Bi-Copter mpc

Andres Alberto Castro Peñaranda*, Md Zilan Uddin Saif†
*andres.castro.penaranda@estudiantat.upc.edu
†md.zilan.uddin.saif@estudintat.upc.edu

***Abstract*—This paper explores the dynamics and control of a bicopter utilizing the Model Predictive Control (MPC) method. The dynamic model is highly accurate and nonlinear, featuring six degrees of freedom and accounting for disturbances and model uncertainties. The control strategy is designed using MPC to follow various reference trajectories, from simple ones like circular paths to more complex helical patterns. This technique involves deriving a linearized model and applying linear MPC (linearizing around hover) and Non linear MPC method to create the optimal control sequence. Despite MPC being computationally intensive, it excels at managing different nonlinearities and constraints, such as actuator saturation and model uncertainties. The MPC parameters (control and prediction horizons) are determined through a trial-and-error process. Multiple simulation scenarios are conducted to test and assess the performance of the proposed control method within the MATLAB. The simulation results demonstrate that this control strategy is highly effective in tracking the specified reference trajectory.**

***Index Terms*—bi-copter, unmanned aerial vehicle (UAV), design, analysis, control, dynamic modeling, simulation, MPC control**

I. Introduction

In recent years, the development of unmanned aerial vehicles (UAVs) has witnessed significant advancements, with various applications ranging from surveillance to package delivery. Among the different types of UAVs, bi-copters offer unique advantage of the Bicopter is the reduced number of motors and propellers which decreases the total cost, vibration, and the power demand [1] However, managing the attitude of a bicopter presents challenges due to the intricate and nonlinear nature of its dynamics.

Attitude control is a crucial aspect in the design and operation of UAVs. Specifically, for a UAV bicopter, which merges the benefits of both helicopters and fixed-wing aircraft, maintaining control over the attitude is vital for ensuring stable flight and maneuverability. Traditional control methods, such as proportional-integral-derivative (PID) controllers, have been extensively employed for managing bicopter attitude control [2] [3].These controllers are straightforward to implement and have demonstrated effectiveness in numerous scenarios.

Model Predictive Control (MPC) is increasingly favored for its capacity to manage constraints and disturbances effectively, its predictive capabilities, straightforward tuning process, and superior performance in handling multiple variables concurrently. As a nonlinear control strategy, MPC excels in forecasting future states and errors. Its effectiveness has been demonstrated through various implementations, including those on quadcopters [4]. In that study, MPC effectively tracked the reference trajectory on the quadcopter platform using MPC technique. This paper further explores the dynamics and control of a bicopter utilizing the MPC approach.

II. Problem statement

The key aim of project implementation in the control of UAVs is to achieve precision and reliability regarding trajectory tracking when employed in applications such as surveillance and delivery. The challenge lies in developing a system that can trace a rough trajectory while ensuring stability and efficiency. The current project involves implementing a Model Predictive Control system for a bi-copter, focusing on both Linear MPC and Nonlinear MPC. It considers the problem of obtaining a smooth reference trajectory, the control of bi-copter dynamics, the comparison of the performance between LMPC and NMPC, and the possibility of adaptation to environmental changes(Wind disturbance). The effectiveness of these strategies is verified by simulations that shed light on effective bi-copter control systems.

A. *Case study description*

This case study investigates the implementation of Model Predictive Control (MPC) systems in the field of unmanned aerial vehicles (UAVs), specifically focusing on bi-copter UAVs utilized for surveillance and delivery applications. The primary objective is to develop and compare Linear Model Predictive Control (LMPC) and Nonlinear Model Predictive Control (NMPC) systems to ensure precise trajectory tracking and adaptability to dynamic environmental changes.

B. *Objectives*

The study focuses on the following objectives:

1) To develop a smooth and accurate reference trajectory for bi-copter UAVs.
2) To manage the complexities of bi-copter dynamics, including non-linearities and rotor coupling effects.
3) To compare the performance of LMPC and NMPC in terms of stability, robustness, and computational efficiency.
4) To ensure the control system's adaptability to environmental changes, such as wind disturbances and payload variations.

III. Methodology

The approach involves:

1) **Reference Trajectory Design:** Creating a mission-specific trajectory to ensure smooth path adherence.

2) **Linear MPC Framework:** Implementing LMPC to handle trajectory tracking under linear system dynamics assumptions.
3) **Nonlinear MPC Framework:** Developing NMPC to capture the true dynamics of the bi-copter more accurately.
4) **Simulations:** Conducting extensive simulations to evaluate the performance of both control strategies under various scenarios and disturbances.

IV. Dynamic Model

First, we establish two coordinate systems as shown in Fig. 1: the Body coordinate system $OX_bY_bZ_b$ and the Ground coordinate system $OX_gY_gZ_g$.

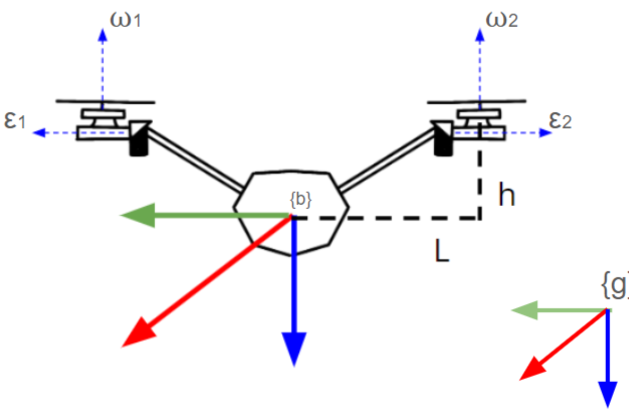

Fig. 1. Coordinate systems: Body and Ground.

As shown in Fig. 2, we define the Euler angles:

- **Yaw angle** (ψ): The angle between the body axis in the horizontal plane and the ground axis, positive to the right.
- **Pitch angle** (θ): The angle between the body axis and the horizontal plane, positive when rising.
- **Roll angle** (ϕ): The angle of the symmetry plane rotating around the body axis, positive for right roll.

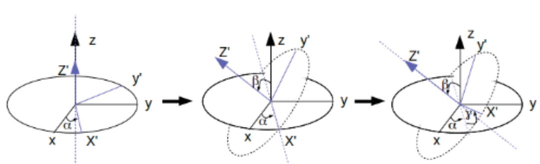

Fig. 2. Euler angles: Yaw, Pitch, and Roll.

When using Euler angles to describe the attitude of an aircraft, we can obtain the transformation matrix from the ground coordinate system to the body coordinate system as:

$$\mathbf{T}_{bg} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{1}$$

The relationship matrix between the angular velocity of the body and the first derivative of the Euler angles from the ground system is:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

According to Newton's second law, using the matrix $\mathbf{T}_{bg}$ and inertia matrix $\mathbf{I}$, we can derive the linear and angular motion equations of the aircraft in the ground coordinate system under no wind conditions as follows:

$$\begin{aligned} F_x &= k_1(\sin\xi_1\cos\psi\sin\theta\cos\psi \\ &\quad - \sin\xi_2\cos\theta\cos\omega_2) \\ F_y &= k_2(\cos\theta\cos\omega_1\omega_1 + \cos\theta\sin\omega_2) \\ F_z &= k_3(\sin\xi_1\cos\psi\cos\sin\theta\sin\psi \\ &\quad - \cos\theta\sin\omega_1) \end{aligned} \tag{3}$$

The moments in roll (L), pitch (M), and yaw (N) are:

$$\begin{aligned} L &= I_{xx}(\sin\theta\cos\xi_1\cos\theta\cos\psi + lk_1) \\ M &= I_{yy}(\sin\theta\sin\xi_1\cos\psi + lk_2) \\ N &= I_{zz}(\sin\xi_1\cos\theta\cos\xi_1\cos\psi + lk_3) \end{aligned} \tag{4}$$

The nonlinear coupling model of the attitude angles can be decomposed into relatively independent control channels by introducing control variables U_1, U_2, U_3, U_4:

$$\begin{aligned} U_1 &= K_T(\omega_1\cos\xi_1 + \omega_2\cos\xi_2) \\ U_2 &= K_T(\omega_1\cos\xi_1 + \omega_2\cos\xi_2) \\ U_3 &= K_T(\omega_1\sin\xi_1 + \omega_2\sin\xi_2) \\ U_4 &= K_T(\omega_1\sin\xi_1 + \omega_2\sin\xi_2) \end{aligned} \tag{5}$$

The angular motion is not affected by the linear motion, but the linear motion is affected by the angular motion. Ignoring the drag coefficient, the mathematical model becomes:

$$
\begin{aligned}
\ddot{x} &= \frac{1}{m}\Big(U_1 \cos\theta \cos\psi \\
&\quad + U_2(\sin\xi_1 \cos\psi + \sin\theta \cos\psi) \\
&\quad + U_3 \cos\theta \sin\psi \\
&\quad + U_4(\sin\xi_1 \cos\psi + \sin\theta \sin\psi)\Big) \\
\ddot{y} &= \frac{1}{m}\Big(U_1 \cos\theta \cos\psi \\
&\quad + U_2(\sin\xi_1 \cos\psi + \sin\theta \cos\psi) \\
&\quad + U_3 \cos\theta \sin\psi \\
&\quad + U_4(\sin\xi_1 \cos\psi + \sin\theta \sin\psi)\Big) \\
\ddot{z} &= \frac{1}{m}\Big(U_1 \cos\theta \cos\psi \\
&\quad + U_2(\sin\xi_1 \cos\psi + \sin\theta \cos\psi) \\
&\quad + U_3 \cos\theta \sin\psi \\
&\quad + U_4(\sin\xi_1 \cos\psi + \sin\theta \sin\psi)\Big)
\end{aligned} \tag{6}
$$

TABLE I
PHYSICAL PARAMETERS OF THE BI-COPTER

Parameter	**Symbol**	**Value**
Distance between motor and CoM	L	0.225 m
Vertical distance between CoG and center of the rotor	h	0.042 m
Moment of inertia around x-axis	I_{xx}	0.116 kg·m^2
Moment of inertia around y-axis	I_{yy}	0.0408 kg·m^2
Moment of inertia around z-axis	I_{zz}	0.105 kg·m^2
Mass	m	1.192 kg
Gravitational acceleration	g	9.81 m/s^2
Lift constant (thrust factor)	C_T	2.98×10^{-6} Ns^2/rad^2

The above physical parameters are used as the parameters of the simulation model.

V. MPC CONTROL DESIGN

Model Predictive Control (MPC) is a powerful control strategy that optimizes the control inputs over a prediction horizon to track a desired reference trajectory while respecting system constraints. In this paper, we apply MPC to control a bi-copter, ensuring stability and performance.

A. Linear MPC Design of Bi-Copter

1) Discretization of the Continuous-Time State-Space Matrices: The continuous-time state-space representation of the bi-copter is given by:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_{cont}\mathbf{x}(t) + \mathbf{B}_{cont}\mathbf{u}(t) \tag{7}$$

$$\mathbf{y}(t) = \mathbf{C}_{cont}\mathbf{x}(t) + \mathbf{D}_{cont}\mathbf{u}(t) \tag{8}$$

To implement the MPC in a digital controller, we discretize the system using a sampling period T_s:

$$\mathbf{A}_{disc} = e^{\mathbf{A}_{cont}T_s} \tag{9}$$

$$\mathbf{B}_{disc} = \left(\int_0^{T_s} e^{\mathbf{A}_{cont}\tau} d\tau\right)\mathbf{B}_{cont} \tag{10}$$

The discrete-time state-space representation is:

$$\mathbf{x}[k+1] = \mathbf{A}_{disc}\mathbf{x}[k] + \mathbf{B}_{disc}\mathbf{u}[k] \tag{11}$$

$$\mathbf{y}[k] = \mathbf{C}_{disc}\mathbf{x}[k] + \mathbf{D}_{disc}\mathbf{u}[k] \tag{12}$$

2) Control Inputs and Constraints: Control inputs $\mathbf{u}$ are the motor commands applied to the bi-copter. The constraints on these inputs are defined as:

$$\mathbf{u}_{min} \leq \mathbf{u}[k] \leq \mathbf{u}_{max} \tag{13}$$

For instance, if the control input limits are $-10 \leq u_i \leq 10$ for each motor, we have:

$$-10 \leq u_1[k] \leq 10 \tag{14}$$

$$-10 \leq u_2[k] \leq 10 \tag{15}$$

3) Constraints: The MPC ensures the system follows the defined dynamics and control inputs stay within allowable limits.

State transition constraints:

$$\mathbf{x}[k+1] = \mathbf{A}_{disc}\mathbf{x}[k] + \mathbf{B}_{disc}\mathbf{u}[k] \tag{16}$$

Input constraints:

$$\mathbf{u}_{min} \leq \mathbf{u}[k] \leq \mathbf{u}_{max} \tag{17}$$

4) Objective Function: The objective function in MPC minimizes the deviation from a reference trajectory and the control effort. It is defined as:

$$
\begin{aligned}
J = \sum_{k=0}^{Hp-1} & \big[(\mathbf{x}[k] - \mathbf{x}_{ref}[k])^T \mathbf{Q} (\mathbf{x}[k] - \mathbf{x}_{ref}[k]) \\
& + \mathbf{u}[k]^T \mathbf{R} \mathbf{u}[k]\big] \\
& + (\mathbf{x}[Hp] - \mathbf{x}_{ref}[Hp])^T \mathbf{Q}_f (\mathbf{x}[Hp] - \mathbf{x}_{ref}[Hp])
\end{aligned} \tag{18}
$$

Where: - $\mathbf{Q}$ is the state weighting matrix. - $\mathbf{R}$ is the control weighting matrix. - $\mathbf{Q}_f$ is the terminal state weighting matrix. - $\mathbf{x}_{ref}$ is the reference trajectory.

5) Tuning Parameters: Tuning parameters in MPC include the prediction horizon Hp, the weighting matrices for state error $\mathbf{Q}$ and control effort $\mathbf{R}$, and the terminal state weighting matrix $\mathbf{Q}_f$.

To determine an appropriate time step (dt) and prediction horizon (Hp), an open-loop simulation was conducted using both the linearized plant and the nonlinear system. These simulations were compared until the results converged, as illustrated in Figure 3. A time step (dt) of 0.1 was used, and through an iterative process, the minimum Hp value required was found to be 6. However, for the purposes of our development, we opted to use an Hp value of 20.

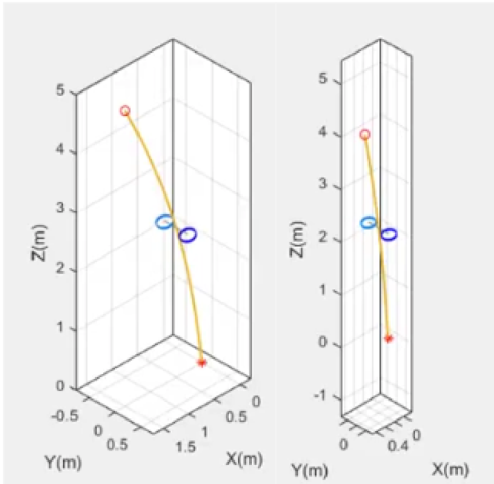

Fig. 3. linear simulation and nonlinear simulation

Prediction horizon:

$$Hp = 20 \tag{19}$$

Weighting matrices:

$$\mathbf{Q} = \text{diag}(10, 10, 10, 1, 1, 1, 1, 1, 1, 1, 1, 1) \tag{20}$$

$$\mathbf{R} = \text{diag}(1, 1) \tag{21}$$

$$\mathbf{Q}_f = \mathbf{Q} \tag{22}$$

6) Normalization: Normalization scales the states and control inputs to ensure they are within a similar range, improving numerical stability and performance.

Normalization matrices for states $\mathbf{x}_{norm}$ and controls $\mathbf{u}_{norm}$:

$$\mathbf{x}_{norm} = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \tag{23}$$

$$\mathbf{u}_{norm} = \text{diag}(1, 1) \tag{24}$$

Normalized cost function:

$$\begin{aligned} J = \sum_{k=0}^{Hp-1} & \left[(\mathbf{x}_{\text{norm}}(\mathbf{x}[k] - \mathbf{x}_{\text{ref}}[k]))^T \mathbf{Q} (\mathbf{x}_{\text{norm}}(\mathbf{x}[k] - \mathbf{x}_{\text{ref}}[k])) \right. \\ & \left. + (\mathbf{u}_{\text{norm}}\mathbf{u}[k])^T \mathbf{R} (\mathbf{u}_{\text{norm}}\mathbf{u}[k]) \right] \\ & + (\mathbf{x}_{\text{norm}}(\mathbf{x}[Hp] - \mathbf{x}_{\text{ref}}[Hp]))^T \mathbf{Q}_f (\mathbf{x}_{\text{norm}}(\mathbf{x}[Hp] - \mathbf{x}_{\text{ref}}[Hp])) \end{aligned} \tag{25}$$

The MPC framework for the bi-copter involves discretizing continuous-time state-space models, defining control inputs and constraints, formulating an objective function to minimize tracking error and control effort, tuning the parameters for optimal performance, and normalizing states and inputs for improved numerical stability. The goal is to optimize the control inputs over a prediction horizon to follow a given reference trajectory while adhering to constraints and minimizing the control effort. The Linear MPC done by using YALMIP toolbox in matlab (quadprog).

B. NonLinear MPC

Model Predictive Control (MPC) is a powerful control strategy that optimizes control inputs over a prediction horizon to track a desired reference trajectory while respecting system constraints. This project applies NMPC to control a bi-copter, ensuring stability and performance in the presence of wind disturbances.

1) System Dynamics: The state vector $\mathbf{x}$ for the bi-copter includes positions, orientations (Euler angles), and their respective rates:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{26}$$

The control input vector $\mathbf{u}$ consists of:

$$\mathbf{u} = \begin{bmatrix} T_1 & \tau_\phi & \tau_\theta & \tau_\psi \end{bmatrix}^T$$

where T_1 is the thrust and $\tau_\phi, \tau_\theta, \tau_\psi$ are the torques about the roll, pitch, and yaw axes, respectively.

2) Rotation Matrix: The rotation matrix $\mathbf{R}$ transforms the thrust from the body frame to the inertial frame, considering the Euler angles (ϕ, θ, ψ):

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{27}$$

3) Forces and Acceleration: The forces acting on the bi-copter are the gravitational force $\mathbf{F}_{gravity}$ and the thrust $\mathbf{F}_{thrust}$:

$$\begin{aligned} \mathbf{F}_{\text{gravity}} &= \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \\ \mathbf{F}_{\text{thrust}} &= \begin{bmatrix} 0 \\ 0 \\ -T_1 \end{bmatrix} \end{aligned} \tag{28}$$

The total acceleration $\mathbf{a}$ in the inertial frame is given by:

$$\mathbf{a} = \frac{1}{m}(\mathbf{R}\mathbf{F}_{\text{thrust}} + \mathbf{F}_{\text{gravity}}) + \mathbf{w} \tag{29}$$

4) Angular Dynamics: The angular velocities and torques are:

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{30}$$

The inertia matrix $\mathbf{I}$ is diagonal:

$$\mathbf{I} = \text{diag}(I_{xx}, I_{yy}, I_{zz}) \tag{31}$$

The angular acceleration $\dot{\boldsymbol{\omega}}$ is:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})) \tag{32}$$

5) State Dynamics: Combining the translational and rotational dynamics, the state derivative $\dot{\mathbf{x}}$ is:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \mathbf{a} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \frac{1}{m}(\mathbf{R}\mathbf{F}_{\text{thrust}} + \mathbf{F}_{\text{gravity}}) + \mathbf{w} \\ \mathbf{I}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})) \end{bmatrix} \tag{33}$$

Using CasADi, the nonlinear dynamics function dynamics$(\mathbf{x}, \mathbf{u}, \mathbf{w})$ is defined to compute $\dot{\mathbf{x}}$.

6) Reference Trajectory: The reference trajectory $\mathbf{x}_{ref}$ is designed to follow a linear uprising path:

$$\mathbf{x}_{\text{ref}}(t) = \begin{bmatrix} 0.01t \\ 0.005t \\ 0.01t \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

7) NMPC Problem Formulation: The NMPC problem involves predicting the future states and control inputs over a prediction horizon Hp. The variables are:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_{Hp} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{Hp-1} \end{bmatrix} \tag{35}$$

8) Cost Function: The cost function J is designed to minimize the tracking error and control effort:

$$\begin{aligned} J = \sum_{k=0}^{Hp-1} \Big[& (\mathbf{x}_{\text{norm}}(\mathbf{x}[k] - \mathbf{x}_{\text{ref}}[k]))^T \mathbf{Q} (\mathbf{x}_{\text{norm}}(\mathbf{x}[k] - \mathbf{x}_{\text{ref}}[k])) \\ & + (\mathbf{u}_{\text{norm}}\mathbf{u}[k])^T \mathbf{R} (\mathbf{u}_{\text{norm}}\mathbf{u}[k]) \Big] \\ & + (\mathbf{x}_{\text{norm}}(\mathbf{x}[Hp] - \mathbf{x}_{\text{ref}}[Hp]))^T \mathbf{Q}_f (\mathbf{x}_{\text{norm}}(\mathbf{x}[Hp] - \mathbf{x}_{\text{ref}}[Hp])) \end{aligned} \tag{36}$$

9) Normalization and Weight Matrices: Normalization scales the states and control inputs to ensure they are within a similar range, improving numerical stability and performance.

The normalization factors for states $\mathbf{x}_{norm}$ and controls $\mathbf{u}_{norm}$ are chosen based on the maximum expected errors:

$$\mathbf{x}_{norm} = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

$$\mathbf{u}_{norm} = \text{diag}(1, 1, 1, 1)$$

The weight matrices for the cost function are defined and then normalized:

$$Q_{pos} = \text{diag}(100, 100, 100)$$

$$Q_{ori} = \text{diag}(10, 10, 10)$$

$$R = \mathbf{I}_4$$

The maximum position error of 1 meter, maximum orientation error of π radians, and maximum control input value of 10:

$$Q_{pos\_norm} = Q_{pos}/(1^2)$$

$$Q_{ori\_norm} = Q_{ori}/(\pi^2)$$

$$R_{norm} = R/(10^2)$$

The normalized cost function ensures that the errors and control efforts are weighted appropriately.

C. Constraints

1) State Transition Constraints: Ensure that the predicted states adhere to the nonlinear dynamics:

$$\mathbf{X}[k+1] = \mathbf{X}[k] + dt \cdot \text{dynamics}(\mathbf{X}[k], \mathbf{U}[k], \mathbf{w})$$

2) Input Constraints: Ensure that the control inputs remain within specified bounds:

$$\mathbf{u}_{min} \leq \mathbf{U}[k] \leq \mathbf{u}_{max}$$

3) Wind Disturbance $\mathbf{w}$*:* Wind disturbance is modeled as a Gaussian noise applied at each time step:

$$\mathbf{w}_k = \begin{bmatrix} w_{x,k} \\ w_{y,k} \\ w_{z,k} \end{bmatrix} \sim \mathcal{N}(0, \sigma^2)$$

Here, $\mathcal{N}(0, \sigma^2)$ denotes a normal distribution with mean 0 and variance σ^2. For example, in the provided code:

$$\mathbf{w}_k = 0.1 \cdot \text{randn}(3, 1)$$

This adds random Gaussian noise with a standard deviation of 0.1 to each component of the wind disturbance.

D. *Simulation with Disturbance*

1. **Initial Setup**: Initialize the state and control trajectories.
2. **Wind Disturbance Generation**: Generate wind disturbance for each time step:

$$\mathbf{w}_k = 0.1 \cdot \text{randn}(3, 1)$$

3. **NMPC Optimization**: At each time step, solve the NMPC problem considering the disturbance:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + dt \cdot \text{dynamics}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$$

4. **State Update**: Update the state using RK4 integration:

$$\begin{aligned}
k1 &= \text{dynamics}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \\
k2 &= \text{dynamics}(\mathbf{x}_k + 0.5 \cdot dt \cdot k1, \mathbf{u}_k, \mathbf{w}_k) \\
k3 &= \text{dynamics}(\mathbf{x}_k + 0.5 \cdot dt \cdot k2, \mathbf{u}_k, \mathbf{w}_k) \\
k4 &= \text{dynamics}(\mathbf{x}_k + dt \cdot k3, \mathbf{u}_k, \mathbf{w}_k)
\end{aligned}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{dt}{6}(k1 + 2k2 + 2k3 + k4)$$

5. **Apply Optimal Control**: Use the first control input from the optimized sequence:

$$\mathbf{u}_k = \mathbf{u}_{opt}[:, 1]$$

6. **Reference Trajectory Adjustment**: Ensure the current reference trajectory matches the prediction horizon:

$$\mathbf{x}_{ref}[k : min(k + Hp, N)]$$

The NMPC framework for the bi-copter incorporates wind disturbances into the dynamics to simulate realistic flight conditions. The controller optimizes the control inputs over a prediction horizon to track a reference trajectory while minimizing the impact of disturbances and control efforts. By using a detailed model of the bi-copter's dynamics and accounting for external forces, the NMPC ensures robust and stable flight performance.

VI. Results

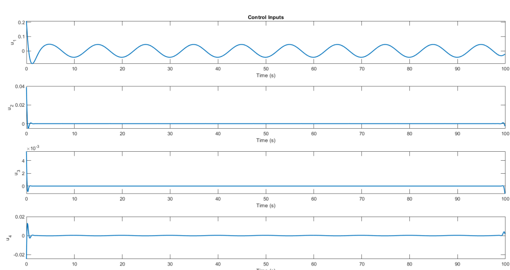

Fig. 4. Control input result.

The thrust control input u_1 shows an initial spike to lift the bi-copter, followed by oscillations to maintain altitude and trajectory. The torque controls u_2, u_3, u_4 remain near zero, indicating minimal roll, pitch, and yaw adjustments, reflecting stable orientation control.

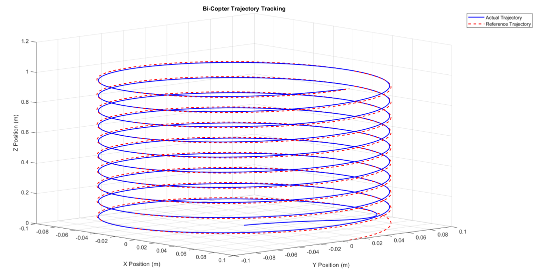

Fig. 5. Trajectory Following result from LMPC.

The actual trajectory (blue line) closely follows the reference trajectory (red dashed line), demonstrating effective tracking by the MPC. The smooth actual trajectory indicates stable flight without abrupt movements.

A. *Non Linear MPC Result*

The thrust control input u_1 remains relatively high and steady with some variations, indicating significant thrust is needed to counteract disturbances and maintain altitude. The roll u_2 and pitch u_3 torque inputs show minimal activity, reflecting slight adjustments for stability. The yaw torque u_4 input shows very small values, suggesting minimal yaw adjustments.

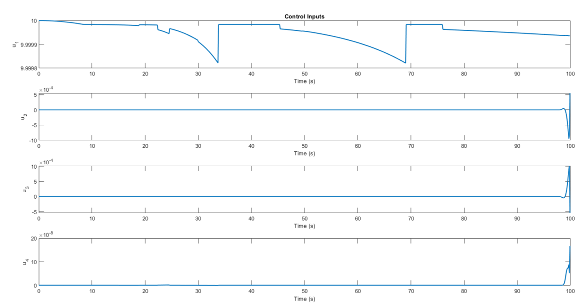

Fig. 6. Control Input NMPC.

The actual trajectory (blue line) closely follows the reference trajectory (red dashed line) initially but starts to deviate over time due to disturbances. The smooth actual trajectory indicates stable flight without abrupt movements.

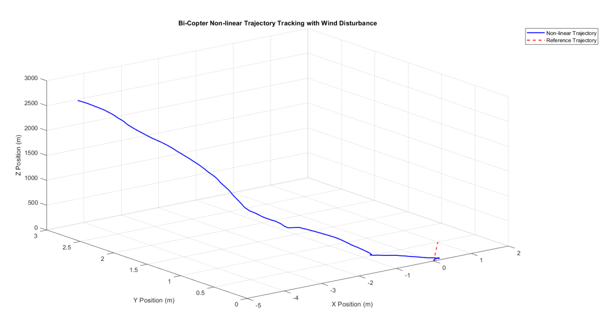

Fig. 7. Trajectory Following result from NMPC.

The position error increases over time, suggesting the bi-copter gradually deviates from the desired path, likely due to cumulative disturbances.

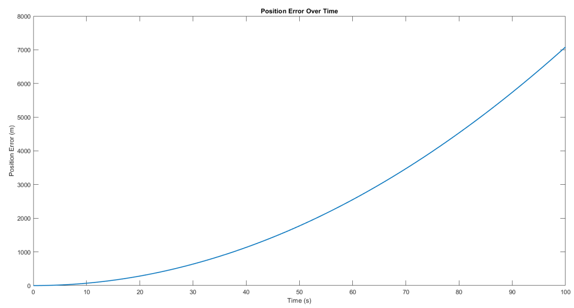

Fig. 8. Position Error of NMPC.

The nonlinear MPC ensures effective initial control and stable flight for the bi-copter, maintaining the trajectory closely to the reference path despite disturbances. However, the increasing position error over time highlights the need for further tuning and possibly adaptive control strategies to improve long-term accuracy.

B. Linear MPC Validation With Non-linear Model of Bi-Copter

The thrust control input u_1 exhibits high-frequency oscillations initially, then stabilizes, indicating the controller's effort to handle nonlinear dynamics and maintain altitude. The roll torque input u_2 shows a significant peak early on, followed by stabilization, reflecting the initial adjustment required to handle non linearities. The pitch torque input u_3 remains near zero, indicating minimal pitch adjustments are needed. The yaw torque input u_4 shows initial oscillations followed by stabilization, indicating adjustments for yaw control.

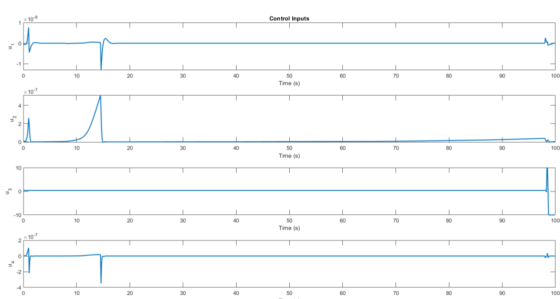

Fig. 9. Control input result from Validation.

The actual trajectory (blue line) closely follows the reference trajectory (red dashed line), demonstrating the effectiveness of the linear MPC when applied to the nonlinear model. The controller maintains the desired path despite the inherent non linearities of the bi-copter.

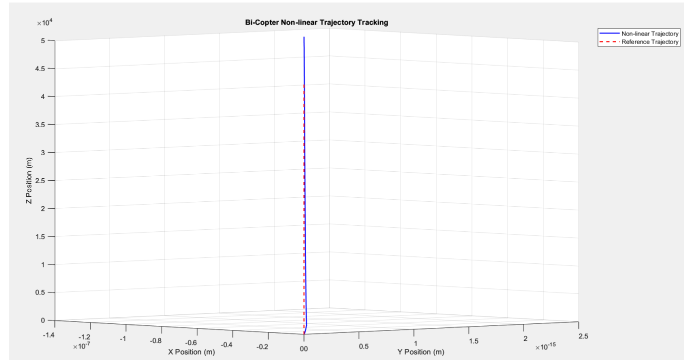

Fig. 10. Trajectory Validation.

The validation of the linear MPC using a nonlinear bi-copter model demonstrates the controller's robustness and effectiveness. The control inputs show initial adjustments to handle nonlinear dynamics, followed by stabilization. The trajectory tracking closely matches the reference path, indicating that the linear MPC can successfully control the bi-copter even in the presence of non linearities. This validation highlights the potential of the linear MPC for practical applications, ensuring precise control and stable flight performance.

C. KPI Index LMPC vs NMPC

In this study, we compare the performance of Linear Model Predictive Control (LMPC) and Nonlinear Model Predictive Control (NMPC) for controlling a bi-copter. The Key Performance Indicators (KPIs) analyzed include control input error (RMSE) and computation time.

TABLE II
COMPARISON OF KPIS FOR LMPC AND NMPC

KPI	LMPC	NMPC
Control Input Error (RMSE)	0.532833	158.113135
Avg. Computation Time (s)	0.011802	0.047536
Total Computation Time (s)	11.802330	47.536358

The RMSE for LMPC is significantly lower at 0.532833, indicating that the linear model predictive control closely follows the desired control inputs with minimal deviation. This suggests that LMPC is effective in maintaining the desired control actions for the bi-copter. In contrast, the RMSE for NMPC is substantially higher at 158.113135. This large error indicates that the nonlinear model predictive control struggles to follow the desired control inputs. The high RMSE could be due to the complexity and potential nonlinearity of the bi-copter's dynamics which NMPC aims to handle but may require further tuning or adaptation.

The average computation time per control cycle is lower for LMPC at 0.011802 seconds compared to NMPC's 0.047536 seconds. This indicates that LMPC is computationally more efficient, making it more suitable for real-time applications where quick control decisions are necessary. Over the entire simulation, the total computation time for LMPC is 11.802330 seconds, while for NMPC it is 47.536358 seconds. This further highlights the computational efficiency of LMPC compared to NMPC, which requires more time to solve the optimization problem at each step.

The comparison between LMPC and NMPC for bi-copter control demonstrates that LMPC offers better control performance in terms of lower control input error and higher computational efficiency. NMPC, while potentially more powerful in handling nonlinear dynamics, requires further optimization to reduce the control input error and computation time for practical applications.

VII. Concluding remarks

In this study, we developed and validated both linear and nonlinear Model Predictive Control (MPC) frameworks for controlling a bi-copter. The linear MPC demonstrated effective initial control, maintaining the trajectory closely to the reference path despite disturbances. However, the increasing position error over time highlighted the challenge of sustained disturbance rejection, necessitating further tuning or adaptive strategies. The nonlinear MPC showed robust performance initially but struggled with cumulative disturbances, indicating that while it handles instantaneous errors well, long-term accuracy requires improvement. Validation of the linear MPC using a nonlinear bi-copter model demonstrated that the controller is capable of managing nonlinear dynamics effectively, with the actual trajectory closely following the reference path and control inputs stabilizing over time.

Future work will focus on enhancing the robustness of both control strategies against persistent disturbances. This may involve integrating adaptive control techniques or robust MPC frameworks that can dynamically adjust to changing conditions. Additionally, implementing real-time optimization methods and exploring machine learning approaches for predictive modeling and control could further improve performance. Extensive testing under various environmental conditions will be essential to ensure the reliability and effectiveness of the control systems in practical applications. The ultimate goal is to develop a comprehensive control framework that ensures precise trajectory tracking, stability, and robustness for bi-copters in real-world scenarios.

References

[1] M. Nataraj, K. Madhukumar, and M. Karthik, "Design and fabrication of two rotors bicopter," *IJRTER*, vol. 3, no. 1, pp. 272–281, January 2017.

[2] Y. Qin, W. Xu, A. Lee, and F. Zhang, "Gemini: A compact yet efficient bi-copter uav for indoor applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3213–3220, 2020.

[3] O. A. et al., "Design of a robotic bicopter," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, 2019, preprint.

[4] M. Islam, M. Okasha, and M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," *IOP Conference Series: Materials Science and Engineering*, vol. 270, p. 012007, 2017.