Nonlinear Model Predictive Controller Trajectory Tracking for a Quadrotor

Zachary Serocki
Dept. of Robotics Engineering
Worcester Polytechnic Institute
Worcester, United States of America
zdserocki@wpi.edu

***Abstract*—This paper presents the design and implementation of a Nonlinear Model Predictive Controller (NMPC) for trajectory tracking of a quadrotor system. The controller was developed using a dynamic model that incorporates position, velocity, and attitude states, and was tuned to balance tracking accuracy with control effort through appropriate weighting of the cost function terms. Various test trajectories, including quintic point-to-point motion and circular paths, were used to evaluate controller performance. The NMPC achieved low root-mean-square error (RMSE) values and real-time computation rates exceeding 700 Hz. Disturbance rejection tests revealed the controller's limitations under large external forces, emphasizing the importance of accurate modeling and constraint formulation. Overall, the results validate the effectiveness of the NMPC framework for precise and efficient quadrotor control under nominal operating conditions.**

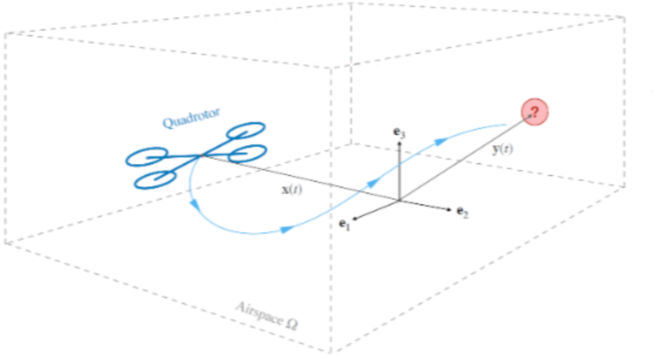

Fig. 1. Conceptual illustration of 3D trajectory tracking with a quadrotor

I. Introduction

Autonomous aerial vehicles such as quadrotors have become indispensable platforms for research, inspection, and delivery applications due to their high maneuverability and ability to hover, take off, and land vertically. However, their nonlinear and underactuated dynamics pose significant challenges for control, particularly when precise trajectory tracking is required in constrained environments. In this project, we develop a Nonlinear Model Predictive Controller (NMPC) for a quadrotor to track a time-parameterized three-dimensional (3D) trajectory while respecting physical constraints on both the actuators and the vehicle states.

The NMPC framework optimizes future control inputs over a finite prediction horizon by explicitly considering the full nonlinear dynamics of the system, as well as input and state limits. This approach enables real-time adaptation to changing conditions and disturbances, offering superior performance compared to linear or fixed-gain control schemes. Given a desired reference trajectory $z_{des} = \left(r_{des}(t)^T \quad \psi_{des}(t)^T\right)^T$, consisting of position and yaw profiles, the controller aims to minimize tracking error while maintaining stability and feasibility. The desired closed-loop behavior is illustrated conceptually in 1, where the quadrotor follows a 3D trajectory within a restricted airspace. To realize this objective, the project involves several key components: accurate modeling of the quadrotor's rigid-body dynamics using the Newton–Euler formalism, formulation of the optimal control problem (OCP) with appropriate state and input constraints, and implementation of the NMPC solver using acados, a high-performance framework for embedded optimization. The quadrotor model includes translational and rotational dynamics, actuator saturation, and orientation kinematics represented in both Euler angles and quaternions to avoid singularities during aggressive maneuvers.

The NMPC is implemented and tested in a PyBullet simulation environment, which provides a realistic physics engine for visualization and disturbance injection. Benchmark trajectories, such as a straight line, circular path, and 3D lemniscate, are used to evaluate performance in terms of tracking accuracy, robustness, and computational efficiency. The final controller is expected to achieve smooth and stable trajectory tracking with minimal steady-state error, while satisfying all physical constraints.

Through this project, we demonstrate the integration of nonlinear control theory, real-time optimization, and simulation-based validation for advanced quadrotor control. The work provides a foundation for future extensions such as disturbance estimation, fault-tolerant control, and experimental validation on hardware platforms.

II. Dynamics

Modeling the Dynamics of the drone is essential to estimate the state for subsequent control actions. The controller leverages this knowledge of the future states to determine the current controls. The state vector is composed of

$$X = \begin{pmatrix} r^T & \dot{r}^T & q^T & \omega^T \end{pmatrix}^T \tag{1}$$
$$X \in \mathbb{R}^{13}$$

where r is the position of the system in meters, $r = \begin{pmatrix} x & y & z \end{pmatrix}^T$, $\dot{r}$ is the velocity of the quadrotor $\dot{r} = \begin{pmatrix} \dot{x} & \dot{y} & \dot{z} \end{pmatrix}^T$. q is the attitude of the drone represented by quaternions $q = \begin{pmatrix} q_w & q_x & q_y & q_z \end{pmatrix}^T$ quaternions are used to avoid gimble lock of Euler angles. ω is the angular velocity of the system exptessed by $\omega = \begin{pmatrix} \omega_x & \omega_x & \omega_x \end{pmatrix}^T$ and represents the angular velocity about the three principal axis of the quadrotor. to control the robot the virtual inputs

$$U = \begin{pmatrix} f_z & \tau_x & \tau_y & \tau_z \end{pmatrix}^T \tag{2}$$
$$U \in \mathbb{R}^4$$

for numerical stability. The rotor speeds can be determined using the relationship

$$\begin{pmatrix} f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \dfrac{k_m}{k_f} & -\dfrac{k_m}{k_f} & \dfrac{k_m}{k_f} & -\dfrac{k_m}{k_f} \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \tag{3}$$
$$F_i = k_f \omega_i^2$$

where ω_i is the rotational velocity of each propellor.f_z is the The dynamics of the system

$$\dot{x} = f(x, u) \tag{4}$$

can be expressed by

$$f(x, u) = \begin{pmatrix} \dot{r} \\ \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \dfrac{1}{m} R_b^a \begin{pmatrix} 0 \\ 0 \\ f_z \end{pmatrix} \\ \dfrac{1}{2} \Omega(\omega) q \\ -\mathcal{I}^{-1} [\omega] \mathcal{I} \omega + \mathcal{I}^{-1} \tau \end{pmatrix} \tag{5}$$

where $\mathcal{I}$ is the rotational inertia, m the mass of the quadrotor, R_b^a is the rotation matrix from the body frame to the world frame

$$\begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) \end{pmatrix}$$

this rotation matrix holds if $\parallel q \parallel = 1$, where $\Omega(\omega)$ is

$$\Omega(\omega) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix}$$

The dynamics of the system were incorporated with CasADI, and verified using the CasADI integrator during free fall, pure thrust, angular acceleration about each principal axis.

III. NMPC Formulation

Using Runge Kutta forth order approximation the continuous dynamics were discretized to

$$x_{k+1} = f_d(x_k, u_k), \quad k = 0, \ldots, N-1$$

to be used used in the NMPC Formulation

$$\begin{aligned} \underset{x_i, u_i}{min} \quad l_N(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i, i) \\ s.t. \; x_0 = \hat{x}(t_k), \\ x_{i+1} = f_d(x_i, u_i), \quad i = 0, \ldots, N-1, \\ g(x_i, u_i) \leq 0, \\ h(x_i, u_i) = 0, \\ x_i \in \mathcal{X}, \; u_i \in \mathcal{U} \end{aligned} \tag{6}$$

where

$$\begin{aligned} l(x, u, t) = \parallel r - r_{des} \parallel_{Q_r}^2 + \parallel \dot{r} - \dot{r}_{des} \parallel_{Q_v}^2 + \parallel q - q_{des} \parallel_{Q_q}^2 \\ + \parallel \omega \parallel_{Q_\omega}^2 + \parallel u - u_0 \parallel_R^2 \end{aligned} \tag{7}$$

and terminal cost

$$l_N(x_N) = \parallel r - r_{des} \parallel_{Q_r}^2 + \parallel q - q_{des} \parallel_{Q_q}^2 \tag{8}$$

Implemented in Acados solved using Sequential Quadratic Programming (SQP) using High Performance Inter Pints method (HPIPM). utilizing the Acados optimized cost function Linear Least Squares formulation

$$l(x, u) = \parallel V_x x + V_u u - y_{ref} \parallel_W^2 \tag{9}$$

$$m(x, u) = \frac{1}{2} \parallel V_x^e x - y_{ref}^e \parallel_{W^e}^2 \tag{10}$$

y is the optimization variable defined by

$$y = V_x x + V_u u$$

and

$$y_e = V_x^e x$$

where V_x, V_u, V_x^e are selection matrices that determine what states are in the optimization domain. For the formulation above $V_x = \begin{pmatrix} I_{13 \text{ x } 13} \\ 0_{4 \text{ x } 13} \end{pmatrix}$, $V_u = \begin{pmatrix} 0_{13 \text{ x } 4} \\ I_4 \end{pmatrix}$, and $V_x^e = \begin{pmatrix} I_{3 \text{ x } 3} & 0_{3 \text{ x } 10} \\ 0_{4 \text{ x } 6} & I_{4 \text{ x } 4} & 0_{4 \text{ x } 3} \end{pmatrix}$. The weight matrix W is comprised of the block diagonal of each individual weight matrices. To ensure that the controller produces a feasible set of controls the constraints $-u_{min} \leq u \leq u_{max}$ where u_{max} is derived from the rotor limits.

A. Tuning

The NMPC was tuned to achieve a balance between tracking performance, stability, and constraint satisfaction. The process involved adjusting key parameters, including the prediction horizon, control horizon, and weighting matrices within the cost function, to refine the trade-off between trajectory accuracy and control effort. Increasing the prediction horizon was found to enhance prediction accuracy at the expense of higher computational demand, while modifications to the control weights influenced actuator smoothness and responsiveness. The parameters were iteratively refined through simulation until the desired closed-loop performance and robustness were obtained.

IV. Trajectory Results

A. Line

To evaluate the tracking performance of the controller, a quintic trajectory was generated from the origin to the point (1, 1, 1), with the corresponding velocity defined by the derivative quartic polynomial. The attitude and its higher-order derivatives were held constant throughout the test. By design, the controller prioritized position tracking over attitude regulation, resulting in limited orientation adjustments—a necessary compromise for following trajectories of this form. The quadrotor achieved a positional RMSE of (1.7567e-02, 1.7625e-02, 1.2512e-02), demonstrating high positional accuracy and effective trajectory tracking performance.

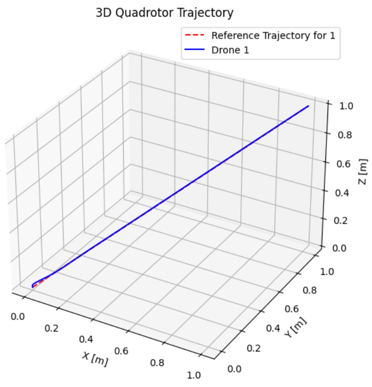

Fig. 2. Quintic trajectory from (0,0,0) to (1,1,1). The quadrotor followed the trajectory with minimal deviation. The gains were set to be significantly more than the actuation cost forcing it to follow the trajectory closely.

B. Circle

To generate a smooth and dynamically feasible circular trajectory, a parametric representation was defined as

$$r(\beta) = \big(\cos(\beta)\ \sin(\beta)\ z\ \ -\sin(\beta)\dot{\beta}\ \cos(\beta)\dot{\beta}\ 0\big)\,,$$

where the parameter β represents the angular position along the desired path. A quintic polynomial trajectory was then constructed over the interval $[0, 2\pi]$, ensuring continuity in position, velocity, and acceleration. This formulation guarantees that both the initial and final velocities are zero, resulting in a smooth, closed-loop motion suitable for stable tracking and control analysis. The desired attitude for this test was set equal to the initial state, while maintaining the same weighting parameters as in previous trials. This configuration ensured that the controller primarily prioritized convergence of the position states, while simultaneously penalizing unnecessary deviations in the attitude. By doing so, the control effort was directed toward achieving accurate positional tracking without inducing excessive or unwarranted rotational motion. The quadrotor started from rest, ascended to the desired height within one second, and then traced a circular trajectory of radius 1 meter for a duration of ten seconds. The system demonstrated strong tracking performance, achieving a positional RMSE of (0.0732, 0.068, 0.034), indicating precise and stable trajectory following throughout the maneuver.

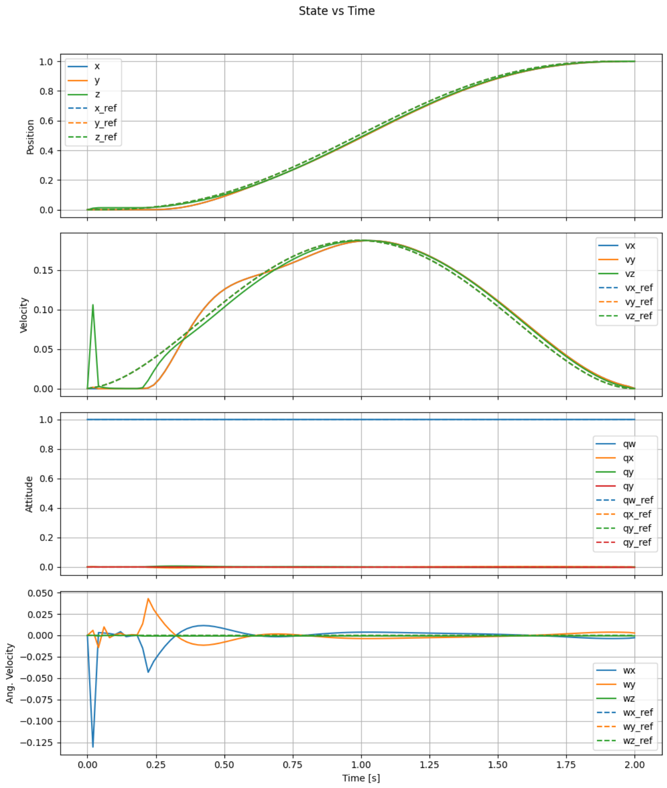

Fig. 3. The position closely follows the desired trajectory, demonstrating accurate tracking performance. A minor deviation is observed in the velocity plot, caused by the model briefly clipping into the floor and exhibiting a delayed velocity response due to its lower cost weighting relative to position. The angular velocity remains near zero throughout the motion, reflecting the strong penalization applied to rotational dynamics in the cost function.

C. Disturbance Rejection

To evaluate the disturbance rejection capability of the controller, an external force of 0.008 mN was applied to the quadrotor body. As shown in Figure 6, the quadrotor was able to track the desired trajectory with minimal positional deviation, demonstrating reasonable robustness to small disturbances. However, the state responses in Figure 7 reveal that the drone began to experience significant attitude instability, eventually flipping over and approaching a crash condition. This behavior became more pronounced as the disturbance magnitude increased, with the quadrotor losing stability almost immediately under stronger external forces.

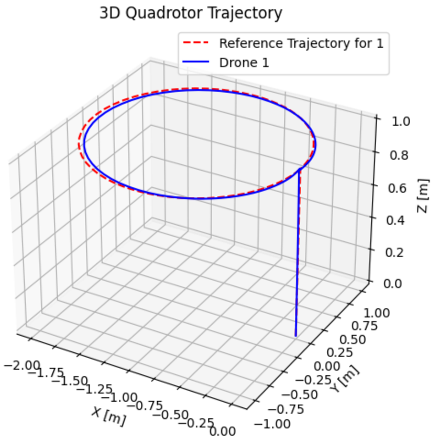

Fig. 4. The quadrotor started from rest and moved to a desired height in 1 second then traced a circle or radius 1 for 10 seconds. The quadrotor followed the trajectory very well as it had a positional RMSE of (0.0732, 0.068, 0.034).

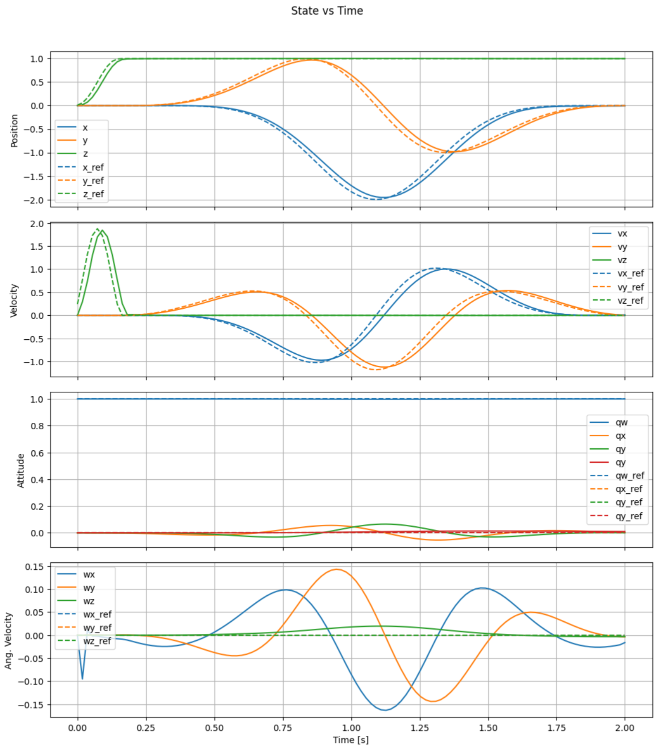

Fig. 5. The position closely follows the desired trajectory, demonstrating accurate tracking performance. The angular velocity remains near zero for most of the motion, reflecting the strong penalization of rotational dynamics in the cost function. However, small variations in angular velocity occur as the quadrotor pitches appropriately to follow the circular trajectory.

D. Solve Performance

The NMPC implementation achieved an average solve time of 1.42 ms (approximately 704 Hz), while utilizing the SPI-RTI solver with model approximations further reduced the

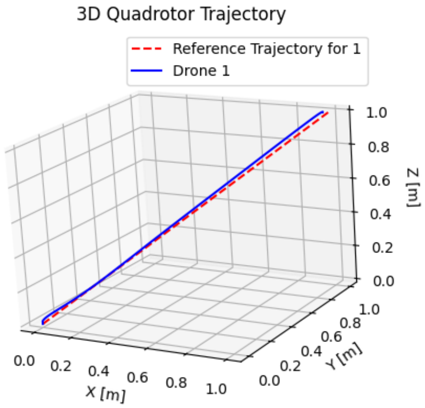

Fig. 6. Following the same quintic trajectory from (0,0,0) to (1,1,1) wind was added to determine the disturbance rejection of the controller . The quadrotor followed the trajectory with and was only pushed off the trajectory minimally.

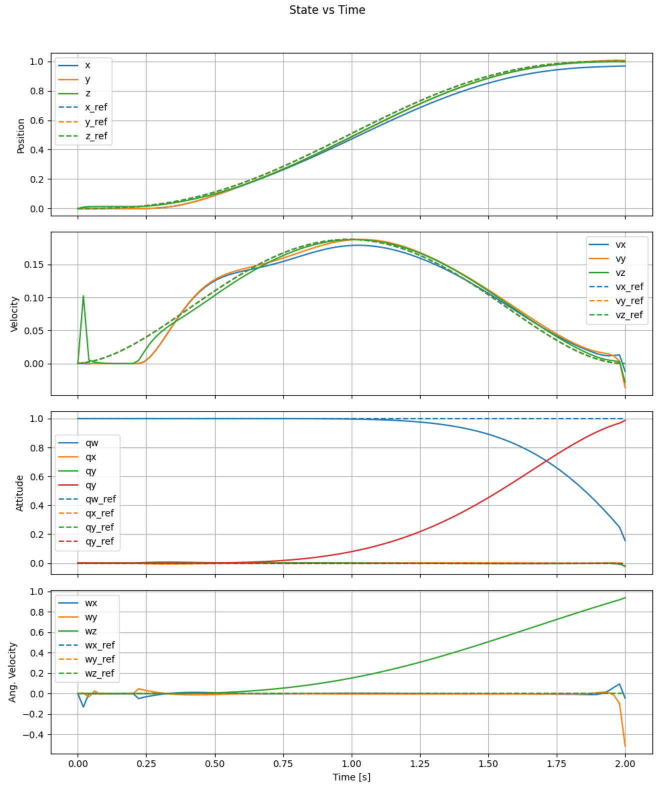

Fig. 7. The position closely follows the desired trajectory, with the velocity exhibiting a slight delay in tracking response. The orientation axes indicate that the drone experienced significant body torques, suggesting instability that could lead to a flip and potential crash.

average solve time to 0.44 ms (approximately 2273 Hz). Both configurations operated well within real-time computational limits, demonstrating the efficiency and suitability of the controller for high-frequency control applications

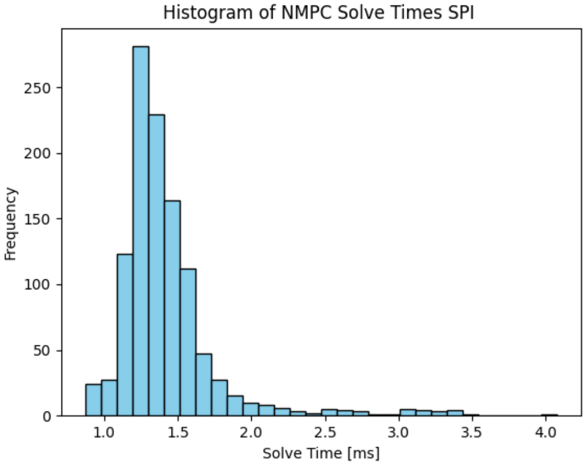

Fig. 8. The SPI HPIPM solver achieved a average solve tome of 1.421 ms with a standard deviation of 0.354 ms. The solver was very grouped about the mean with a tail of longer solve times.

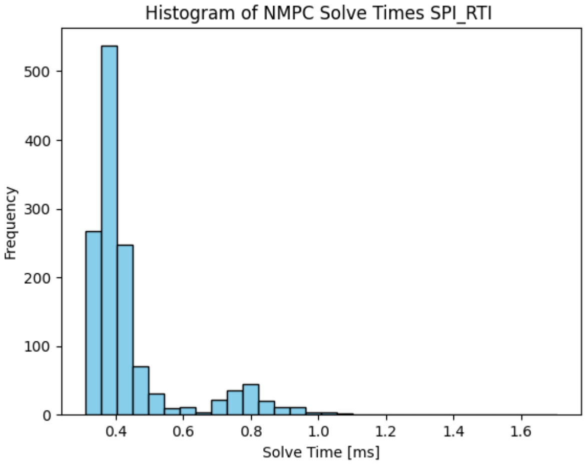

Fig. 9. The SPI-RTI HPIPM solver achieved a average solve tome of 0.44 ms with a standard deviation of 0.149 ms. The solver was very grouped about the mean with a tail of longer solve times. It is faster than the pure SPI solver by leveraging approximations to allow for faster solving times while sacrificing accuracy.

E. Failure Cases

Several failure cases were observed during the controller's operation. The first occurred when the reference trajectory was infeasible; in such cases, the drone was unable to recover once it fell behind the desired path, as illustrated in Figure 10. The second failure mode arose when external disturbances exceeded the controller's compensation capability. It was determined that, under stricter gain settings, the maximum tolerable disturbance was approximately 0.008 mN, whereas more relaxed gains further reduced the system's robustness, leading to instability and eventual crashes.

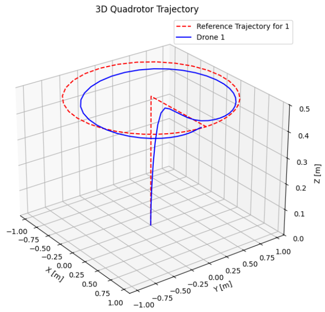

Fig. 10. The trajectory was determined to be infeasible, as it required dynamics beyond the drone's physical capabilities. Consequently, the controller was unable to maintain the prescribed path and instead truncated portions of the trajectory to remain within feasible control limits.

V. Conclusion

A Nonlinear Model Predictive Controller (NMPC) was designed and implemented to achieve precise trajectory tracking for a quadrotor system. The controller successfully followed both smooth and dynamic trajectories while maintaining stability under nominal conditions. Through systematic tuning of the weighting matrices and horizon parameters, the NMPC demonstrated fast solve times well within real-time constraints and achieved low tracking error across multiple test cases.

While the controller exhibited strong performance for feasible trajectories, limitations were observed when tracking highly aggressive or infeasible paths and when subjected to large external disturbances. These challenges highlight the importance of accurate system modeling and constraint design for robust real-world performance. Future work will focus on improving disturbance rejection and incorporating adaptive or learning-based methods to enhance robustness and generalization in more uncertain environments.

Appendix

A. GitHub Release

GitHub Release Link