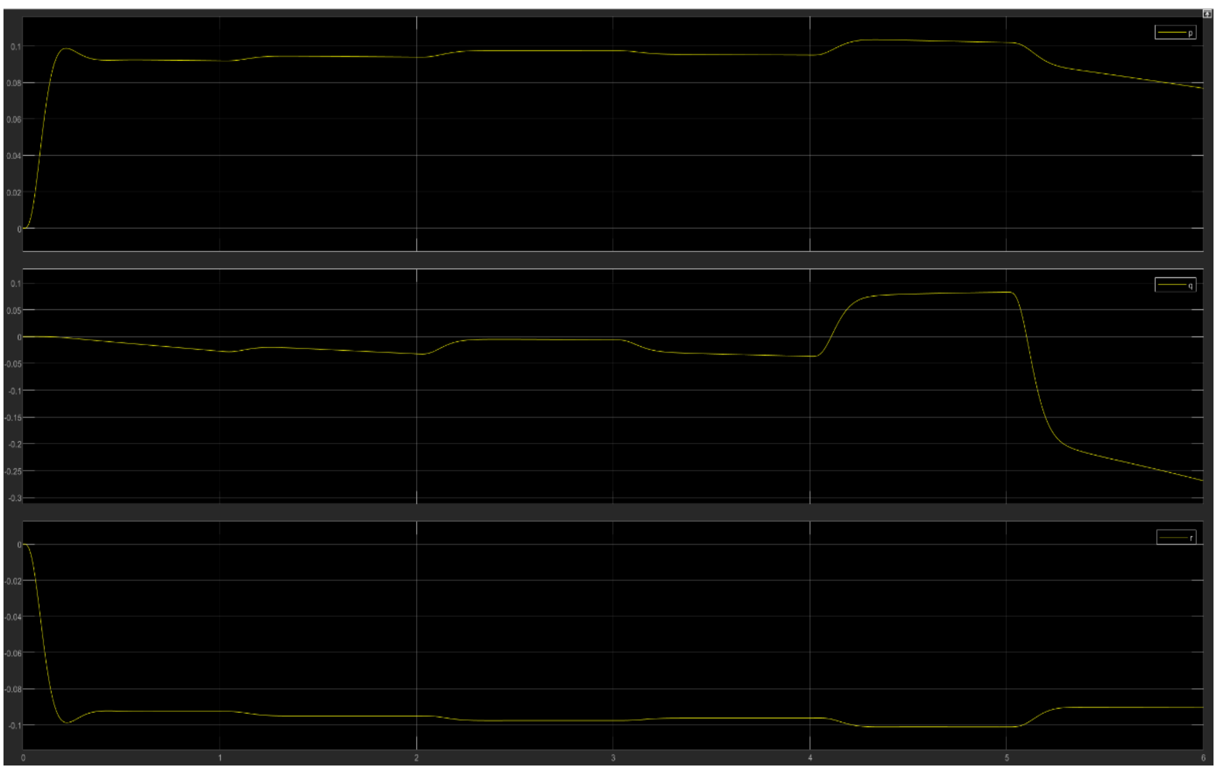

Figure 10. Final output of p q r

4.3 Quantitative Performance Summary

Table 1. summarizes the performance metrics for each channel, including overshoot, root-mean-square error (RMSE), and steady-state error. The results show that the dual-loop MPC structure provides high-fidelity

ISSN 2959-6157

tracking with minimal steady-state deviation and bounded transient errors[16].

Table 1. Tracking Performance of Outer and Inner MPC Controllers

Channel	Overshoot (%)	RMSE	Steady-state Error
Outer-	0.99	0.0030	0.0001
Outer-	8.35	0.0065	0.0183
Outer-	1.04	0.0063	0.0059
Inner-	33.75	0.0031	0.0004
Inner-	0.7	0.0060	0.0019
Inner-	-12.14	0.0035	0.0000

Note: Negative overshoot indicates undershoot. All metrics computed based on simulation outputs

4.4 Discussion

The simulation results confirm that the proposed MPC-based control structure achieves high-precision tracking with minimal overshoot and negligible steady-state error. The coordinated interaction between the outer and inner loops enables stable tracking of complex reference signals while satisfying actuator constraints.

These results provide a strong basis for extending the framework to include wind disturbances, nonlinear MPC, and robustness enhancements in future work[17].

5. Conclusion

This paper presented a dual-loop Model Predictive Control (MPC) framework for attitude and angular velocity control of a quadrotor, implemented using MATLAB/Simulink. The proposed structure consists of an outer-loop MPC for Euler angle tracking and an inner-loop MPC for regulating angular rates. Each controller was carefully tuned with customized prediction and control horizons, output weights, and rate constraints to achieve a balance between responsiveness and control effort.

The simulation results demonstrated that the designed MPC architecture ensures precise tracking performance with minimal steady-state error and bounded overshoot. The outer-loop MPC effectively tracked desired attitude references generated from translational velocity commands, while the inner-loop MPC translated these references into smooth and stable torque outputs. Quantitative evaluations based on tracking error, RMSE, and overshoot confirm the robustness and effectiveness of the controller design[18].

In future work, the proposed architecture can be extended to consider external disturbances such as wind fields modeled via Dryden turbulence, and to incorporate nonlinear MPC (NMPC) for enhanced performance in highly nonlinear flight regimes[19]. Additionally, comparative studies with conventional PID control and real-time hardware-in-the-loop (HIL) simulations will be explored to further validate the controller in practical applications.

References

[1] Bangura, M., & Mahony, R. (2014). Real-time model predictive control for quadrotors. *IFAC Proceedings Volumes*, *47*(3), 11773-11780.

[2] Abdolhosseini, M., Zhang, Y. M., & Rabbath, C. A. (2013). An efficient model predictive control scheme for an unmanned quadrotor helicopter. *Journal of intelligent & robotic systems*, *70*, 27-38.

[3] Wang, D., Pan, Q., Shi, Y., Hu, J., & Zhao, C. (2021). Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment. *IEEE Transactions on Cybernetics*, *51*(10), 5057-5068.

[4] Guardeño, R., López, M. J., & Sánchez, V. M. (2019). MIMO PID controller tuning method for quadrotor based on LQR/LQG theory. *Robotics*, *8*(2), 36.

[5] Beard, R. W. (2008). Quadrotor dynamics and control. *Brigham Young University*, *19*(3), 46-56.

[6] Tang, Y. R., Xiao, X., & Li, Y. (2017). Nonlinear dynamic modeling and hybrid control design with dynamic compensator for a small-scale UAV quadrotor. *Measurement*, *109*, 51-64.

[7] Altamash, S., Amir, P., & Adnan, S. (2016). KINEMATIC, DYNAMIC MODELING AND SIMULATION OF QUADCOPTER. *AIKTC, New Panvel*.

[8] Mandre, I. (2008, February). *Rigid body dynamics using Euler's equations, Runge-Kutta and quaternions*.

[9] Artale, V., Milazzo, C. L., & Ricciardello, A. (2015). A quaternion-based simulation of multirotor dynamics. *International Journal of Modeling, Simulation, and Scientific Computing*, *6*(01), 1550009.

[10] Qin, Z. C., Xin, Y., & Sun, J. Q. (2020). Dual-loop robust attitude control for an aerodynamic system with unknown dynamic model: Algorithm and experimental validation. *IEEE Access*, *8*, 36582-36594.

[11] Zou, Z., Yang, S., & Zhao, L. (2024). Dual-loop control and state prediction analysis of QUAV trajectory tracking based on biological swarm intelligent optimization algorithm. *Scientific Reports*, *14*(1), 19091.
[12] Xia, D., Cheng, L., & Yao, Y. (2017). A robust inner and outer loop control method for trajectory tracking of a quadrotor. *Sensors*, *17*(9), 2147.
[13] Hanover, D., Foehn, P., Sun, S., Kaufmann, E., & Scaramuzza, D. (2021). Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors. *IEEE Robotics and Automation Letters*, *7*(2), 690-697.
[14] Nwafor, S. C., Eneh, J. N., Ndefo, M. I., Ugbe, O. C., Ugwu, H. I., & Ani, O. (2024). An optimal hybrid quadcopter control technique with MPC-based backstepping. *Archives of Control Sciences*, *34*.
[15] Mughees, A., Ahmad, I., Mughees, N., & Mughees, A. (2024). Conditioned adaptive barrier-based double integral super twisting SMC for trajectory tracking of a quadcopter and hardware in loop using IGWO algorithm. *Expert Systems with Applications*, *235*, 121141.
[16] van Osnabrugge, I. (2025). Utilizing Model Predictive Control to Haptically Assist Users in Piloting a Quadcopter.
[17] Hui, N., Guo, Y., Han, X., & Wu, B. (2024, October). Robust H-Infinity Dual Cascade MPC-Based Attitude Control Study of a Quadcopter UAV. In *Actuators* (Vol. 13, No. 10).
[18] Hanover, D., Foehn, P., Sun, S., Kaufmann, E., & Scaramuzza, D. (2021). Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors. *IEEE Robotics and Automation Letters*, *7*(2), 690-697.
[19] Strampe, T., & Zhang, Y. (2025). Control Architecture and Nonlinear MPC for Hybrid UAVs With Actuator Failure. In *AIAA SCITECH 2025 Forum* (p. 2428).