The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

Model Predictive Control for Stabilizing Quadcopter Flight and Following Trajectories

Ahmed J. Abougarair
Electrical and Electronics Engineering, University of Tripoli, Tripoli, Libya
Email: a.abougarair@uot.edu.ly

Mohamed Almograbi
Electrical and Electronics Engineering, University of Tripoli, Tripoli, Libya

Abdusalam A. Alaktiw
Biomedical Engineering, University of Tripoli, Tripoli, Libya

Abstract— **Controlling a quadcopter is inherently challenging due to its nonlinear dynamics and susceptibility to external disturbances such as wind gusts and sensor noise. This paper explores the use of Model Predictive Control (MPC) to tackle these challenges. The goal is to create a robust control system capable of stabilizing and directing a quadcopter along a specified trajectory, even when faced with disturbances. The next step derives the Linear Quadratic Regulator (LQR), which is used to stabilize the quadcopter. The research involves creating a detailed mathematical model of the quadcopter's dynamics, followed by the design and implementation of LQR and MPC. Through extensive simulations, the effectiveness of the MPC approach is validated, demonstrating its ability to maintain stability and achieve precise control under various conditions. This study underscores the potential of MPC as a powerful control strategy for UAVs, offering significant advantages for real-world applications where traditional control methods may fall short. The LQR controller balances the performance of the drone and the energy it consumes by specifying the weighting matrix of performance cost Q and control cost R to calculate an optimized controller. MPC provides a robust and effective control strategy for quadcopters, offering improved performance and reliability. Its ability to handle nonlinear dynamics, manage constraints, and optimize control actions makes it an essential tool for advanced aerial vehicle control.**

Index Terms— **MPC, LQR, Tracking control, UAV, Cost function.**

I. INTRODUCTION

Quadcopters, a type of unmanned aerial vehicle (UAV), have gained significant popularity due to their versatility, agility, and wide range of applications, from recreational use to complex industrial tasks [1]. Their ability to hover, take off, and land vertically makes them particularly useful in environments where fixed-wing aircraft may not be practical. The quadcopter's flight dynamics are characterized by non-linearities and time-varying parameters, which pose significant challenges for control systems [2]. The quadrotor features a straightforward geometric design comprising four rotorsmounted on a rigid frame, with each rotor controlled independently. This mechanical simplicity makes quadrotors appealing for various applications. Unlike other aircraft types, quadrotors lack flapping hinges, and their rotor blades are short and robust. In the quadrotor configuration shown in Figure 1, the rotors are positioned so that rotors 1 and 3, as well as rotors 2 and 4, rotate in opposite pitch directions. When viewed from above, the rotation directions for omega 1 and omega 3 are positive in the clockwise direction. Conversely, when viewed from below, omega 2 and omega 4 also rotate positively in the clockwise direction.

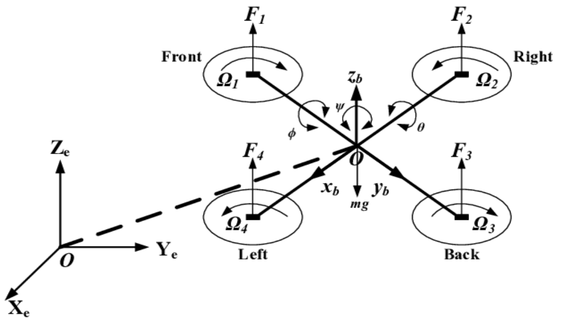

Figure 1. Quadcopter Configuration with rotors Directions [3]

Traditional control methods often fall short in adapting to these variations, leading to suboptimal performance. In contrast, Quadcopters operate with six DOF, encompassing three translational axes (x, y, z) and three rotational axes (roll, pitch, yaw) [3]. Managing this complex control challenge requires sophisticated strategies to address the inherent nonlinear dynamics and coupling between different axes of motion. These dynamics are typically modeled using nonlinear equations that account for thrust generation, aerodynamic forces, and the vehicle's rotational behavior. While linearizing these models around specific operating points, such as hovering, can simplify control design, real-world variability and uncertainties necessitate robust control techniques to maintain performance and stability [4].

Received 06 Oct, 2024 ; Revised 01 Apr, 2025; Accepted 04 Ape ,2025.
Available online 10 Apr , 2025.
DOI: 10.36602/ijeit.v13i2.562

Various types of controllers have been discussed in existing literature. PID controllers are often preferred for their simplicity and satisfactory performance [5,6].

However, they may struggle with complex trajectories that feature considerable curvature. This paper adopts a methodology akin to that of, combining the differential flatness property of quadcopters with feedforward linearization, a technique referred to as flatness model predictive control (FMPC) [7,8].

MPC has emerged as a pivotal strategy in the stabilization and trajectory tracking of quadcopters, addressing the complexities inherent in their dynamic flight behavior [9]. MPC offers a robust framework that predicts future states of the system based on a mathematical model, allowing for proactive adjustments to control inputs. This capability not only enhances stability but also facilitates complex maneuvering essential for tasks such as aerial inspections or search-and-rescue operations. This paper delves into the application of MPC techniques tailored for quadcopter systems, emphasizing their ability to manage constraints and optimize performance in real-time scenarios. Furthermore, this paper outlines the formulation of an MPC controller that incorporates feedback mechanisms to improve trajectory tracking accuracy. Through simulations and experimental validations, we demonstrate the effectiveness of MPC in achieving desired flight paths while adapting to real-time disturbances. The objectives of the paper are developing a develop an MPC algorithm to control the quadcopter's position and orientation and optimize the performance of the MPC controller to minimize error and energy consumption while ensuring stability and robustness [10]. Table I summarize the top papers for the MPC based quadcopter system.

TABLE I. SUMMARIZE THE NEW PAPERS FOR MPC

Paper	Insights
[11]	The paper does not specifically address MPC for stabilizing quadcopter flight. Instead, it focuses on dynamic feedback design using piecewise linear feedback with saturation, enhancing robustness in tracking systems under velocity and control constraints.
[12]	The paper does not discuss MPC; instead, it focuses on an event-driven mechanism for altitude and attitude tracking in quadrotors, utilizing a Terminal Sliding Mode Control approach to handle uncertainties and actuator saturation effectively.
[13]	MPC enhances quadrotor stability and trajectory tracking. It effectively manages dynamic environments, adapting to disturbances while ensuring precise flight paths, making it suitable for complex applications like urban navigation.
[14]	The paper does not discuss MPC for stabilizing quadcopter flight or following trajectories. Instead, it focuses on two control strategies ensuring robust trajectory tracking despite uncertainties and disturbances, achieving global finite-time convergence of tracking errors.
[15]	The paper presents a model predictive control (MPC) strategy for quadcopters, ensuring uniform almost global asymptotic stability while effectively tracking fast trajectories. It combines an outer loop for acceleration reference generation with a nonlinear inner loop for attitude control.
[16]	The paper presents an improved Model Predictive Path Integral (MPPI) controller for quadcopter trajectory tracking. It integrates a Multilayer Perceptron neural network to adaptively adjust control inputs, significantly reducing trajectory tracking errors.
[17]	The paper does not discuss Model Predictive Control (MPC) for stabilizing quadcopter flight or trajectory following. Instead, it focuses on an observer-based adaptive neural control framework using a high-gain disturbance observer and neural-network-based adaptive fractional sliding mode control.
[18]	MPC is utilized in the outer loop of the proposed control architecture to stabilize quadrotor flight and ensure accurate trajectory tracking, effectively managing variations in mass, center of gravity, and external disturbances during operation.
[19]	The paper focuses on Sliding Mode Control (SMC) for quadcopter trajectory tracking and stability under disturbances, rather than MPC. It demonstrates SMC's effectiveness through simulations, contrasting it with PID control in disturbance scenarios.
[20]	The paper does not discuss MPC for stabilizing quadcopter flight or following trajectories. Instead, it focuses on a hybrid controller (PFOIDSMCBS) designed for trajectory tracking under disturbances, demonstrating superior performance compared to existing methods.
[21]	The paper does not discuss MPC for stabilizing quadcopter flight or trajectory following. It focuses on linear PID, nonlinear geometric tracking, and robust Sliding Mode Control techniques for enhancing trajectory tracking in autonomous multirotor robots.
[22]	The paper does not discuss MPC; it focuses on Super Twisting Sliding Mode Control with a novel Fuzzy PID Surface for trajectory tracking of quadrotors, enhancing robustness against disturbances and improving tracking performance compared to other control methods.
[23]	MPC effectively stabilizes quadcopter flight and follows trajectories by addressing sensor issues, input and state constraints, and adapting to disturbances, as demonstrated through LMPC and NMPC frameworks.
[24]	The paper does not discuss MPC; instead, it focuses on adaptive sliding mode control for trajectory tracking of quadrotors, addressing input saturation and disturbances through an innovative control strategy that enhances robustness and adaptability.
[25]	The paper integrates MPC with an Extended State Observer (ESO) to stabilize quadcopter flight and ensure precise trajectory tracking by addressing external disturbances and internal uncertainties, enhancing flight coordination in complex environments.
[26]	The paper does not discuss MPC for stabilizing quadcopter flight or following trajectories. Instead, it focuses on a hybrid control strategy combining bioinspired backstepping and sliding mode control to enhance trajectory tracking and stability.
[27]	The paper employs an adaptive MPC algorithm to stabilize quadcopter flight and predict trajectories. Experimental validation with the Parrot Bebop 2 demonstrates effective trajectory tracking and stable autonomous navigation, meeting real-time operational requirements.
[28]	The paper does not specifically address MPC for stabilizing quadcopter flight or following trajectories. Instead, it focuses on a Type-2 Fuzzy Logic Controller with Genetic Algorithm tuning for robust trajectory tracking in windy environments.
[29]	The paper focuses on trajectory tracking using reinforcement learning and PD control, not MPC. It emphasizes achieving optimal trajectory following under noise conditions, utilizing simulations for position and attitude control of the quadcopter.

[30]	The paper does not specifically address MPC for stabilizing quadcopter flight or following trajectories. Instead, it focuses on adaptive optimization control using backstepping, neural networks for uncertainty estimation, and reinforcement learning for optimal problem-solving in UAV tracking control.

II. CONFIGURATION AND MATHEMATICAL MODEL OF QUADCOPTER

With two booms crossing in the middle and propellers symmetrically positioned at the extremities of four arms, quadcopters use four propellers to provide thrust. The autopilot and other equipment are located in the center of the fuselage [31].

Cross Configuration: As seen in Figure 1, this configuration can be separated into two types: X and plus (+). Because additional rotors help control pitch and roll and because their forward field of view is less obstructed than that of plus-configuration quadcopters, X-configuration quadcopters are more maneuverable and therefore more popular.

Ring Configuration: The ring configuration offers a more rigid structure than the traditional cross fuselage, which helps reduce vibrations from the motors and propellers. However, this design results in a heavier fuselage, potentially reducing maneuverability.

In quadcopter dynamics and control, reference frames are essential for describing the motion of the vehicle. Two primary frames of reference are commonly used: the Terrestrial Coordinate System and the body or inertial frame [12].

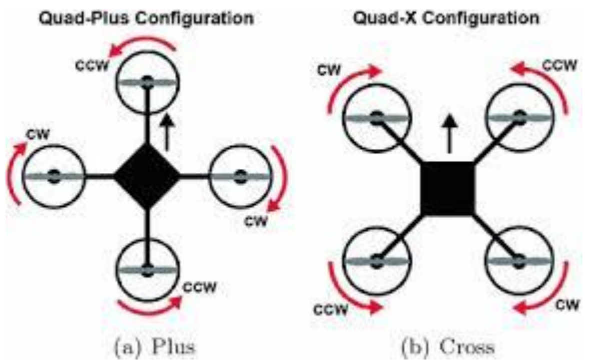

Figure 1. Plus and cross quadcopter

Terrestrial Coordinate System: The Terrestrial Coordinate System, or Earth frame, is a Cartesian reference frame used for describing positions and motions relative to the Earth. This frame is particularly useful for determining the global position of the quadcopter. The Earth frame consists of orthogonal x, y, and z axes [3,32].

x-axis: Typically represents the forward direction in the horizontal plane.

y-axis: Represents the lateral direction in the horizontal plane.

z-axis: Points upwards, perpendicular to the Earth's surface.

Body or Inertial Frame: The Body or Inertial Frame is a reference frame attached to the quadcopter itself. This frame moves with the quadcopter and provides a local perspective on its orientation and dynamics.

The relationship between the Earth frame and the Body frame is central to quadcopter control. The quadcopter's position and orientation are typically expressed in the Earth frame, while the forces and torques applied by the rotors are more conveniently described in the Body frame.

To convert between these frames, rotation matrices or quaternions are used. For example, the orientation of the quadcopter relative to the Earth frame can be described using Euler angles (roll, pitch, yaw) and transformed into a rotation matrix. This matrix can then be used to convert velocity, force, and acceleration vectors between the two frames [33,34].

The origin of the Body frame, denoted as Oβ, coincides with the quadcopter's center of gravity (CG). The axes of the Body frame are aligned with the quadcopter's principal axes.

x-axis: Points forward along the quadcopter's nose.

y-axis: Points to the right of the quadcopter.

z-axis: Points downward through the center of gravity, perpendicular to the quadcopter's plane.

In quadcopter dynamics, Euler angles describe the orientation of the quadcopter relative to a fixed reference frame. The three basic rotations, known as φ, θ, and ψ, correspond to rotations around the x, y, and z axes of the quadcopter's body-fixed frame [35,36].

Rotation Around the x-axis (Roll, φ): This rotation tilts the quadcopter left or right. The corresponding rotation matrix $R_x(\phi)$is:

$$ {}_{n}^{b}R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad (1)$$

Rotation Around the y-axis (Pitch, θ): This rotation tilts the quadcopter forward or backward. The corresponding rotation matrix $\boldsymbol{R_y}(\theta)$ is:

$$ {}_{n}^{b}R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2)$$

Rotation Around the z-axis (Yaw, ψ): This rotation turns the quadcopter left or right around its vertical axis. The corresponding rotation matrix $R_z(\psi)$ is:

$$ {}_{n}^{b}R_z(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix}$$

(3)

To transform coordinates from the inertial (Earth) frame to the body-fixed frame, we apply these rotations in sequence: yaw (ψ), then pitch (θ), and finally roll (φ). The combined transformation matrix $R_{bodxy-to-inertial}$ is the product of these individual rotations:

$$R_{body-to-inertial} = R_x(\phi).\,R_y(\theta).\,R_z(\psi) \quad (4)$$

Given that the inertial frame can be perceived as rotating in the opposite direction from the body-fixed frame's perspective, the inverse of this transformation matrix is applied when converting from the inertial frame to the body-fixed frame [37]:

$$R_{inertial-to-body} = R^{-1}_{body-to-inertial} = R_z(-\psi).R_y(-\theta).R_x(-\phi) \quad (5)$$

This transformation allows us to express vectors in the inertial frame as coordinates in the body-fixed frame and vice versa, making it possible to analyze and control the quadcopter's motion in either reference frame. The complete transformation matrix is:

$$\boldsymbol{R_e^b} = \begin{bmatrix} c\,\theta\, c\,\psi & c\,\psi\, s\,\theta\, s\,\phi - s\,\psi\, c\,\phi & c\,\psi\, s\, s\,\psi\, s\,\phi \\ c\,\theta\, s\,\psi & s\,\psi\, s\,\theta\, s\,\phi + c\,\psi\, c\,\phi & s\,\psi\, s\,\theta\, c\,\phi - c\,\psi\, s\,\phi \\ -s\,\theta & c\,\theta\, s\,\phi & c\,\theta\, c\,\phi \end{bmatrix} \quad (6)$$

The angular rate in the body frame is given by b_ω.

$b_\omega = [\omega_{xb}\ \omega_{yb}\ \omega_{zb}]^T$

These components describe the rotation rates around the body axes.

To analyze or control the motion of the system with respect to a fixed (inertial) reference frame, we need to transform the angular velocity components from the body frame to the Earth (inertial) frame. This transformation involves using Euler angles to rotate the body-fixed frame with respect to the Earth frame [38].

After rotating around each axis, the angular velocity in the body frame is expressed as:

$$b_\omega = \dot{\psi}.k_3 + \dot{\theta}.n_2 + \dot{\phi}.b_1 \quad (7)$$

This equation states that the angular velocity is a combination of the rates of change of the Euler angles multiplied by unit vectors in the respective directions of the rotations. Specifically [39]:

- $\dot{\psi}.k_3$ represents the rotation about the yaw axis,
- $\dot{\theta}.n_2$ represents the rotation about the pitch axis,
- $\dot{\phi}.b_1$ represents the rotation about the roll axis.

The rotation matrices that relate the different frames are given by:

$$R_b^n = R_z(\phi),\ R_b^k = R_y(\phi)R_z \quad (8)$$

These matrices define how to rotate from one frame to another:

R_b^n is the rotation matrix for the roll angle ϕ.

R_b^k is the combined rotation matrix for both pitch θ and roll ϕ.

By substituting the rotation matrices into the angular velocity equation, we get:

$$b_\omega = \dot{\psi}.R_z(\phi).R_y(\theta) + \dot{\theta}.R_z(\phi) + \dot{\phi} \quad (9)$$

We introduce the body rates p, q, r, which are the components of angular velocity in the body frame:

$$b_\omega = [p\ q\ r]^T \quad (10)$$

Therefore, the relationship between the Euler angles' rates of change and the body rates is given by:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (11)$$

This matrix equation expresses the body rates in terms of the rates of change of the Euler angles. Finally, we express the angular velocity vector in terms of a rate matrix W and the Euler angle rates [40]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W.\,b_\omega \quad (12)$$

The matrix W is a function of the Euler angles and describes how the angular velocity components relate to the Euler angle rates. The matrix W is given by:

$$\theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix},\ W = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}$$

This matrix helps convert angular velocities to body rates and is essential in the control and simulation of the quadcopter's motion as shown in Figure 2.

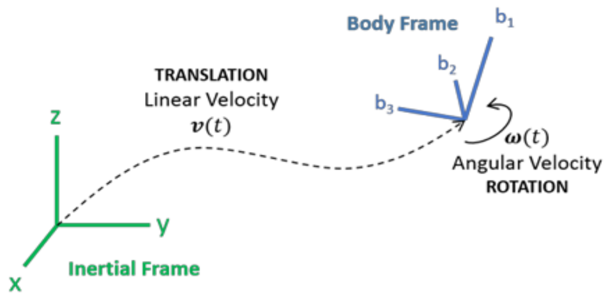

Figure 2. Modelling quadcopter's dynamics

For rotational motion, torque τ_b is defined by the time derivative of angular momentum L.

$$\tau_b = \frac{\mathrm{dt}}{\mathrm{dL}} = \frac{d}{dt}(\mathrm{J}\cdot\omega)$$

(13)

Where:

- J is the inertia tensor of the multi-copter.
- ω is the angular velocity.

Assuming a symmetric mass distribution, J simplifies to:

$$\mathrm{J} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}$$

The angular dynamics equation becomes:

$$\tau_b = \mathrm{J}\cdot\dot{\omega} + \omega\times(\mathrm{J}\cdot\omega) \quad (14)$$

Expanding this for each axis:

$$\tau_x = (J_z - J_y)qr + J_x\dot{p}$$
$$\tau_y = (J_x - J_z)pr + J_y\dot{q}$$
$$\tau_z = (J_y - J_x)pq + J_z\dot{r}$$

To model the quadcopter's translational motion based translational kinematics:

- Let $p = [x\ y\ z]^T$ represent the position of the multi-copter in the Earth's frame. The time derivative of position $\dot{p}$.
- In the body frame: $\dot{p} = R_e^b$ .v, where R_e^b is the rotation matrix from the Earth frame to the body frame.

The control effectiveness matrix relates the thrust and moments generated by the propellers to the forces and torques applied to the quadcopter. The control effectiveness matrix for a "plus" configuration quadcopter is given by [15, 41]:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ 0 & -dC_T & 0 & dC_T \\ dC_T & 0 & -dC_T & 0 \\ C_M & -C_M & C_M & -C_M \end{bmatrix} = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (15)$$

This matrix expresses how the propeller speeds ω_i contribute to the overall thrust and moments acting on the quadcopter, allowing for control and stabilization of the vehicle's flight.

The nonlinear model describes the quadcopter's dynamics by gathering the system's states. These states interact based on the rigid dynamics and kinematics model, represented by matrix equations [21].

To simplify the nonlinear model, linearization around an equilibrium point, specifically the hovering point, is performed. The equilibrium point X_e, U_e is defined as:

$X_e = [0,0,0,0,0,0,0,0,0,0,0,0]^T$ *and* $U_e = [mg, 0,0,0]^T$
m is the mass and g is the gravitational constant.

These points are used in Taylor series expansion to linearize the nonlinear model. The higher-order terms in Taylor's series are neglected for small changes around the equilibrium point, leading to a linear state-space model. The linear state-space model is then obtained by rewriting the nonlinear equations into a linear form [22]:

$$\bar{\dot{x}} = f(xe, ue) + \partial \frac{\partial}{\partial x} \partial x\, f(x,u) \Big|_{\substack{x = xe \\ u = ue}} (\vec{x} - xe) + \frac{\partial}{\partial x} f(x,u) \Big|_{\substack{x = xe \\ u = ue}} (\vec{u} - ue) \quad (16)$$

At higher order Terms can be neglected for small change around equilibrium point yields linear state space model. The matrices A and B are the Jacobian matrices calculated using partial derivatives:

$$A = [\frac{\partial f}{\partial x}]_{\substack{xe \\ ue}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x^*_1, \dots, x^*_n) & \cdots & \frac{\partial f_1}{\partial x_n}(x^*_1, \dots, x^*_n) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_1}(x^*_1, \dots, x^*_n) & \cdots & \frac{\partial f_n}{\partial x_n}(x^*_1, \dots, x^*_n) \end{bmatrix} \quad (17)$$

$$B = [\frac{\partial f}{\partial u}]_{\substack{xe \\ ue}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(x^*_1, \dots, x^*_n) & \cdots & \frac{\partial f_n}{\partial u_1}(x^*_1, \dots, x^*_n) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1}(x^*_1, \dots, x^*_n) & \cdots & \frac{\partial f_n}{\partial u_n}(x^*_1, \dots, x^*_n) \end{bmatrix} \quad (18)$$

Finally, the nonzero partial derivatives are only considered to simplify the model, leading to the matrices A and B.

III. LINEAR QUADRATIC REGULATOR CONTROL

Optimal control involves the process of determining suitable control signals for a system, taking into consideration physical constraints and optimizing a cost or performance measure. The primary objective of this approach is to solve an optimization problem that guides the system's state (x(t)) towards a desired trajectory ($x(t)_d$), while simultaneously minimizing costs and efficiently utilizing control inputs (actuators). To accomplish this objective, several key factors need to be addressed [3,42]:
Development of a precise model that accurately describes the behavior of the dynamic system under control.
Definition of a cost function (J) that incorporates the specific requirements and specifications outlined by the designer.
Consider a dynamic system where the state is represented by x and the input is represented by u:

$$\dot{x}(t) = f[x(t), u(t), t] \quad (19)$$

The cost function defines as,

$$J = e[x(t_f)] + \int_{t_0}^{t_f} w[x(t), u(t), t]\, dt \quad (20)$$

The boundary conditions are as follows:
$x(t_0) = x_0$.
$x(t_f)$ is unconstrained, and t_f can take any value.
Based on this, an optimization problem can be defined to find the solution:

$$u(t), \forall\, t \in [t_0, t_f] \quad (21)$$

The objective of the optimization problem is to minimize the cost index J. As the time interval approaches infinity, the system and cost index can be expressed as follows [43]:

$$\dot{x} = Ax + Bu \quad and \quad y = Cx$$

Where A is a system matrix, B is an input matrix and C is output matrix.

$$J = \int_{t_0}^{\infty} \{ u(t)^T.\ R.u(t) + [x(t) - x_d(t)]^T.Q.[x(t) - x_d(t)]^T \}\, dt \quad (22)$$

R and Q are matrices representing the costs associated with the control inputs and system state, respectively. It has been shown that the control input u(t) that minimizes the cost functional is a state linear feedback, which can be expressed as:

$$u(t) = -K\ .\ [x(t) - x_d(t)] \quad (23)$$
$$K = R^{-1}\ .B^T\ .S$$

The positive definite matrix S is a solution to the algebraic Riccati equation [44].

$$S.A + A^T.S - S.BR^{-1}.B^T.S + C^T.Q.C = 0 \quad (24)$$

IV. MODEL PREDICTIVE CONTROL (MPC)

MPC is a sophisticated control approach that has become well-known for its ability to effectively manage dynamic, complex systems while staying within restrictions. The fundamental idea behind MPC is using a model to predict how a system will behave in the future so that control actions can be optimized based on these predictions.
This approach enables the handling of various operational challenges, making it suitable for a wide range of applications [15]. This approach is characterized by solving an optimization problem at each control step, where the objective is to minimize a predefined cost function over a future time horizon while adhering to system constraints.
Continuous real-time optimization of a mathematical model of the system is the foundation of MPC. By using this model, MPC predicts how the system will behave in the future, which helps the optimization process determine the best course for the controlled variable u (see Fig. 3). As a result, MPC offers an intuitive way to parameterize by fine-tuning the process model, although this involves greater computational demands compared to traditional controllers [45].

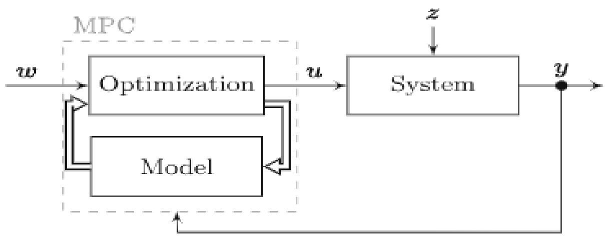

Figure 3. Simplified block diagram of MPC-based control loop

At every time step, MPC determines the optimal control inputs by solving an optimization problem designed to minimize a cost function. This function usually consists of components related to tracking errors and control efforts. Additionally, the problem includes constraints on system states and control inputs, ensuring that the control actions stay within acceptable boundaries. The calculated control inputs are applied to the system, but only the first input from the optimized sequence is implemented; this process is then repeated at the subsequent time step using updated state information [46].

It uses the current state to solve an optimal control problem over a finite horizon at each sample period. As shown in Fig. 4, the first action from this optimal control sequence is carried out, and the procedure is repeated using the new measured (or estimated) state at the subsequent sampling time. Typically, the performance measure used is quadratic, incorporating penalties for both the control inputs and the system states. Consequently, MPC can often be regarded as a moving horizon or receding horizon control (RHC) problem, where each sampling instant involves solving a finite horizon constrained linear quadratic (LQ) problem [47].

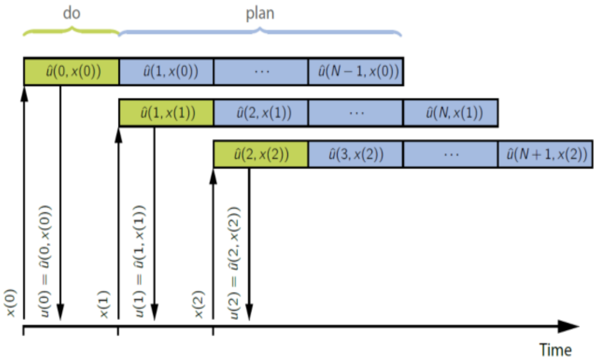

Figure 4. The receding-horizon principle [17]

In the context of quadcopters, MPC offers significant advantages due to its ability to manage the nonlinear dynamics and constraints associated with aerial vehicles [48]. Quadcopter dynamics are inherently complex, involving interactions between thrust, torque, and aerodynamic forces. MPC can effectively handle these nonlinearities either by using a nonlinear model or by linearizing the model around a specific operating point [12]. One of the notable benefits of MPC in quadcopter control is its capability to manage constraints directly. Quadcopter operations often involve various constraints, such as maximum thrust limits, battery life, and altitude boundaries. MPC ensures that these constraints are respected by incorporating them explicitly into the optimization problem, thereby improving safety and operational efficiency [49].

At each time step, an optimization problem must be created and resolved as part of the mathematical formulation of MPC. In MPC, the cost function is typically designed to guarantee that, over a prediction horizon N2, the system output y will match a specified reference r (see Fig. 5). At each time step, this prediction and optimization process is repeated, with the system only receiving the first value from the optimized output path. MPC is commonly referred to as "receding horizon" control because of this iterative process. The underlying idea is that short-term predictive optimization can yield optimal performance over an extended period, based on the premise that errors in near-term forecasts are minimal compared to those further out. A key difference between MPC and traditional control methods lies in its integration of prediction and optimization, rather than depending on precomputed control laws. To capture the impact of changes in the manipulated variable u on the control variable y, the prediction horizon N needs to be long enough. Either a shorter prediction horizon N1 or the incorporation of delays into the system model—the latter being frequently more intuitive can be used to manage delays. In order to account for computing time, the shorter prediction horizon is typically set to N1=1, which means that the solution u is only used at the subsequent time step [18].

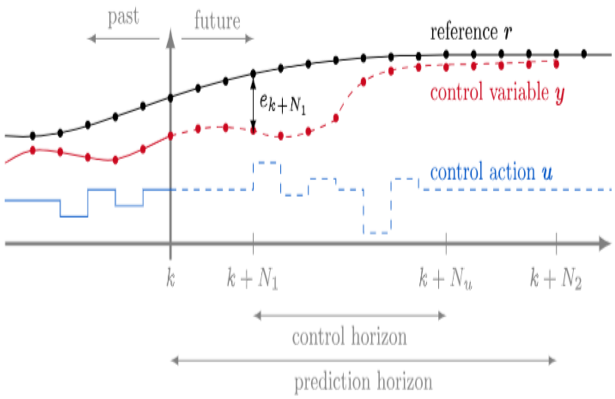

Figure 5. Function principle of a model-based predictive with horizons

Subject to the restrictions and dynamics of the system, the goal is to minimize a cost function across the prediction horizon. A state-space representation of the system is necessary for MPC to function. The following linear equations are commonly used to characterize the state-space model [20,50].

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k) \tag{25}$$

The optimization problem can be expressed as:

$$J = \sum_{i=0}^{N_p-1} [x(k+i)^T Q x(k+i) + u(k+i)^T R u(k+i))$$

Subject to:

$$x\left(k+i+\frac{1}{k}\right) = Ax\left(k+\frac{i}{k}\right) + Bu(k+\frac{i}{k})$$

$$x_{min} \le x(k+i) \le x_{max}, u_{min} \le u(k+i) \le u_{max}$$

where:

- Np is the prediction horizon.
- Q and R are weighting matrices that define the relative importance of tracking error and control effort.
- x_{min}, x_{max}, u_{min}, and u_{max} are the state and input constraints.

MPC operates on the receding horizon principle: after applying u(k), the process repeats at the next time step with updated state measurements. This ensures that the control strategy is continuously optimized based on the latest available information [51-55].

V.SIMULATION RESULTS

This section presents the results of the quadcopter control system simulation using MPC and LQR. The performance of the system is evaluated in terms of open-loop response, closed-loop response with MPC, and the effect of noise on the system's behavior. The open-loop response of the quadcopter system provides a baseline for evaluating the effectiveness of the MPC controller. In an open-loop configuration, there is no feedback control applied, meaning the system is subject to its natural dynamics without any corrective action.

Figure 6 plot shows the open-loop response of the quadcopter in terms of its position (x, y, z) and orientation angles (ϕ, θ, ψ). The open-loop response indicates that the system cannot maintain stability on its own, particularly in terms of controlling its position and angels. This reinforces the necessity of implementing a closed-loop control strategy, such as MPC, to achieve desired performance. The closed-loop response with MPC as shown in Fig. 7, demonstrates the effect of applying the MPC controller to the quadcopter system. The MPC controller is designed to track the desired reference trajectory while respecting the physical constraints of the system. The simulation runs for 20 seconds, starting from rest (x_0 = zeros (12, 1)), with the goal of reaching and maintaining a reference position of 1 meter in x, y, and z, with no rotation (zero angles).

- ***Position (x, y, z) over time***

The positions (x, y, z) quickly rise from 0 to 1 meter and stabilize. This corresponds to the quadcopter moving to the desired position of 1 meter in all three spatial dimensions. The rapid stabilization is due to the controller's effort to minimize the difference between the current position and the target (1 meter). The positions (x, y, z) quickly reach a steady-state value of approximately 1 meter and remain constant, indicating a stable position response. The response settles within approximately 3 seconds. The higher weight on the position in the MPC design prioritizes accurate position tracking, which is why the system reaches the desired position quickly and holds it steady.

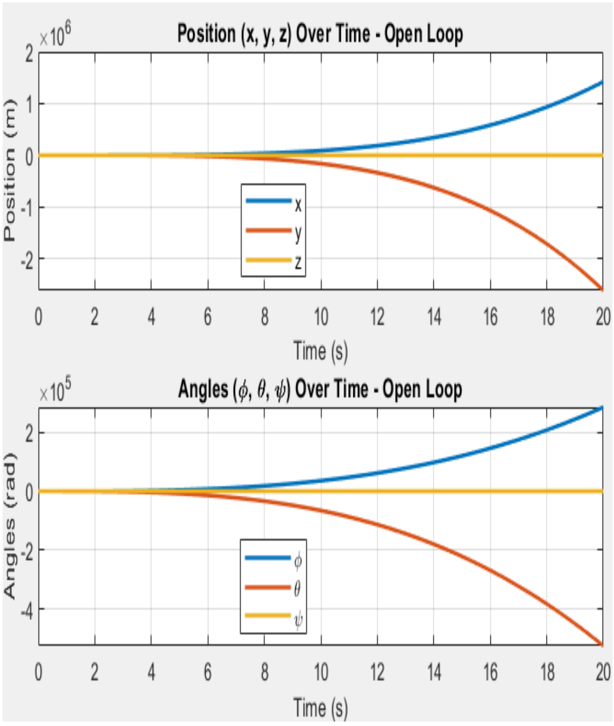

Figure 6. Open loop response

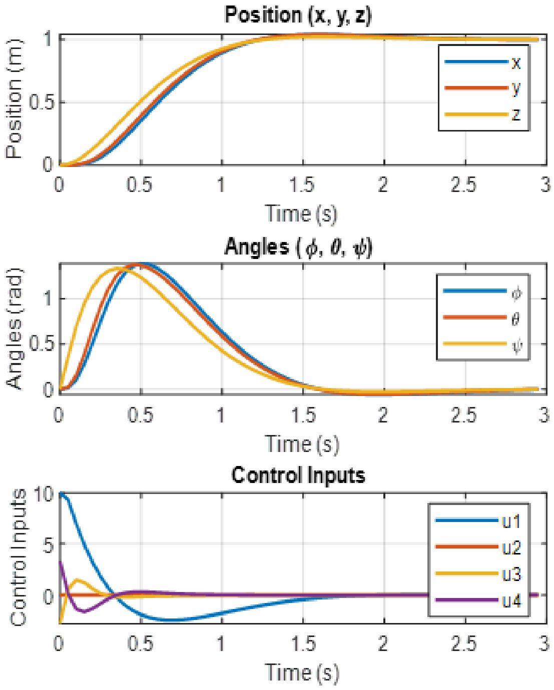

Figure 7. Response of closed loop system

- ***Angles (ϕ, θ, ψ) over time:***

The angles ϕ, θ, ψ initially spike and then quickly return to zero. This represents the quadcopter's orientation adjusting during the initial movement and then stabilizing to maintain the desired level flight. The angles initially exhibit a sharp deviation before quickly stabilizing at zero. This transient behavior suggests that the system experiences an initial disturbance before the controller successfully returns the angles to their desired positions. The rapid settling indicates that the MPC controller effectively reduces oscillations and achieves stability. The transient spikes in the angles reflect the quadcopter's response to the control inputs that move it to the target position. The system quickly dampens these spikes due to the controller's designed response to minimize deviations in orientation.

The control inputs initially show spikes, corresponding to the strong corrective actions needed to move the quadcopter to its desired position and stabilize its angles. After the initial phase, the inputs stabilize at lower values. The initial spikes are necessary to overcome the system's inertia and achieve the desired position and orientation. The controller then reduces the control input magnitudes once the system is close to the target, maintaining stability with minimal effort. The Fig. 8 and Fig. 9 shows the trajectory of the quadcopter in space, where the reference trajectory is a path from the initial position (x, y, z) = (0, 1, 3) to the final position (x, y, z) = (2, 2, 5), x-Axis (x in meters): - Ranges from 0 to 2 meters. The quadcopter is moving along the x-axis from its initial position at 0 meters to 2 meters, y-Axis (y in meters): - Ranges from 0 to 1 meter. The reference starts from 1 meter, and the quadcopter eventually moves towards 2 meters in the y direction, z-Axis (z in meters): - Ranges from 0 to 5 meters. The quadcopter is moving from an initial height of 3 meters to a final height of 5 meters.

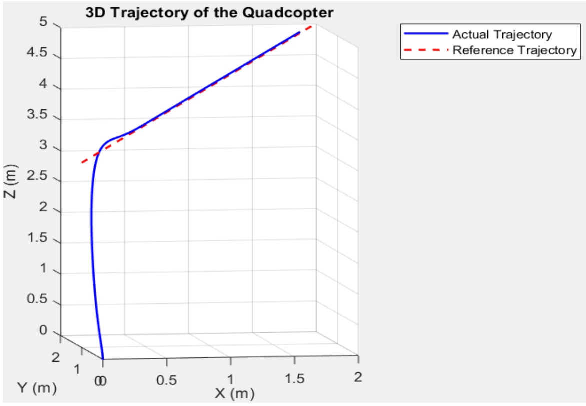

Figure 8. 3D Trajectory of the Quadcopter.

The actual trajectory closely follows the reference trajectory. Initially, there is a slight deviation as the quadcopter stabilizes and corrects its path. After this adjustment, the trajectory closely aligns with the reference, demonstrating the effectiveness of the controller in guiding the quadcopter along the desired path.

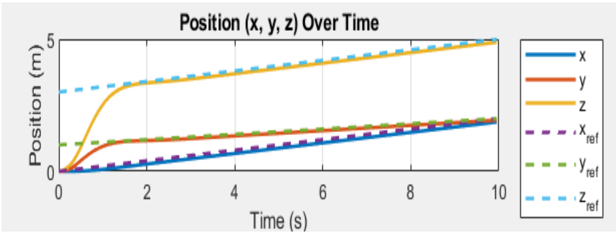

Figure 9. Trajectory of Position (x, y, z)

All three positions (x, y, z) follow the reference trajectories with minimal lag or overshoot. The controller effectively adjusts the quadcopter's position over time to follow the reference path. So, The MPC controller effectively stabilizes the quadcopter, allowing it to track the desired position and orientation accurately. The response is smooth, with minimal overshoot and a fast-settling time, demonstrating the robustness of the MPC strategy. To quantify the performance improvement provided by the MPC controller, key metrics such as rise time, settling time, and overshoot are computed for the position of the quadcopter. These metrics are summarized in Table II.

TABLE II. THE PERFORMANCE OF QUADCOPTER USING MPC

Axis	Rise Time (s)	Settling Time (s)	Overshoot (%)
x	0.72	2.091	4.143
y	0.72	2.063	3.9725
z	0.77	1.745	2.1275

The MPC controller exhibits a rapid rise time and settling time, with minimal overshoot across all axes. This indicates that the controller effectively manages the quadcopter's dynamics, ensuring prompt and precise tracking of the reference signals.

Figure 10 illustrates the spiral quadcopter's trajectory after applying the MPC strategy. The next step involves tracking this predefined path, where a smooth trajectory is essential for stable and efficient quadcopter operation.

To assess the effectiveness of the MPC controller, a 3D reference trajectory will be used. By comparing the quadcopter's actual path with the desired trajectory as shown in Fig. 11-14, we can evaluate key performance metrics such as:

- Tracking accuracy (deviation from the reference path)
- Response time (how quickly the quadcopter adjusts to changes).
- Stability (smoothness of motion without oscillations)

The results will help determine the MPC's robustness in handling complex maneuvers, ensuring optimal performance in real-world flight scenarios.

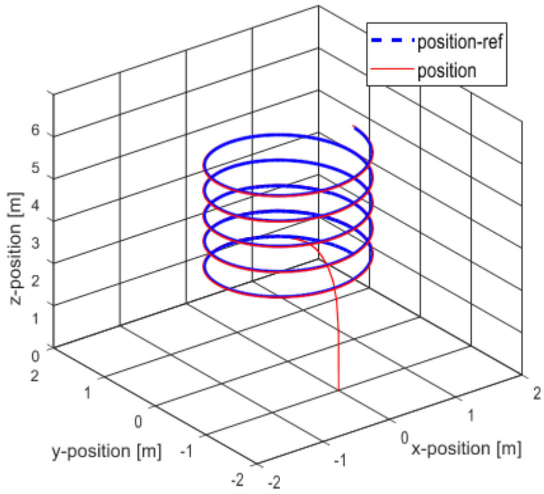

Figure 10. Spiral trajectory

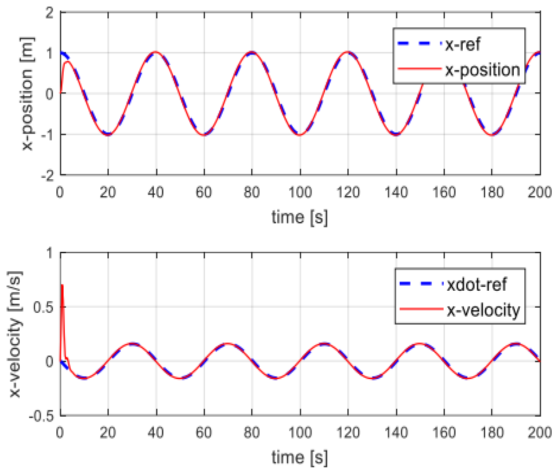

Figure 11. Response of position and velocity-based x-axis

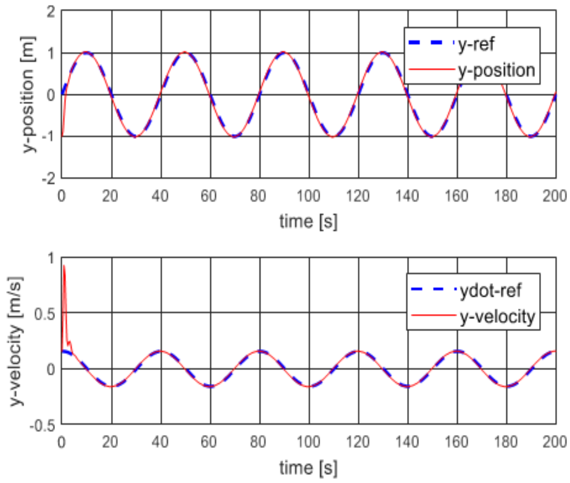

Figure 12. Response of position and velocity-based y-axis

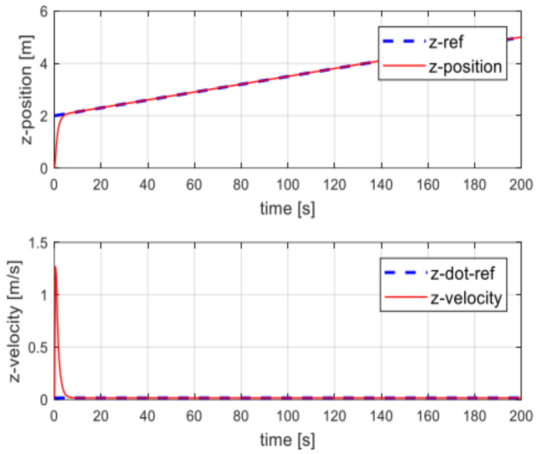

Figure 13. Response of position and velocity-based z-axis

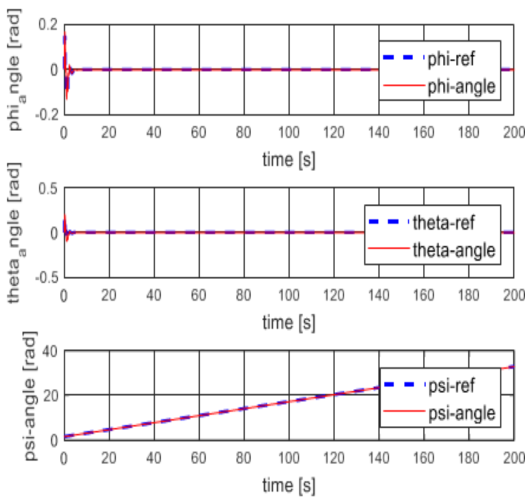

Figure 14. φ, θ, ψ values as a function of time

In the simulation of quadcopter dynamics, incorporating noise is essential for evaluating the robustness of the control system under realistic and challenging conditions. Noise represents random disturbances that can affect the system's performance, such as environmental factors like wind or sensor inaccuracies. To assess the robustness of the MPC controller, noise was introduced into the system to simulate real-world disturbances. This section details the introduction of noise into the simulation, its impact on the quadcopter's performance, and the MPC controller's ability to compensate for these disturbances.

For this simulation, Gaussian noise is used to model these random disturbances. Gaussian noise is characterized by its mean and standard deviation, where the standard deviation determines the intensity of the noise. By introducing Gaussian noise, we aim to replicate the effect of unpredictable environmental conditions on the quadcopter, providing a more rigorous test of the control system's resilience. One of the most common sources of disturbance for quadcopters in real-world scenarios is wind. Wind exerts aerodynamic drag on the quadcopter, which can cause deviations from its desired trajectory. The drag force F_d due to wind can be described by the following equation [26].

$$F_d=\frac{1}{2}\,\rho C_d A v^2.$$

- ρ is the air density, approximately 1.225 kg/m^3 at sea level.
- C_d is the drag coefficient, assumed to be 1.0 for the Quadcopter.
- A is the cross-sectional area facing the wind, assumed to be 0.1m^2.
- v is the wind speed in meters per second (m/s).

In this context, the noise level in the simulation is related to the disturbances caused by wind. A higher wind speed results in a greater drag force, which is reflected in the increased noise level in the simulation. To estimate the wind speed that corresponds to a noise level of 0.1 in the simulation, we can relate the positional deviation Δx caused by the drag force F_d to the quadcopter's dynamics:

$$\Delta x = \frac{F_d \cdot \Delta t^2}{m}$$

m is the mass of the quadcopter (1.38 kg).
Δt is the simulation time step (0.05 seconds).
Determining noise level for a specific wind speed (50 km/h).

$$F_d=\frac{1}{2}\times 1.225\times 1.0\times 0.1\times (13.89)^2 \approx 11.82 \text{ N}$$

$$\text{Noise level} \approx \frac{F_d \cdot 0.05^2}{1.38} \approx 0.021$$

A wind speed of 50 km/h corresponds to a noise level of approximately 0.021.

The MPC controller demonstrates robust performance even in the presence of noise as shown in Figures 15 and 16. While noise introduces minor deviations in the response, the controller quickly compensates, maintaining stability and keeping the quadcopter on its desired trajectory. This showcases the MPC's capability to handle disturbances effectively, making it a reliable control strategy for real-world applications.

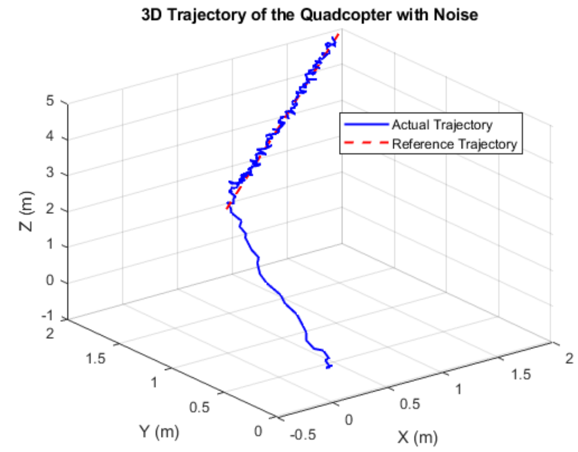

Figure 15. 3D Trajectory of the Quadcopter with noise effect.

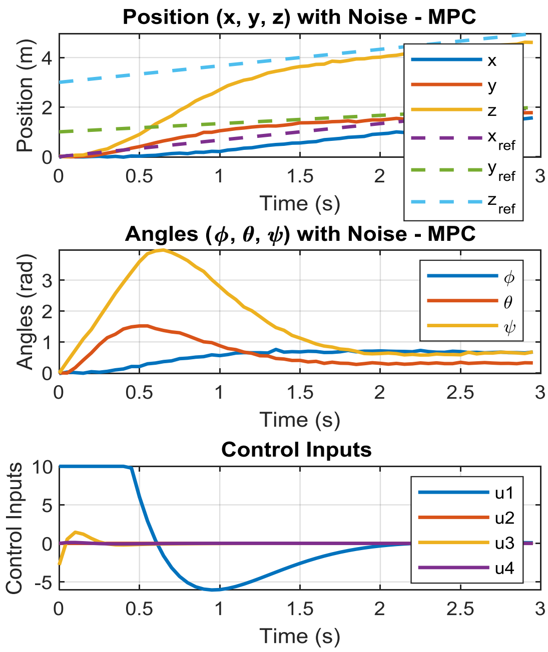

Figure 16. The system's behavior in the presence of noise.

The introduction of noise with a standard deviation of 0.021 provides a more rigorous test of the MPC controller’s robustness. This noise level introduces moderate disturbances that challenge the system's ability to maintain accurate control and follow the desired trajectory. By analyzing the simulation results with this noise level, we gain valuable insights into how effectively the MPC controller can compensate for significant disturbances, ensuring its reliability and performance in real-world conditions.

To create an optimal state feedback controller using the LQR for comparison with MPC, it is essential to define the cost matrices Q and R [3]. The diagonal elements of Q indicate the penalties associated with the respective state variables, while the diagonal elements of R reflect the penalties for the control inputs. The time response of the closed-loop system can be simulated, as illustrated in Fig. 17, utilizing the LQR controller The evolution of the state system over time, depicted in Fig. 18, can be analyzed using tools like MATLAB's Simulink. The weighting matrix R is vital for influencing the performance of the LQR controller. By adjusting the values of R, we can fine-tune the controller's focus on minimizing control inputs while still maintaining the desired performance levels. In the case of the quadcopter, modifications to the weighting matrix R can lead to significant changes in its behavior [3].

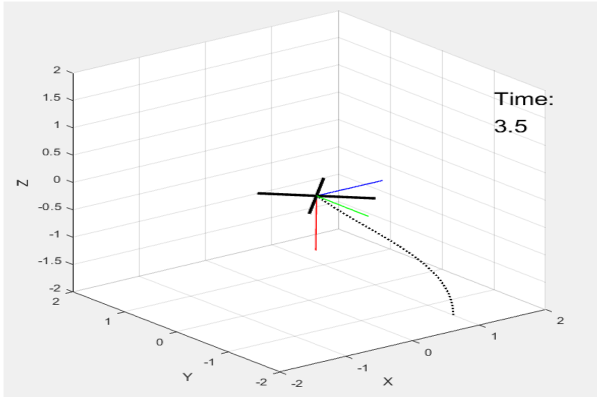

Figure 17. 3D response using LQR [3]

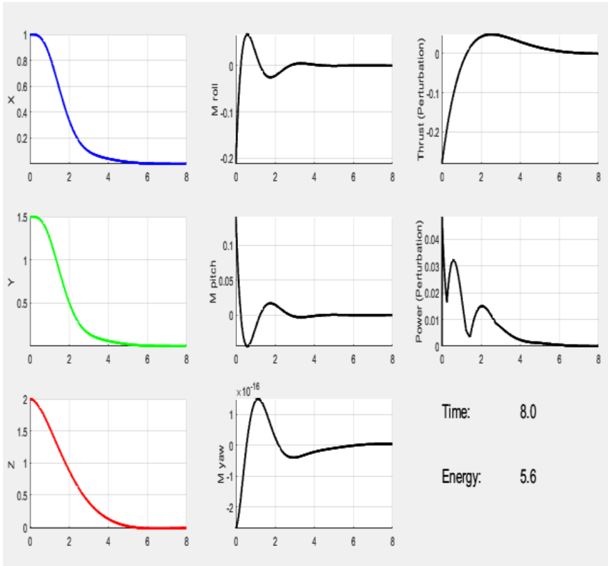

Figure 18. States system using LQR [3]

Table III provides a comparison of the quadcopter's behavior based LQR and MPC. In summary, this paper provided a comprehensive analysis of the quadcopter control system's performance using MPC and LQR. MPC is superior for handling constraints, tracking complex trajectories, and adapting to dynamic environments but is computationally heavier. LQR is simpler and less resource-intensive, making it suitable for linear systems with predictable behavior but may struggle with nonlinearities and constraints.

TABLE III. COMPARISON OF MPC AND LQR FOR QUADCOPTER CONTROL

Feature Model	MPC	LQR
Handling Constraints	Explicitly incorporates constraints on inputs and states, making it suitable for complex environments.	Does not inherently handle constraints; constraints must be managed separately.
Trajectory Tracking	More effective for tracking complex trajectories due to its predictive nature.	Can handle linear trajectories well, but may struggle with highly nonlinear paths.

Adaptability	Adapts to changing conditions by re-evaluating the control inputs at each time step.	Less adaptable; the controller is designed for a specific set of conditions.
Computational Complexity	Typically requires more computational resources due to real-time optimization calculations.	Generally, less computationally intensive, as it involves simpler calculations.
Performance with Nonlinear Dynamics	Better suited for nonlinear systems, as it can model and predict future states.	Assumes linear dynamics; performance may degrade with significant nonlinearities.
Implementation	More complex to implement due to the need for optimization algorithms and a predictive model.	Easier to implement with a straightforward design process based on state space.
Tuning	Requires careful tuning of prediction horizons and cost matrices to achieve desired performance	Tuning focuses on the weighting matrices Q and R, which can be simpler but still requires careful consideration.
Robustness	Can be more robust to disturbances and uncertainties due to its predictive nature.	May be less robust in the face of significant disturbances unless designed with robustness in mind.

VI. CONCLUSION

The robustness of MPC was demonstrated by effectively stabilizing and controlling a quadcopter, even in challenging scenarios involving noise and environmental disturbances. MPC was validated for maintaining the quadcopter's intended trajectory with high precision, highlighting its reliability for UAV applications. The advantages of a closed-loop system were emphasized, showcasing MPC's critical role in maintaining stability and performance by adapting to varying conditions.

This confirms MPC's potential and practical suitability for real-world UAV deployments, particularly in unpredictable environments. For the future work, integrate machine learning algorithms to enable the MPC system to self-tune its parameters based on real-time environmental inputs, enhancing its adaptability and responsiveness. In practice, the choice between MPC and LQR will depend on the specific application requirements, computational resources, and the expected operating conditions of the quadcopter.

REFERENCES

[1] N. Roy and P. Abbeel, Construction of Cubic Structures with Quadrotor Teams, MITP, pp. 177-184, 2012.

[2] J. Turner, P. Kenkel, R. B. Holcomb and B. Arnall, "Economic Potential of Unmanned Aircraft in Agricultural and Rural Electric Cooperatives", Southern Agricultural Economics Association 2016 Annual Meeting, February 6-9, 2016.

[3] H. Almgallesh, et. al., Dynamics and Optimal Control of Quadcopter, 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), 2024.

[4] A. Abougarair, Adaptive Neural Networks Based Optimal Control for Stabilizing Nonlinear System , 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA2023), 2023.

[5] A. Elmolihi, et. al., Robust Control and Optimized Parallel Control Double Loop Design for Mobile Robot , IAES International Journal of Robotics and Automation (IJRA), vol. 9, no. 3, 2020.

[6] A. Zrigan, et. al., Optimized PID Controller and Generalized Inverted Decoupling Design for MIMO System, 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET'2023), 2023.

[7] A. Abougarair, Neural Networks Identification and Control of Mobile Robot Using Adaptive Neuro Fuzzy Inference System, ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020, 2020.

[8] A. Abougarair, Model Reference Adaptive Control and Fuzzy Optimal Controller for Mobile Robot, Journal of Multidisciplinary Engineering Science and Technology, pp 9722-9728, vol. 6, issue 3, 2019.

[9] A. Oun, et. al., Model Predictive Control for Optimizes Battery Charging Process, 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), 2024.

[10] M. Ramos and M. Gupta, Trajectory Tracking for Quadrotor UAV Using Model Predictive Control, Journal of Aerospace Engineering, vol. 30, no. 4, pp. 04017034, 2017.

[11] G. Kokunko, A. rasnova & A. Utkin, Synthesis of Robust Quadcopter Control Algorithms Considering Speeds and Lift Force Constraints. Мехатроника, Автоматизация, Управление, 26(1), 39–52, 2025. https://doi.org/10.17587/mau.26.39-52.

[12] M. Pal, S. Mondal, & N. Basu, Trajectory tracking control of quadrotor UAV with uncertainties and actuator saturation: An event driven approach. Measurement & Control, 2025. https://doi.org/10.1177/00202940241312863

[13] X. Huang, Research of Different Flight Controls and Methods for Quadrotor Aircraft. Applied and Computational Engineering, 127(1), 116–121, 2025. https://doi.org/10.54254/2755-2721/2025.20259.

[14] S. Kumar, Robust trajectory tracking of an uncertain quadrotor with almost global finite-time convergence. Proceedings of The Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2025. https://doi.org/10.1177/09544100241309050.

[15] M. Izadi, et. al., High-Performance Model Predictive Control for Quadcopters with Formal Stability Guarantees, 2024. https://doi.org/10.48550/arxiv.2412.20277.

[16] Y. Li, et al., Quadcopter Trajectory Tracking Based on Model Predictive Path Integral Control and Neural Network. Drones, 9(1), 9, 2024. https://doi.org/10.3390/drones9010009.

[17] R. Mousavi, et. al., Observer-Based Adaptive Neural Control of Quadrotor Unmanned Aerial Vehicles Subject to Model Uncertainties and External Disturbances. Actuators, 13(12), 529, 2024. https://doi.org/10.3390/act13120529.

[18] R. Derakhshan, & M. Danesh, Trajectory tracking of a quadrotor using model predictive and backstepping control with anti-disturbance compensation subjected to variations in the center of gravity. Transactions of the Institute of Measurement and Control. ,2024. https://doi.org/10.1177/01423312241291262.

[19] V. Nguyen, et. al., Trajectory Tracking Control for a Quadcopter under External Disturbances. Engineering,

Technology & Applied Science Research, 14(6), 17620–17628, 2024. https://doi.org/10.48084/etasr.8449.

[20] S. Kumar, & L. Dewan, Trajectory tracking controller design for quadcopter under disturbances environment: using a hybrid approach. Cybernetics and Physics, 13(3), 248–260, 2024. https://doi.org/10.35470/2226-4116-2024-13-3-248-260.

[21] Y. Alqudsi, Advanced Control Techniques for High Maneuverability Trajectory Tracking in Autonomous Aerial Robots. 1–8, 2024. https://doi.org/10.1109/iceti63946.2024.1077725.

[22] A. Gedefaw, An Improved Trajectory Tracking Control of Quadcopter Using a Novel Sliding Mode Control with Fuzzy PID Surface. PLOS ONE, 19(11), e0308997, 2024. https://doi.org/10.1371/journal.pone.0308997.

[23] N. Canh, et. al., Quadrotor Trajectory Tracking Using Linear and Nonlinear Model Predictive Control, 2024. https://doi.org/10.48550/arxiv.2411.06707.

[24] J. Kuang, et. al., Adaptive Sliding Mode Control for Trajectory Tracking of Quadrotor Unmanned Aerial Vehicles Under Input Saturation and Disturbances. Drones, 8(11), 614, 2024. https://doi.org/10.3390/drones8110614.

[25] L. Hu, et. al., Quadcopter Unmanned Aerial Vehicle Formation Trajectory Tracking based on Extended State Observer with Model Predictive Control. 240–246, 2024. https://doi.org/10.1109/rcae62637.2024.10834167

[26] J. Li, et. al., A Novel Robust Hybrid Control Strategy for a Quadrotor Trajectory Tracking Aided with Bioinspired Neural Dynamics. Applied Sciences, 14(20), 9592, 2024. https://doi.org/10.3390/app14209592.

[27] Z. Liu, et. al., Autonomous Navigation Control of a Quadcopter Based on MPC. 1603–1608, 2024. https://doi.org/10.1109/icus61736.2024.1083990.

[28] O. Moali, et. al., UAV Trajectory Tracking Using Proportional-Integral-Derivative-Type-2 Fuzzy Logic Controller with Genetic Algorithm Parameter Tuning. Sensors, 24(20), 6678, 2024. https://doi.org/10.3390/s24206678.

[29] M. Karahan, Reinforcement Learning and PD Control Based Trajectory Tracking for a Quadcopter UAV. Journal of Computer Science and Technology Studies, 6(4), 131–141, 2024. https://doi.org/10.32996/jcsts.2024.6.4.15.

[30] M. Yang, et. al., Adaptive Optimization Tracking Control for an Unmanned Aerial Vehicle with Disturbance Suppression. 765–771, 2024. https://doi.org/10.1109/smc54092.2024.10831220.

[31] M. Swedan, et. al., Stabilizing of Quadcopter Flight Model, IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA2023), 2023.

[32] M. Bangura and R. Mahony, Real-Time Model Predictive Control for Quadrotors. IFAC Proc., 47, 11773–11780, 2014.

[33] Y. Wang, S. Boyd, Fast Model Predictive Control Using Online Optimization. IEEE Trans. Control. Syst., 18, 267–278, 2010.

[34] M. Greeff, A. Schoellig, Flatness-Based Model Predictive Control for Quadrotor Trajectory Tracking. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, pp. 6740–6745, 2018.

[35] I. Attawil, et. al., Enhancing Lateral Control of Autonomous Vehicles through Adaptive Model Predictive Control, IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), pp. 148-155, 2024.

[36] E. Lefeber, S. Eijnden and H. Nijmeijer, almost global tracking control of a quadrotor UAV, IEEE 56th Annual Conference on Decision and Control (CDC), pp. 1175-1180, Dec 2017.

[37] X. Hu and X. Wang, Nonlinear Model Predictive Control for Quadrotor UAV, IEEE Transactions on Control Systems Technology, vol. 23, no. 2, pp. 706-717, 2015.

[38] L. Xie and R. S. Rezaei, Disturbance Rejection in Model Predictive Control for Quadcopters," IEEE Control Systems Letters, vol. 6, pp. 1389-1394, 2022.

[39] R. Lee and M. Kim, Control Strategies for Quadcopter Dynamics, International Journal of Aerospace Engineering, vol. 10, no. 3, pp. 201-215, 2021.

[40] M. Bangura and R. Mahony, Real-time model predictive control for quadrotors, IFAC Proceedings Volumes, vol. 47, no. 3, pp. 11773-11780, 2014.

[41] M. Aburakhis, et. al., Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems, International Journal of Robotics and Control Systems, vol. 2, no. 1, pp. 37-56, 2022.

[42] I. Buzkhar, et. al., Modeling and Control of a Two-Wheeled Robot Machine with a Handling Mechanism, IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA2023), 2023.

[43] A. Elmilhi, et. al., Sliding Mode Control for the Satellite with the Influence of Time Delay, (MI-STA2023), 2023.

[44] M. Aburakhis, et. al., Performance of Anti-Lock Braking Systems Based on Adaptive and Intelligent Control Methodologies, Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 2022.

[45] X. Wang, H. Grip, A. Saberi and T. A. Johansen, A new low-and-high gain feedback design using MPC for global stabilization of linear systems subject to input saturation, 2012 American Control Conference (ACC), pp. 2337-2342, June 2012.

[46] M. Edardar, et. al., Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems, International Journal of Robotics and Control Systems, vol. 2, no. 1, pp. 37-56, 2022.

[47] M. Xue and I. Hiskens, Alternative strategies for designing stabilizing model predictive controllers, 52nd IEEE Conference on Decision and Control, pp. 4491-4497, Dec 2013.

[48] A. Emhemmed, et. al., Intelligent Control Design for Linear Model of Active Suspension System, 30th International Conference on Microelectronics (IEEE), 2018.

[49] R. Lozano, et. al., Discrete-time stabilization of integrators in cascade: Real-time stabilization of a mini-rotorcraft, Proceedings of the 45th IEEE Conference on Decision and Control, pp. 6265-6270, 2006.

[50] I. Buzkhar, et. al., Design and Implementation of Hydric Controller for Two Wheeled Robot, IJEIT on Engineering and Information Technology, vol.11, no. 1, 2023.

[51] N. Shashoa, et. al., Integrated Controller Design for Underactuated Nonlinear System, IEEE, Second International Conference on Power, Control and Computer Technologies (ICPC2T-2022), 2022.

[52] S. Elwefati, et. al., Identification and control of epidemic disease based neural networks and optimization technique, International Journal of Robotics and Control Systems, vol. 3, issue 4, pp.780-803, 2023.

[53] S. Sawan, et. al., Cancer treatment precision strategies through optimal control theory, Journal of Robotics and Control (JRC), vol. 5, issue 5, pp. 1261-1290, 2024.

[54] M. Bakouri, et. al., Optimizing cancer treatment using optimal control theory, AIMS Mathematics, vol. 9, issue 11, pp. 31740-31769, 2024.

[55] H Maghfiroh, et. al., Cancer treatment precision strategies through optimal control theory, Journal of Robotics and Control (JRC), vol. 5, issue 5, pp. 1261-1290, 2024.