Trajectory-Invariant Nonlinear Model Predictive Control for Quadrotor Tracking: Comparison with Optimized PD Control

Neelkumar Ahir, Jyotindra Narayan *Member, IEEE*, and Garima Bhandari *Member, IEEE RAS*

Abstract— **This paper presents a nonlinear model predictive control (NMPC) framework for quadrotor trajectory tracking under nonlinear dynamics and aerodynamic disturbances. Unlike conventional proportional–derivative (PD) controllers that require trajectory-dependent gain retuning, the proposed NMPC employs a single set of weighting matrices to track multiple three-dimensional trajectories. The proposed controller is evaluated on helical and Lissajous trajectories and compared against an optimized PD controller. Simulation results show that the NMPC framework achieves improved tracking accuracy, reduced overshoot, and smoother control effort across varying trajectory complexities. Quantitatively, NMPC reduces the Integral of Squared Error (ISE) by 53–99.6% and the Integral of Time Absolute Error (ITAE) by 82.5–98.5% compared to the optimized PD controller, while maintaining robust performance without retuning. These results demonstrate the suitability of fixed-tuning NMPC for agile quadrotor trajectory tracking.**

Nonlinear model predictive control, quadrotor trajectory tracking, optimized PD control, fixed-tuning NMPC, UAV control.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), particularly quadrotors, have emerged as highly maneuverable aerial platforms with widespread applicability in domains such as aerial surveillance, environmental monitoring, and infrastructure inspection. Their ability to hover, perform agile maneuvers, and operate in constrained environments has made them indispensable in both civilian and industrial settings [1], [2]. However, achieving precise and reliable trajectory tracking for quadrotors remains a persistent challenge due to the system's inherently nonlinear, underactuated dynamics and its susceptibility to external disturbances such as wind and aerodynamic coupling [3]–[5].

Conventional control strategies, particularly Proportional–Derivative (PD) controllers, are widely used in quadrotor systems due to their simplicity, ease of implementation, and low computational overhead [6], [7]. Despite these advantages, conventional PD control often requires manual re-tuning of gain parameters when operating conditions or desired trajectories change, reducing controller adaptability and limiting performance in dynamic or unstructured environments [8]. Addressing these limitations is critical to enhancing the robustness and autonomy of quadrotor systems in real-world applications.

Neelkumar Ahir and Garima Bhandari are with the Department of Mechanical Engineering, California State University, Sacramento, CA, USA.

Jyotindra Narayan is with the Indian Institute of Technology (IIT), Patna, India.

Corresponding author: Garima Bhandari (e-mail: garima.bhandari@csus.edu).

Although advanced control methods such as Linear Quadratic Regulator (LQR), Fuzzy Logic Control (FLC), and Sliding Mode Control (SMC) have been applied to quadrotor systems, they exhibit limitations for real-time trajectory tracking. LQR relies on linearized dynamics and is less effective for highly nonlinear and underactuated quadrotors [9], [10]. FLC lacks a systematic tuning procedure and scales poorly for multi-input UAV systems [11], [12], while SMC, despite its robustness, suffers from chattering and potential actuator wear [13]. These limitations motivate the adoption of Model Predictive Control (MPC), which enables constraint-aware, high-performance tracking in dynamic environments [14], [15].NMPC is attractive because constrained MPC has a well-established theoretical foundation for stability, feasibility, and optimality under appropriate assumptions, and prior quadrotor studies have demonstrated its practical effectiveness [14], [15].

Despite advances in quadrotor control, many approaches still require trajectory-dependent retuning for robust tracking. The main contributions of this paper are:

1) A fixed-tuning NMPC framework for quadrotor trajectory tracking using a single set of weighting matrices across all trajectories with a reduced state-weighting design.
2) A comparative study between optimized PD control and NMPC, highlighting the superior robustness of NMPC without trajectory-dependent retuning.

This paper is organized as follows: Section II provides the dynamics of the quadrotor. Section III covers both optimized PD and proposed NMPC control methodologies in detail. The results of the simulation testing are discussed in Section IV. Finally, Section V concludes the paper.

II. DYNAMIC MODELING

The development and evaluation of robust quadrotor trajectory control strategies require a rigorous dynamic model that accurately represents system behavior across varying operating conditions [2], [16]. This section presents a quadrotor dynamic model derived from the Newton–Euler equations, capturing both translational and rotational motion in six degrees of freedom (6-DOF). A standard quadrotor consists of a rigid body and four propellers arranged in a cross configuration, as shown in Fig. 1. The propellers generate lift (thrust) and control moments, enabling coupled translational and rotational maneuvers [17]. The inertial frame is denoted by $\{A\}$ and the body-fixed frame by $\{B\}$, located at the quadrotor center of gravity (CG).

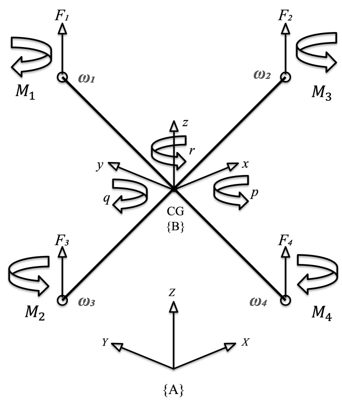

Fig. 1. Schematic of quadrotor.

In the inertial frame, the quadrotor position is given by $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$, and the attitude is represented by the Euler angles $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$. The translational dynamics in the x, y, and z directions are expressed as

$$\mathbf{F} + m(\mathbf{v} \times \boldsymbol{\omega}_b) + {}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R}^T \mathbf{W} - \mathbf{C}_d \mathbf{v} = m\dot{\mathbf{v}}. \tag{1}$$

Here, $\mathbf{F} = \begin{bmatrix} 0 & 0 & F_Z \end{bmatrix}^T$ is the total thrust force along the body-fixed frame z-axis, where

$$F_Z = F_1 + F_2 + F_3 + F_4.$$

The quadrotor mass is denoted by m, and

$$\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$$

is the linear velocity vector expressed in the body frame. The body-fixed frame angular velocity is

$$\boldsymbol{\omega}_b = \begin{bmatrix} r & p & s \end{bmatrix}^T,$$

where r, p, and s are the roll, pitch, and yaw rates.

The rotation matrix ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R}$, which maps vectors from the inertial frame to the body-fixed frame, is defined as

$${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R} = \begin{bmatrix} \mathrm{c}\theta\,\mathrm{c}\psi & \mathrm{s}\phi\,\mathrm{s}\theta\,\mathrm{c}\psi - \mathrm{c}\phi\,\mathrm{s}\psi & \mathrm{c}\phi\,\mathrm{s}\theta\,\mathrm{c}\psi - \mathrm{s}\phi\,\mathrm{s}\psi \\ \mathrm{c}\theta\,\mathrm{s}\psi & \mathrm{s}\phi\,\mathrm{s}\theta\,\mathrm{s}\psi + \mathrm{c}\phi\,\mathrm{c}\psi & \mathrm{c}\phi\,\mathrm{s}\theta\,\mathrm{s}\psi - \mathrm{s}\phi\,\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\,\mathrm{c}\theta & \mathrm{c}\phi\,\mathrm{c}\theta \end{bmatrix},$$

where ϕ, θ, and ψ denote the roll, pitch, and yaw angles, respectively. The gravitational force in the inertial frame is

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T,$$

and the aerodynamic drag matrix is

$$\mathbf{C}_d = \mathrm{diag}(C_{dx}, C_{dy}, C_{dz}).$$

The rotational dynamics in the body-fixed frame are given by

$$\mathbf{T} + \boldsymbol{\omega_b} \times \mathbf{I}\boldsymbol{\omega_b} = \mathbf{I}\dot{\boldsymbol{\omega}}_{\boldsymbol{b}}, \tag{2}$$

where $\mathbf{T} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T$ is the control torque vector for roll, pitch, and yaw motion, and $\mathbf{I} = \mathrm{diag}(I_x, I_y, I_z)$ is the inertia matrix.

III. Control Methodology

Building on the quadrotor dynamic model, this section presents the control methodologies used for trajectory tracking. The classical PD controller is described first to highlight its reliance on trajectory-dependent tuning, followed by the proposed NMPC framework, which alleviates this limitation by employing a single fixed set of control parameters.

A. Optimized PD Control

A Proportional–Derivative (PD) control scheme is employed to achieve accurate trajectory tracking of the quadrotor. The controller utilizes position and velocity tracking errors to generate corrective control inputs that ensure stability and tracking precision. As shown in Figure 2, the reference position $(X_{\mathrm{ref}}, Y_{\mathrm{ref}}, Z_{\mathrm{ref}})$ and yaw angle ψ_{ref} are directly supplied to the control architecture.

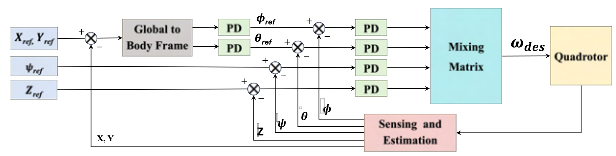

Fig. 2. Schematic of PD controller

In contrast, the roll and pitch reference angles are generated internally based on lateral position errors according to

$$\phi_{\mathrm{ref}} = k_{p,Y}\, Y_e + k_{d,Y}\, \dot{Y}_e \tag{3}$$

$$\theta_{\mathrm{ref}} = k_{p,X}\, X_e + k_{d,X}\, \dot{X}_e \tag{4}$$

where $k_{p,Y}$, $k_{d,Y}$, $k_{p,X}$, and $k_{d,X}$ are proportional and derivative gains. The lateral position errors are defined in the body-fixed frame relative to the reference trajectory as

$$X_e = (X_{\mathrm{ref}} - X)\cos\psi + (Y_{\mathrm{ref}} - Y)\sin\psi \tag{5}$$

$$Y_e = -(X_{\mathrm{ref}} - X)\sin\psi + (Y_{\mathrm{ref}} - Y)\cos\psi \tag{6}$$

The resulting position and orientation errors are processed by PD controllers to generate control commands. These commands are mapped through a mixing matrix to compute the desired propeller angular velocities $\boldsymbol{\omega}_{\mathrm{des}}$, which are applied to the quadrotor for trajectory tracking:

$$\boldsymbol{\omega}_{\mathrm{des}}^{\mathsf{T}} = \begin{bmatrix} \omega_1^{\mathrm{des}} \\ \omega_2^{\mathrm{des}} \\ \omega_3^{\mathrm{des}} \\ \omega_4^{\mathrm{des}} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_{\mathrm{lift}} \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix} \tag{7}$$

Here, $\omega_h = \sqrt{\frac{mg}{4k_f}}$ denotes the hover angular velocity. Here k_f is propeller thrust coefficient.The incremental control terms governing altitude and attitude regulation are defined as

$$\Delta\omega_{\mathrm{lift}} = k_{p,Z}(Z_{\mathrm{ref}} - Z) + k_{d,Z}(\dot{Z}_{\mathrm{ref}} - \dot{Z}) \tag{8}$$

$$\Delta\omega_\phi = k_{p,\phi}(\phi_{\mathrm{ref}} - \phi) + k_{d,\phi}(\dot{\phi}_{\mathrm{ref}} - \dot{\phi}) \tag{9}$$

$$\Delta\omega_\theta = k_{p,\theta}(\theta_{\mathrm{ref}} - \theta) + k_{d,\theta}(\dot{\theta}_{\mathrm{ref}} - \dot{\theta}) \tag{10}$$

$$\Delta\omega_\psi = k_{p,\psi}(\psi_{\mathrm{ref}} - \psi) + k_{d,\psi}(\dot{\psi}_{\mathrm{ref}} - \dot{\psi}) \tag{11}$$

where the proportional and derivative gains correspond to vertical motion, roll, pitch, and yaw dynamics, respectively.

In total, six PD controllers are employed for trajectory tracking, resulting in twelve gains that require tuning. Manual tuning through trial-and-error is inefficient and time-consuming. To address this, the controller gains are optimized using the procedure described in Algorithm 1.

The optimization objective minimizes the cumulative squared tracking error over the trajectory:

$$\mathcal{J}_{\text{total}} = W\,\Delta t \sum_{k=0}^{N} \|\mathbf{e}[k]\|_2^2 \tag{12}$$

where $\mathbf{e}[k]$ represents the position and orientation tracking error at time step k over a 80 s horizon.

The gain vector is defined as

$$\mathbf{K} = \begin{bmatrix} K_{p,Z}, K_{p,\phi}, K_{d,Z}, K_{d,\phi}, K_{p,Y}, K_{p,\theta}, \\ K_{d,Y}, K_{d,\theta}, K_{p,X}, K_{p,\psi}, K_{d,X}, K_{d,\psi} \end{bmatrix} \tag{13}$$

Algorithm 1 FMINCON-Based PD Gain Optimization

Require: Δt, N, x_0, $r[0{:}N]$, K_0, $K_{\min}{=}0$, $K_{\max}{=}1000$, $W{=}1000$, $\omega_{\max}{=}1500$ rad/s

Ensure: Optimal gains $\mathbf{K}^\star$

```
Define: e[k] = r[k] − x[k],  u[k] = K_p e[k] + K_d ė[k],  ẋ = f(x, u)
function OBJECTIVE(K)
    x ← x_0,  J ← 0
    for k = 0 to N do
        J ← J + ‖e[k]‖_2^2;  x ← F(x, u; Δt)
    end for
    return W Δt J
end function
K* = fmincon(Objective, K_0, lb=K_min, ub=K_max, ω_n(K) ≤ ω_max)
```

The optimization seeks the gain vector $\mathbf{K}^\star$ that minimizes $\mathcal{J}_{\text{total}}$ while enforcing the constraint $\omega_{\max} < 1500$ rad/s. Manually tuned gains are used as the initial guess, and the optimization is performed offline using fmincon, a gradient-based constrained nonlinear optimization solver. For offline PD-gain optimization, the continuous-time quadrotor model is numerically propagated in discrete time with sampling interval Δt over the reference trajectory. The resulting gains shown in Table I remain fixed during flight execution in simulation.

B. NMPC

Model Predictive Control (MPC) has gained significant popularity in recent decades, primarily due to its effectiveness in managing multi-input–multi-output (MIMO) systems while explicitly considering constraints on both control inputs and system states [18].

Figure 3 illustrates the nonlinear MPC (NMPC) framework used for quadrotor trajectory tracking. The reference trajectory is compared with the estimated quadrotor states to compute tracking errors, which are processed by the NMPC optimizer together with a prediction model to generate a control sequence. At each sampling instant, only the first control action is applied to the quadrotor, and the updated states are fed back to close the loop, enabling accurate trajectory tracking.

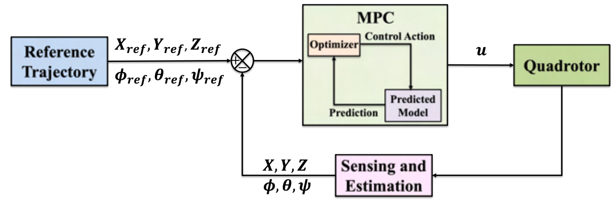

Fig. 3. Schematic of NMPC controller

The control problem is formulated as the minimization of a weighted cost function that balances trajectory tracking accuracy and control effort. The cost function J is defined as

$$\mathbf{J} = \sum_{k=0}^{N-1} \left(\mathbf{e}_{\mathrm{p},k}^{\mathsf{T}} \mathbf{Q}_{\mathrm{p}} \mathbf{e}_{\mathrm{p},k} + \mathbf{e}_{\mathrm{o},k}^{\mathsf{T}} \mathbf{Q}_{\mathrm{o}} \mathbf{e}_{\mathrm{o},k} + \Delta\mathbf{u}_k^{\mathsf{T}} \mathbf{R}\, \Delta\mathbf{u}_k \right) \tag{14}$$

where N denotes the prediction-horizon length; $\mathbf{e}_{\mathrm{p},k}$ is the position and velocity tracking error at the k^{th} prediction step with weighting matrix $\mathbf{Q}_{\mathrm{p}} \in \mathbb{R}^{6\times6}$; $\mathbf{e}_{\mathrm{o},k}$ is the orientation tracking error at the k^{th} prediction step with weighting matrix $\mathbf{Q}_{\mathrm{o}} \in \mathbb{R}^{3\times3}$; $\Delta\mathbf{u}_k$ is the control-input deviation; and $\mathbf{R} \in \mathbb{R}^{4\times4}$ is the penalty to maintain the optimal control effort.

The position and orientation tracking errors at the k^{th} prediction step are computed as

$$\mathbf{e}_{\mathrm{p},k} {=} \big[\, X_{\text{ref}}\ Y_{\text{ref}}\ Z_{\text{ref}}\ \dot{X}_{\text{ref}}\ \dot{Y}_{\text{ref}}\ \dot{Z}_{\text{ref}} \,\big]^{\mathsf{T}} - \big[\, X\ Y\ Z\ \dot{X}\ \dot{Y}\ \dot{Z} \,\big]_k^{\mathsf{T}}$$
$$\mathbf{e}_{\mathrm{o},k} {=} \big[\, \phi_{\text{ref}}\ \theta_{\text{ref}}\ \psi_{\text{ref}} \,\big]^{\mathsf{T}} - \big[\, \phi\ \theta\ \psi \,\big]_k^{\mathsf{T}}$$

where $X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}}$ denote the reference positions along the x, y, and z direction, respectively, and $\dot{X}_{\text{ref}}, \dot{Y}_{\text{ref}}, \dot{Z}_{\text{ref}}$ denote the corresponding reference linear velocities. The control-input deviation is defined as,

$$\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_h \tag{15}$$

where $\mathbf{u}_k$ is the predicted control input at the k^{th} step and $\mathbf{u}_h$ is the hover input. The predicted input is parameterized over a control horizon $N_c \le N$ as,

$$\mathbf{u}_k = \begin{cases} \begin{bmatrix} F_z & T_x & T_y & T_z \end{bmatrix}_k^{\mathsf{T}}, & 0 \le k < N_c - 1, \\ \begin{bmatrix} F_z & T_x & T_y & T_z \end{bmatrix}_{N_c-1}^{\mathsf{T}}, & N_c - 1 \le k \le N - 1 \end{cases}$$

and the hover input is given by,

$$\mathbf{u}_h = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \tag{16}$$

The optimization is constrained by: the initial condition, discrete dynamics, and input bounds:

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{17}$$
$$\mathbf{x}_{k+1} = f(\mathbf{x}_k,\, \mathbf{u}_k). \tag{18}$$
$$u_{\min} \le u_k \le u_{\max}, \qquad k = 0, \ldots, N_{c-1} \tag{19}$$

TABLE I

PD CONTROL PARAMETERS

Parameters	Helical	Lissajous	Parameters	Helical	Lissajous
$K_{p,x}$	0.1200	0.1300	$K_{p,\phi}$	380.8515	380.8515
$K_{d,x}$	0.3004	0.15004	$K_{d,\phi}$	150.4867	150.4867
$K_{p,y}$	0.1000	0.1000	$K_{p,\theta}$	400.4488	400.4488
$K_{d,y}$	0.7000	0.1000	$K_{d,\theta}$	100.0240	100.1353
$K_{p,z}$	193.0297	163.0297	$K_{p,\psi}$	100.1353	100.1353
$K_{d,z}$	120.0335	120.0335	$K_{d,\psi}$	280.7893	280.7893

where, $\mathbf{x_0}$ is the initial state vector, $\mathbf{x_{k+1}}$ is the discrete dynamics, $\mathbf{u}_{\min}$, and $\mathbf{u}_{\max}$ represent the minimum and maximum allowable inputs, respectively.

The state vector is ordered as

$$\mathbf{x}(t) = [\, x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ \phi\ \theta\ \psi\ p\ q\ r \,]^T$$

The nonlinear model serves as the prediction model for NMPC. Since the quadrotor is underactuated (six degrees of freedom with four control inputs), the primary regulated states are selected through the weighting matrix Q, while constraints are applied to other states to limit their variation, which is important for NMPC effectiveness [18]. Additionally, control input weights in R are selected lower than the tracking weights to keep the system near the hovering equilibrium. This study uses the CasADi framework for quadrotor trajectory tracking. Algorithm 2 summarizes the nonlinear MPC procedure used in this work.

Algorithm 2 Non-linear MPC

Require: T_s, horizons N, $N_c \le N$, hover input u_{hov}, bounds on u

$\hat{x} \leftarrow \hat{x}_0$
while controller active **do**
  Build references $(r_i^{\text{pos}}, r_i^{\psi}) = \text{Ref}(t_k, i)$ for $i = 0{:}N$
  Solve OCP: *minimize* J
  s.t. $X_0 = \hat{x}$, bounds on U_j
  for $i = 0$ to $N-1$ **do**
    if $i < N_c - 1$ **then**
      $X_{i+1} = F_{\text{RK4}}(X_i, U_i; T_s)$
    else
      $X_{i+1} = F_{\text{RK4}}(X_i, U_{N_c-1}; T_s)$
    end if
  end for
  Apply $u_k \leftarrow U_0^\star$ for T_s; update $\hat{x} \leftarrow \textsc{Observer}(\cdot)$; $t_{k+1} \leftarrow t_k + T_s$
end while

In our implementation, the control parameters used in the NMPC formulation are summarized in Table II. Weights are assigned to the six translational states $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$ and the three Euler angles $[\phi, \theta, \psi]$, while angular velocity terms are intentionally excluded from the cost formulation. This choice reduces computational complexity and improves solver speed, supporting real-time implementation. Meanwhile, hard constraints (see (19)) maintain actuator-safe operation by keeping angular rates within feasible limits, preserving safety and feasibility. This design supports agile maneuvering without sacrificing robustness and is suitable for computationally constrained platforms such as onboard flight controllers [19].

TABLE II

NMPC CONTROL PARAMETERS

Parameter	Value
$\mathbf{Q}_p$	diag(16, 16, 10, 2, 1.2, 0.09)
$\mathbf{Q}_o$	diag(3.0, 3.0, 3.0)
$\mathbf{R}$	diag(0.1, 0.1, 0.1, 0.1)
T_s	0.01 s
N	50 steps
N_c	20 steps
$\mathbf{x}_0$	$[0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]^T$
$\mathbf{u}_{min}$	$[0 \quad -2.00 \quad -2.00 \quad 1.00]^T$
$\mathbf{u}_{max}$	$[20.93 \quad 2.00 \quad 2.00 \quad 1.00]^T$

IV. RESULTS & DISCUSSION

The proposed NMPC controller was compared against a classical optimized PD controller in a simulation environment. The quadrotor platform parameters are summarized in Table III. Tracking performance was evaluated over an 80 s simulation using two reference trajectories of increasing complexity: a helical path and a Lissajous curve.

TABLE III

QUADROTOR PARAMETERS

Description	Symbol	Value	Units
Inertia (roll)	I_x	0.0153	kg·m^2
Inertia (pitch)	I_y	0.0153	kg·m^2
Inertia (yaw)	I_z	0.0307	kg·m^2
Mass	m	1.587	kg
Gravity	g	9.81	m/s^2
Drag coefficient	c_d	0.25	

A. Helical Trajectory Tracking

In the helical trajectory, the quadrotor must track a circular path in the xy-plane while simultaneously ascending. This trajectory is more challenging, as it demands coordination between horizontal motion and climbing. The reference equations are as follows:

$$X_{\text{ref}}(t) = 3\cos(0.01t), \qquad Y_{\text{ref}}(t) = 3\sin(0.01t),$$
$$Z_{\text{ref}}(t) = 0.5\,t \text{ m}, \qquad \psi_{\text{ref}}(t) = 0 \text{ rad}.$$

Fig. 4 presents the tracking performance of the proposed NMPC controller in comparison with the classical optimized PD controller for one complete loop of a helical trajectory. The position responses along the x, y, and z axes are shown in Figs. IV-A–4c, while the corresponding three-dimensional trajectory is illustrated in Fig. 4d.

Both controllers are able to follow the reference trajectory with stable behavior; however, noticeable differences arise in the transient and steady-state responses. As summarized in Table IV, the NMPC controller achieves markedly shorter settling times across all axes, with reductions of up to 63.2% and 37.8% along the x and y directions, respectively. In addition, NMPC substantially reduces overshoot (including complete elimination along the x-axis) and decreases steady-state tracking errors by more than 80% compared to the optimized PD controller, indicating improved tracking accuracy and transient performance throughout the maneuver.

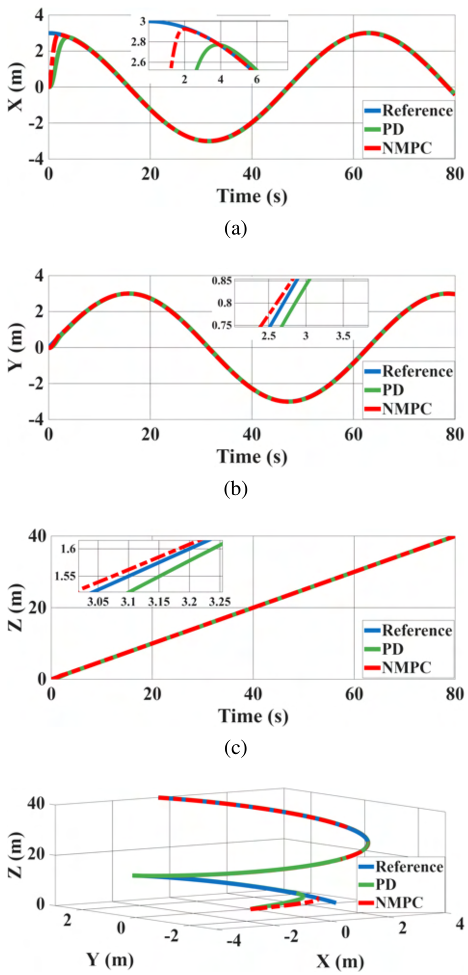

Fig. 4. Simulation results for Quadrotor traversing the reference helical trajectory in (a) x - direction (b) y - direction (c) z - direction, and (d) 3D trajectory overlay.

TABLE IV
HELICAL TRAJECTORY PERFORMANCE: NMPC VS. PD

Axis	Metric	NMPC	PD
X	Settling time (s)	2.52	6.85
	Overshoot (m)	No overshoot	0.04558
	Steady-state error (m)	0.0017	0.0467
Y	Settling time (s)	3.48	5.59
	Overshoot (m)	0.054677	0.176405
	Steady-state error (m)	0.00291	0.01576
Z	Settling time (s)	3.39	3.66
	Overshoot (m)	0.08065	0.153012
	Steady-state error (m)	0.004	0.0244

TABLE V
LISSAJOUS TRAJECTORY PERFORMANCE: NMPC VS. PD

Axis	Metric	NMPC	PD
X	Settling time (s)	8.17	5.06
	Overshoot (m)	0.17293	0.07526
	Steady-state error (m)	0.0933	0.0872584
Y	Settling time (s)	7.50	11.23
	Overshoot (m)	0.132619	0.197442
	Steady-state error (m)	0.0032	0.00576
Z	Settling time (s)	19.41	21.02
	Overshoot (m)	0.0027	0.0050
	Steady-state error (m)	0.0010	0.0011

B. Lissajous Trajectory Tracking

The Lissajous curve represents a geometrically intricate planar trajectory with repeated changes in direction and curvature. For a quadrotor, accurately tracking this path requires continuous adjustment of velocity, making it well suited for evaluating control robustness. The reference trajectory is defined as follows:

$$X_{\text{ref}}(t) = 3\sin(0.02t), \qquad Y_{\text{ref}}(t) = 3\sin(0.01t),$$
$$\psi_{\text{ref}}(t) = 0 \text{ rad}$$

To ensure smooth takeoff a fifth-order polynomial minimum-snap profile is applied in the z-direction:

$$Z_{\text{ref}}(t) = c_5t^5 + c_4t^4 + c_3t^3 + c_2t^2 + c_1t + c_0,$$

Fig. 5 illustrates the tracking performance of the proposed NMPC controller in comparison with the classical optimized PD controller for one complete cycle of the Lissajous trajectory. The position responses along the x, y, and z axes are shown in Figs. 5a–5c, while the corresponding three-dimensional trajectory is depicted in Fig. 5d.

Both controllers achieve stable trajectory tracking; however, NMPC demonstrates improved performance along the y and z axes, reducing settling time by 33.2% and 7.7%, respectively, while also yielding lower steady-state errors across all axes. The most significant steady-state error reduction is observed along the y-axis (44.4%), highlighting the enhanced tracking accuracy of NMPC for complex planar trajectories.

These results are obtained using the same NMPC tuning parameters as in the previous trajectory, whereas the optimized PD controller requires retuning for different reference trajectories. As summarized in Table VI, NMPC achieves significantly lower tracking errors for the helical trajectory, with an ISE of 5.36 versus 11.41 and an ITAE of 2.37 versus 162.36 for the optimized PD controller. A similar trend is observed for the Lissajous trajectory, where NMPC maintains low error measures despite the increased trajectory complexity. These results demonstrate that NMPC provides superior tracking accuracy and robustness across varying trajectories without requiring retuning.

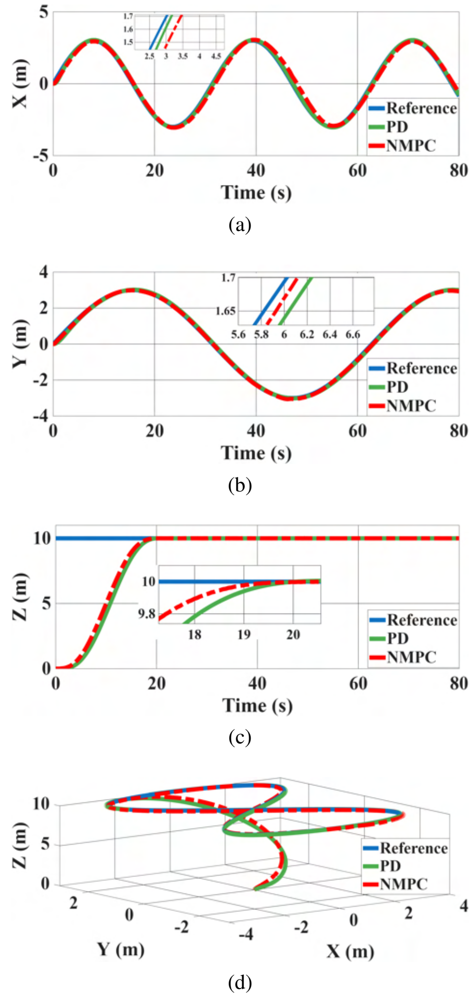

Fig. 5. Simulation results for Quadrotor traversing the reference lissajous trajectory in (a) x - direction (b) y - direction (c) z - direction, and (d) 3D trajectory overlay.

TABLE VI
ISE AND ITAE METRICS FOR NMPC AND PD CONTROLLERS

Trajectory	Metric	NMPC	PD
Helical	ISE	5.3622	11.4109
	ITAE	2.3714	162.3572
Lissajous	ISE	3.6948	864.2550
	ITAE	162.6359	927.3330

V. Conclusion

This paper presented a robust NMPC framework for three-dimensional quadrotor trajectory tracking and compared its performance with an optimized PD controller. Using a single set of tuning weights and constraints, the proposed NMPC achieved accurate tracking across trajectories of varying complexity, including helical and Lissajous paths, with reduced overshoot, faster settling time, and smaller steady-state errors. Quantitative evaluation further demonstrated significant improvements over the optimized PD controller, with ISE reductions of 53–99.6% and ITAE reductions of 82.5–98.5%. Overall, the results highlight the capability of the proposed NMPC framework to reliably execute diverse trajectories using a single parameter set, making it suitable for commercial aerial platforms.

Future work will leverage the collected flight data within learning-based and reinforcement learning–augmented NMPC frameworks to adapt the weighting matrices $Q_{\rm p}, Q_{\rm o}$, and R for enhanced performance across missions and operating conditions.

References

[1] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annual Reviews in Control*, vol. 46, pp. 165–180, 2018.

[2] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[3] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2451 – 2456 vol.3, 2004.

[4] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, "Attitude control of a quadrotor with optimized pid controller," *Intelligent Control and Automation*, 2013.

[5] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, "Modelling and pid controller design for a quadrotor unmanned air vehicle," in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, vol. 1. IEEE, 2010, pp. 1–5.

[6] A. Tassanbi, A. Ali Khan, A. Kashkynbayev, E. Shehab, T. Duc Do, and M. H. Ali, "Performance evaluation of developed controllers for unmanned aerial vehicles with reference camera," *Measurement and Control*, vol. 58, no. 9, pp. 1217–1244, 2025.

[7] A. Ayala, L. Portela, F. Buarque, B. J. Fernandes, and F. Cruz, "Uav control in autonomous object-goal navigation: a systematic literature review," *Artificial Intelligence Review*, vol. 57, no. 5, p. 125, 2024.

[8] A. T. Nguyen, N. H. Nguyen, and M. L. Trinh, "Fuzzy pd control for a quadrotor with experimental results," *Results in Control and Optimization*, p. 100568, 2025.

[9] M. Okasha, J. Kralev, and M. Islam, "Design and experimental comparison of pid, lqr and mpc stabilizing controllers for parrot mambo mini-drone," *Aerospace*, vol. 9, no. 6, p. 298, 2022.

[10] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[11] D. Gautam and C. Ha, "Control of a quadrotor using a smart self-tuning fuzzy pid controller," *International Journal of Advanced Robotic Systems*, vol. 10, no. 11, p. 380, 2013.

[12] J. Yen, H. Wang, and W. C. Daugherity, "Design issues of a reinforcement-based self-learning fuzzy controller for petrochemical process control," *NASA. Johnson Space Center, North American Fuzzy Logic Processing Society (NAFIPS 1992), Volume 1*, 1992.

[13] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor uav," *ISA transactions*, vol. 53, no. 4, pp. 1350–1356, 2014.

[14] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment," *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 5057–5068, 2021.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[16] G. Bhandari, P. M. Pathak, and S. K. Saha, "Bond graph modeling and trajectory control of h-drone," in *2022 13th Asian Control Conference (ASCC)*. IEEE, 2022, pp. 2206–2211.

[17] M. Owis, S. El-Bouhy, and A. El-Badawy, "Quadrotor trajectory tracking control using non-linear model predictive control with ros implementation," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*. IEEE, 2019, pp. 243–247.

[18] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation—an introductory overview," *Advances in control: Highlights of ECC'99*, pp. 391–449, 1999.

[19] M. Kazim, H. Sim, G. Shin, H. Hwang, and K.-K. K. Kim, "Aggressive trajectory tracking for nano quadrotors using embedded nonlinear model predictive control," in *International Conference on Intelligent Autonomous Systems*. Springer, 2023, pp. 317–332.