Robust Nonlinear Model Predictive Control for Quadrotor UAV Trajectory Tracking

Yuwen Zhou

Robotics, University of Bristol, Bristol, UK
lu24151@bristol.ac.uk

Abstract. This study adapts a nonlinear dynamic predictive model for quadrotor UAV trajectory tracking; which involves translational and rotational motions, the gravitational contribution and gyroscopic impact. In this implementation, Initially construct the mathematical model of UAV dynamics, then derive the analytical Jacobians of the state function serves as a means for improving the real-time optimization outcomes. A closed reference trajectory is given an initial figure-eight pattern and is followed by gentle vertical oscillations to test the controller's performance. The MPC algorithm is capped with prediction and control horizons to keep the actuator limits in place as well as accelerate maneuverability. Simulation results over a 10 - second period demonstrate effective tracking in the x and z directions. The investigation accentuates the key tasks of weight tuning and constraints formulation that boost airspace quadrotor control's stability and precision and its potential applications in different environments and the UAV field. Extensive simulations confirm the robustness of the proposed controller effectively.

Keywords: model predictive control, unmanned aerial vehicle, trajectory tracking

1 Introduction

In recent years, Unmanned Aerial Vehicle(UAV) technology has developed rapidly and shown great potential in diversified application scenarios. From military reconnaissance to civilian logistics, from agricultural plant protection to emergency rescue, they are well-known for their flexibility, efficiency, and versatility[1]. In particular, multi-rotor UAVs are gradually becoming popular in practical fields such as urban distribution, agricultural management, and military monitoring due to their simple structure, precise control and compact size.

The quadrotor UAV depicted in Figure1 is adapted in the project with flexible control and relatively controllable cost. The characteristics of quadcopter drones include the features of symmetrical structure that could realize the move of forward and backward through the thrust difference of four propellers, flexibility of hovering and maneuvering, along with precision and stability, indicating that it can be deployed indoors or in densely populated areas[2].

J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_83

With the mathematical models for kinematic and dynamic models of drones[3], certain linear and non-linear controllers have been adopted to deal with quadrotor trajectory issues. Proportional-integration-differentiation (PID)[4] as one type of linear controller excels in altitude and heading control but has limitations working out for extended flight envelopes and complex trajectories. LQR offers optimal control solutions with linear dynamics while less effective in facing nonlinear systems and susceptible to uncertainties[5]. To alleviate the limitations in linear control strategy, certain nonlinear control strategies like adaptive slide mode control(SMC)[6], Neural network control, and predictive control[7]. Model Predictive Control (MPC) rose as a crucial technique in UAV control systems by conquering the issues of a large range of operational variables, physical limits, and dynamic changes implemented. In the case of multi-UAV collisions, the MPC could be a way to address the problem of avoidance of conflicting situations and to find the dynamic relationships between the vehicles and the environment. The distributed Model Predictive Control (MPC) optimization method finds collision intersections for multicopter aerial vehicles in complicated 3D segments[8], with experienced obstacles. Studies showcase MPC's capabilities in ensuring system-wide robustness fluctuations in the environment and increasing tracking accuracy, and input and state condition techniques at the same[8], [9], [10]. The work in [11] uses unit quaternions and considers MPC's performance in tracking and optimizing aircraft trajectories.

This study expands on the work discussed in [10]and adapts a Nonlinear Model Predictive Control (NMPC) approach that considers actual constraints on the actuators. This approach directly inputs low-level commands to the rotor-speed controllers within MATLAB Simulink's nonlinear module. Here, the NMPC solver functions as a controller, managing strict real-time constraints effectively. Unlike traditional methods that rely on differential flatness for modeling system dynamics, our approach extends to a broader range of multi-rotor vehicles. This includes platforms with differently-oriented propellers, encompassing both under-actuated and fully-actuated configurations [11]. Therefore, we use MATLAB Simulink's nonlinear module to advance NMPC application in UAV control, accommodating various vehicle designs and enhancing real-time control under dynamic conditions.

Fig. 1. Quadrotor UAV[12]

2 Methodology

2.1 Mathematical model of UAV dynamics

To implement MPC effectively, an accurate mathematical model of UAV dynamics is needed. The system can be modeled using nonlinear differential equations based on the Newton and Euler equation Two frames on the UAV are defined: one is the initial or fixed frame which represents a global reference frame fixed to the earth, in this project, it is defined as $\vec{\boldsymbol{E}}: \{\vec{\boldsymbol{e}}_1, \vec{\boldsymbol{e}}_2, \vec{\boldsymbol{e}}_3\}$, and another body-fixed frame attached to the UAV itself, typically centered at the UAV' s center of mass, describe certain physics quantity directly on the UAV: $\vec{\boldsymbol{B}}: \{\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{b}}_3\}$. With the reference of these two frames, two vectors which represent the translation and rotation could also be defined as $\boldsymbol{Q}$ for the position and rotation in the $\vec{\boldsymbol{E}}$ frame and $\mathbf{V}$ for the translational and rotational velocity in the $\vec{\boldsymbol{B}}$ frame.

$$\boldsymbol{Q} = [\boldsymbol{X} \quad \boldsymbol{\omega}]^{\boldsymbol{T}} = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z} \quad \boldsymbol{\phi} \quad \boldsymbol{\theta} \quad \boldsymbol{\psi}]^{\boldsymbol{T}} \tag{1}$$

$$\boldsymbol{V} = [\boldsymbol{V} \quad \boldsymbol{\Omega}]^{\boldsymbol{T}} = [\boldsymbol{v}_x \quad \boldsymbol{v}_y \quad \boldsymbol{v}_z \quad \boldsymbol{p} \quad \boldsymbol{q} \quad \boldsymbol{r}]^{\boldsymbol{T}} \tag{2}$$

Where $\underline{\boldsymbol{X}} = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z}]^{\boldsymbol{T}}$ is the relative position of the center of mass of the UAV and $\boldsymbol{\omega} = [\boldsymbol{\phi} \quad \boldsymbol{\theta} \quad \boldsymbol{\psi}]^{\boldsymbol{T}} = [\boldsymbol{Roll} \quad \boldsymbol{Pitch} \quad \boldsymbol{Yaw}]^{\boldsymbol{T}}$ which refers to the orientation of the quadrotor in the $\vec{\boldsymbol{E}}$ frame. The two relations with the physics quantity in two frames could be denoted as:

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{z}} \end{bmatrix} = \boldsymbol{R} \cdot \boldsymbol{V} = \boldsymbol{R} \cdot \begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_y \\ \boldsymbol{v}_z \end{bmatrix} \tag{3}$$

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\psi}} \end{bmatrix} \tag{4}$$

The $\mathbf{R}$ is the rotation matrix used to divert the rotation matrix from the body reference to the inertial reference, and its principle is to calculate the thrust imbalance between the forces: The rotation matrix on the $\boldsymbol{Roll}, \boldsymbol{Pitch}, \boldsymbol{Yaw}$ motion could be written as:

$$\boldsymbol{R_\phi} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{c\phi} & \boldsymbol{s\phi} \\ \mathbf{0} & -\boldsymbol{s\phi} & \boldsymbol{c\phi} \end{bmatrix} \quad (5)$$

$$\boldsymbol{R_\theta} = \begin{bmatrix} \boldsymbol{c\theta} & \mathbf{0} & \boldsymbol{s\theta} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\boldsymbol{s\theta} & \mathbf{0} & \boldsymbol{c\theta} \end{bmatrix} \quad (6)$$

$$\boldsymbol{R_\psi} = \begin{bmatrix} \boldsymbol{c\psi} & -\boldsymbol{s\psi} & \mathbf{0} \\ \boldsymbol{s\psi} & \boldsymbol{c\psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (7)$$

The rotation matrix could be calculated through the multiplication of the matrices in three motions：

$$\boldsymbol{R} = \boldsymbol{R_\psi R_\theta R_\phi} \quad (8)$$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{c\theta c\psi} & \boldsymbol{s\phi s\theta c\psi} - \boldsymbol{c\phi s\psi} & \boldsymbol{c\phi s\theta c\psi} + \boldsymbol{s\phi s\psi} \\ \boldsymbol{c\theta s\psi} & \boldsymbol{s\phi s\theta s\psi} + \boldsymbol{c\phi c\psi} & \boldsymbol{c\phi s\theta s\psi} - \boldsymbol{s\phi c\psi} \\ -\boldsymbol{s\theta} & \boldsymbol{s\phi c\theta} & \boldsymbol{c\phi c\theta} \end{bmatrix} \quad (9)$$

$\boldsymbol{T}$ could be denoted from the relationship of the equation as:

$$\boldsymbol{\Omega} = \left(\boldsymbol{R_\phi^T R_\theta^T} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\boldsymbol{\psi}} \end{bmatrix}^\wedge \boldsymbol{R_\theta R_\phi} + \boldsymbol{R_\phi^T} \begin{bmatrix} \mathbf{0} \\ \dot{\boldsymbol{\theta}} \\ \mathbf{0} \end{bmatrix}^\wedge \boldsymbol{R_\phi} + \begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}^\wedge \right)^\vee \quad (10)$$

Simplify this equation and get the relationship between the rotational velocity in the $\vec{\boldsymbol{B}}$ frame and $\vec{\boldsymbol{E}}$ frame:

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & -\boldsymbol{s\theta} \\ \mathbf{0} & \boldsymbol{c\phi} & \boldsymbol{s\phi c\theta} \\ \mathbf{0} & -\boldsymbol{s\phi} & \boldsymbol{c\phi c\theta} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\psi}} \end{bmatrix} \quad (11)$$

Thus we could get the $\boldsymbol{T}$ matrix:

$$\boldsymbol{T} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & -\boldsymbol{s\theta} \\ \mathbf{0} & \boldsymbol{c\phi} & \boldsymbol{s\phi c\theta} \\ \mathbf{0} & -\boldsymbol{s\phi} & \boldsymbol{c\phi c\theta} \end{bmatrix} \quad (12)$$

To build a dynamic model of a quadrotor UAV, several external forces, thrusts, and inertial moments should be considered: First is the external forces applied on UAV:

1) Gravity: Force related to the mass of the UAV.

2) Thrust from propellers: Generated by the rotation of the four rotors, be the vector, the directions are perpendicular to the plane of rotation the specific values are proportional to the square of the rotor’ s rotational velocity .

$$\boldsymbol{F_t} = \boldsymbol{b}(\boldsymbol{\omega}_1^2 + \boldsymbol{\omega}_2^2 + \boldsymbol{\omega}_3^2 + \boldsymbol{\omega}_4^2) \qquad (13)$$

Where $\boldsymbol{b}$ is the thrust coefficient, **and** $\boldsymbol{\omega_i}$ is the rotational speed of the **i**-th rotor.
3) Air Drag: Consisting mainly of two components:

Drag from the interaction between rotating propeller:

$$\boldsymbol{F_{d,p}} = \boldsymbol{k_p} \sum_{i=1}^{4} \boldsymbol{\omega}_i^2 \qquad (14)$$

Drag on the UAV body along the (x, y, z):

$$\boldsymbol{F_{d,b}} = -\boldsymbol{C_d}\boldsymbol{v} \qquad (15)$$

where $\boldsymbol{C_d}$ is the drag coefficient, and $\boldsymbol{v}$ is the velocity vector.
About the Inertial Moments on the drone dynamic System:
1) Inertial Moments due to Thrust Differences: Because of the asymmetric thrust distribution which affects the UAV’ s x and y axes between the rotors, the separated motor’ s thrust could be calculated from the difference between rotors 1 and 3, and rotors 2 and 4.

$$\boldsymbol{M_x} = \boldsymbol{lb}(\boldsymbol{\omega}_2^2 - \boldsymbol{\omega}_4^2) \qquad (16)$$

$$\boldsymbol{M_y} = \boldsymbol{lb}(\boldsymbol{\omega}_3^2 - \boldsymbol{\omega}_1^2) \qquad (17)$$

$$\boldsymbol{M_z} = \boldsymbol{k}(\boldsymbol{\omega}_1^2 - \boldsymbol{\omega}_2^2 + \boldsymbol{\omega}_3^2 - \boldsymbol{\omega}_4^2) \qquad (18)$$

Where $\boldsymbol{l}$ is the arm length from the center of mass to the rotor, and $\boldsymbol{k}$ is the yaw torque coefficient.
2) Inertial Moments due to Air Drag: Caused by the rotation of propellers. The aerodynamic drag moment on the UAV body is:

$$\boldsymbol{M_d} = -\boldsymbol{C_m}\boldsymbol{\Omega} \qquad (19)$$

where $\boldsymbol{C_m}$ is the moment drag coefficient and $\boldsymbol{\Omega}$ is the UAV angular velocity.
3) Inertial Moments due to Aerodynamic Effects: generated from aerodynamic forces experienced during UAV movement.
4**)** Gyroscopic Effect: High-speed rotation of rotors makes rapid changes in rotor plane orientation difficult. The gyroscopic moment generated by the four rotors is expressed as:

$$\boldsymbol{M}_{gH} = \sum_{i=1}^{4} \boldsymbol{\Omega} \boldsymbol{J}_r \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ (-\mathbf{1})^{i+1} \boldsymbol{\omega}_i \end{bmatrix} \tag{20}$$

where $\boldsymbol{J}_r$ is the rotor inertia and $\boldsymbol{\omega}_i$ is the rotational speed of each rotor.
Additionally, the gyroscopic moment caused by UAV movement itself is:

$$\boldsymbol{M}_{gm} = \boldsymbol{\Omega} \wedge \boldsymbol{J}\boldsymbol{\Omega} \tag{21}$$

where **J** represents the UAV's inertia matrix.

With the mathematical model of external forces and Inertial Moments of the dynamic system of drones, generate the Translation Dynamics Equation:

$$\boldsymbol{m}\dot{\boldsymbol{v}} = \sum \boldsymbol{F} \tag{22}$$

where $\boldsymbol{m}$ is the UAV mass, $\boldsymbol{v}$ is the linear velocity in the inertial frame, and **F** is the total external force on the UAV.
Rotational Dynamics Equation: According to Euler's equation, the rotational dynamics are:

$$\boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \wedge \boldsymbol{J}\boldsymbol{\Omega} = \sum \boldsymbol{M} \tag{23}$$

where $\boldsymbol{\Omega}$ is the UAV's angular velocity, and $\boldsymbol{M}$ is the total external moments acting on the UAV. The moments could vary with certain added forces and disturbances in terms of wind and fixed obstacles.
Then generate the control variable matrix. The input parameters are the U_1, U_2, U_3 *and* U_4. :

$$\begin{bmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \\ \boldsymbol{U}_3 \\ \boldsymbol{U}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{M}_x \\ \boldsymbol{M}_y \\ \boldsymbol{M}_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} & \boldsymbol{b} & \boldsymbol{b} & \boldsymbol{b} \\ \mathbf{0} & -\boldsymbol{b} & \mathbf{0} & \boldsymbol{b} \\ -\boldsymbol{b} & \mathbf{0} & \boldsymbol{b} & \mathbf{0} \\ \boldsymbol{d} & -\boldsymbol{d} & \boldsymbol{d} & -\boldsymbol{d} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_1^2 \\ \boldsymbol{\omega}_2^2 \\ \boldsymbol{\omega}_3^2 \\ \boldsymbol{\omega}_4^2 \end{bmatrix} \tag{24}$$

2.2 MPC implementation

With the given model in 24, the MPC could be implemented either with following two approaches. Linear MPC Approach: The approach starts by determining an operating point where a linear approximation of the nonlinear system is made, which results in a system where the effects that are difficult to model[13], such as linear drag and gyroscopic effects (inertia moments), can be treated as the external disturbances of simplified models. After leveraging this linear model, it is easy to formulate a convex receding horizon optimization problem, which is computationally efficient and feasible to use. Nevertheless, this simplification introduces inaccuracy to the interval of point linearization, which may become a drawback if the usage of airplane design requires a big variation of deviation. Nonlinear MPC Approach[14]: Whereas MPC addresses the nonlinear dynamic system in whole, making use of its entire nonlinear dynamics model, which automatically entails all system constraints and complexities. This approach solves a generally non-convex optimization problem over a receding horizon, enabling the controller to handle a broader range of operating conditions and providing enhanced performance and robustness. Still maintaining a high complexity for the nonlinear algorithm, I use its huge potential of balancing the nonlinear behavior of the process.

The first step is to define the state and control vectors. Define the state vector $\mathbf{x} \in \mathbb{R}^{\mathbf{12}}$ as

$$\boldsymbol{x} = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z} \quad \dot{\boldsymbol{x}} \quad \dot{\boldsymbol{y}} \quad \dot{\boldsymbol{z}} \quad \boldsymbol{\phi} \quad \boldsymbol{\theta} \quad \boldsymbol{\psi} \quad \boldsymbol{p} \quad \boldsymbol{q} \quad \boldsymbol{r}]^{\boldsymbol{T}} \qquad (25)$$

As stated in the model construction part; $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ are the positions in the inertial frame, $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\boldsymbol{z}})$ are the corresponding linear velocities, $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$ are the Euler angles (roll, pitch, yaw), and $(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$ are the body angular velocities.

Then define the control input, which could be specified as:

$$u = [\omega_1^2 \quad \omega_2^2 \quad \omega_3^2 \quad \omega_4^2]^T \qquad (26)$$

It is produced as the produce the total thrust and moments with the calculation in formula 24.

The main concept underlying MPC is the use of system state prediction, which gives the method's capability of anticipating process dynamics, thus being the best control strategy while managing processes that are dynamic and interactive. While all old control techniques are conventional, then the biggest strength of MPC is reacting to the present and past states. Furthermore, the most crucial component of MPC is the equation that reflects the entire system it is trying to control. This dynamic model is applied to estimate the system behavior and to calculate the control quantity based on optimizing the predicted results, which is equal to the control quantity at the present moment [15].

The second basic concept in MPC is to choose the likely initial state of the system by exploiting the past measurements. Thus, essentially the model of the plant is the tool that is central to any of the control strategies based on MPC. The MPC mainly includes three steps: Model prediction, Evolving position results, and dynamic state correction [16].

In the first model prediction step in Figure2: Writing a model prediction often involves predicting the future output of the system based on the historic information of the object and the next output. Let us consider the first example of the output of the system at time K. Suggest that the system has two possible line input trajectories u_1 and u_2 at time K, the output trajectories, $y_3$and y_4, respectively, of the input pairs u_1 and u_2 are devised. During optimization, every possible input trajectory of the system will go through the space to generate the corresponding output, and so-called type-prediction can be used for specifying the comparison conditions. Finally, the most appropriate control input will be selected, which in turn will lead to better control of the system considered.

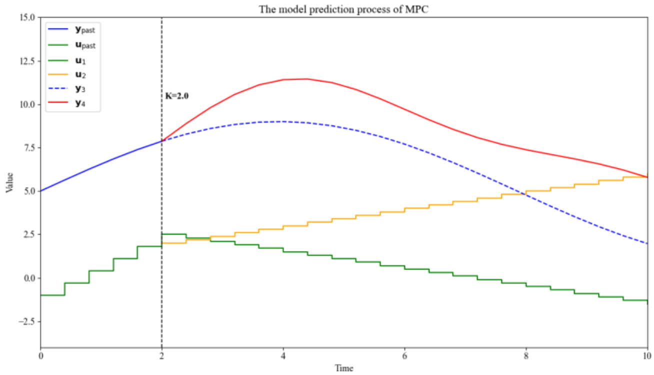

Fig. 2. The model prediction process of MPC

1. Model prediction should obtain a discrete state space equation depicted below

$$x_{k+1} = Ax_k + Bu_k \quad (27) \qquad y_k = Cx_k + Du_k \quad (28)$$

A is the state transition matrix. B is the control input matrix. C is the output matrix, D is the direct transmission matrix.

The next stage utilizes receding horizon optimization, and at each sampling time, the system is re-optimized to get a new optimal trajectory of input. Only the first input is applied to the controlled object at that sampling time. The principle is that we will try to maximize the output using the status variables at time *k, k+1, …, k+* N_c (N_c is the control horizon), where k is the time index. As discussed earlier within discrete space equation: When we choose the control variable in the moment $k+i$, so it will be the output of moment *k+i+1*. Episodically, at each sampling time, we have the receding horizon optimization. The Figure3 can show that at time k: it commands N_c steps and predicts the output, after choosing u_k as the input of the control variable. We realize that our control system gain at time *k+1* has arrived at a predicted output. Meanwhile, we still use the first control variable as input. Furthermore, the system iteratively calculates a new control trajectory every sampling time and then apply only the first input to the controlled object.

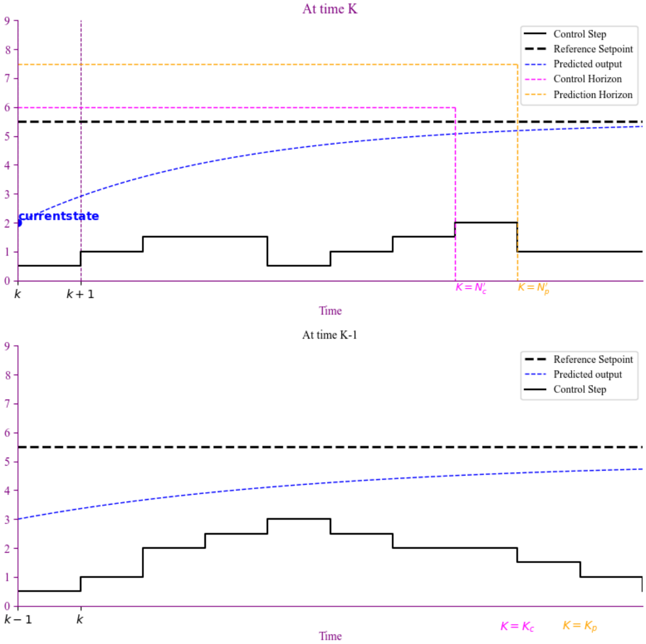

Fig. 3. The receding horizon optimization step

The cost function, which is crucial for determining the optimal control actions of the MPC controller and maintaining system states within prescribed bounds while simultaneously regulating the expenditure of control energy it has an equation in 29:

$$\ell(x,u) = \| y(x) - y_{\text{ref}} \|_Q^2 + \| \Delta u \|_R^2 \quad (29)$$

$\ell(x,u)$ is the total cost or objective function that MPC aims to minimize. represents the state of the system at time k, Δu represents the control, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the penalty matrices. $\boldsymbol{Q}$ positive semi-definite weight matrix that penalizes tracking errors and $\boldsymbol{R}$ is positive definite weight matrix that penalizes the rate of change of the control input. $\| \Delta u \|_R^2$ is the control effort.

During each control cycle the solve the optimization issue in 30(this formula is to minimize a sum of stage costs and a terminal cost) subject to the system dynamics (which are nonlinear) and any constraints on states and control inputs.

$$\min_{\mathbf{u}} J(x(k), \mathbf{u}) \quad (30)$$

A cost function's purpose is to gain a balance of the conflict in the delivery of those two matrices in terms of maximum state fidelity and minimum control energy, whereas this leads to a highlight and recognition of the complex interrelationship between theory and practice of control systems. Through controllers, it is possible to make the intelligent choices that forecast the system performance in the future and avoid the one-hit wonder control strategies. Otherwise, they might cause the decline in the efficiency of a technology.

The last part of an MPC is the feedback of the result-correcting. This is a feedback control process whose goal is to prevent deviations from the predicted model and correct for the uncovered disturbances. This is illustrated in Figure4. After identifying the optimal input trajectory at the moment k, output trajectory generation can be represented in green trajectory graphically. Then, it follows the first one of the two optimal input values, *u(k)* has been applied on the controlled object. First, we can see that there is a predicted value (a system state since the last action *u(k)* taken at time k) for the time (t) k+1, that is point 2, and there is also an actual state (the result of the action *u(k)*) for that time k+1, that is point 3. To correct the prediction error in space of k+1 and *y(k+1)* prediction trajectory1, they are down-shifted to synthesize a new correction prediction trajectory, which is shown by red curve 4. Consequently, the modeling errors can be eliminated, and the prediction error can be contained and avoided from its accumulation.

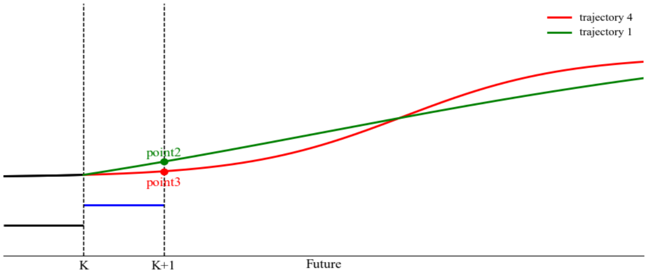

Fig. 4. Feedback correction

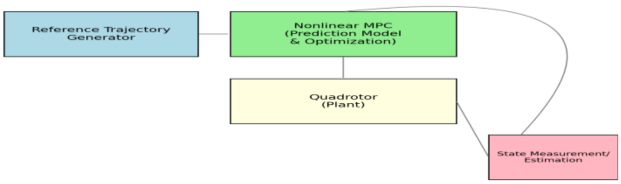

Fig. 5. Figure5. Control logic diagram

Certain descriptions of flowchart in Figure5: Reference Trajectory Generator serves to create the required reference signal (trajectory) to meet the assignment requirements, Optimization Solver is formulated at each sampling time with an optimal control problem, which is then Receding Horizon Strategy: The first action from the control sequence obtained and applied; Actuator and Plant is that the way the quadrotor is influenced by the MPC is through the control commands as the latter are transmitted to the quadrotor (the plant). The actuators therefore transform these control signals into forces and moments in physical form and thereby result in the quadrotor's movement.

3 Simulation and results

In Table 1, the main information is given in terms of MPC settings. The constraints section is where you put controls (in N • m) input range that the actuators should work with to be sure that they are within the safety limits and to avoid saturation. This save stability and make a good general control. Article 1 in the weights section declares the weight of translational motions along the x, y, and z axes as 1.0, 1.3, and 1.0 respectively, while the roll, pitch, and yaw motions are given unity weight as 1.0. Such a preference is implemented in a manner that the system will be able to track the position with smaller errors, at the same time, smoothly controlling the Euler angles of the object, as a consequence.

Table 1. The required parameters of MPC

MPC Parameters	
Sample time (T_s)	0.1 seconds
Prediction Horizon (P)	10 seconds
Control Horizon (M)	3 seconds
Limits for four control inputs	
Limits of U_1	$[0, 10]$ N.m
Limits of U_2	$[-2, 2]$ N.m
Limits of U_3	$[-2, 2]$ N.m
Limits of U_4	$[-2, 2]$ N.m
Weights for Output variables	
Translation motion in x	1.0
Translation motion in y	1.3
Translation motion in z	1.0
Roll motion	1.0
Pitch motion	1.0
Yaw motion	1.0

Figure7 shows the trajectory function adapted to observe the performance in this nonlinear MPC project, which has the function below:

$$\begin{cases} \boldsymbol{x(t) = 4sin(t)} \\ \boldsymbol{y(t) = 4sin(t)cos(t)} \\ \boldsymbol{z(t) = 4 + 0.8cos(2t)} \end{cases} \tag{31}$$

Figure6 depicts one status of UAV during the simulation; The white is the reference UAV and the dark one is the controlled entity. Figure8 show the reference and performance among three axes,This dynamic equation manifests itself in terms of frequency, with a pronounced shift of the signal at y; because of this, more rapid amplitude changes occur, rendering the system more sensitive to noise and non-linearities. Overall, this MPC controller application maintained a reference path and provided proper control inputs and optimal tracking control to achieve the goal of the research efficiently.

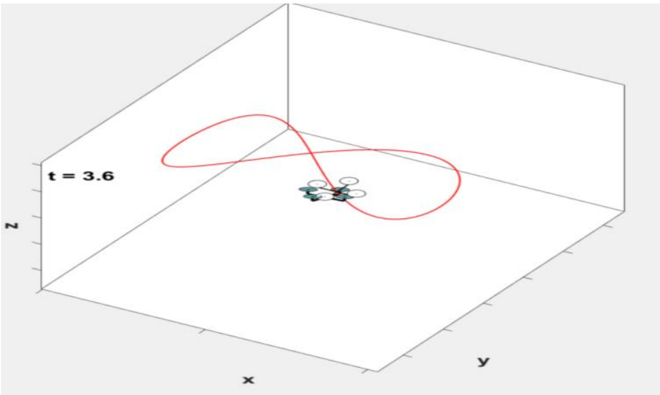

Fig. 6. The trajectory status while t=3.6s

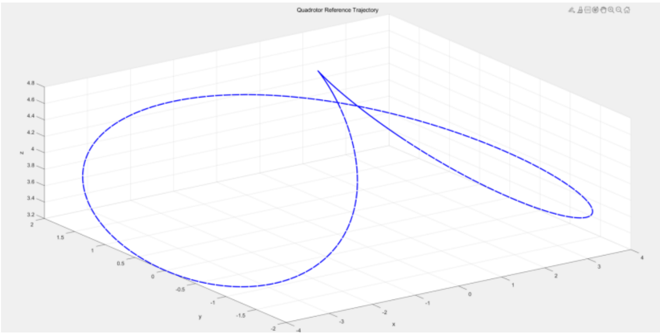

Fig. 7. The reference trajectory

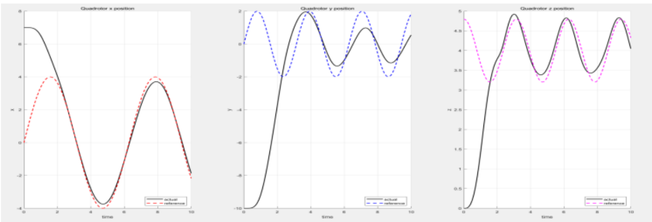

Fig. 8. The Tracking trajectory behavior in xyz axes.

4 Conclusion

The study discusses a highly efficient nonlinear Model Predictive Control (MPC) in the case of quadrotor UAV movement, being the approach that fully deserves to be adopted. The proposed approach exploits the dynamical model that is described by Newton's and Euler's equations, which take into account both translational and rotational aspects. The dynamic model that consists of those physical influencing factors such as thrust, gravity, air drag, and gyro effect forms a theoretical basis of nonlinear MPC framework. As far as the analytical Jacobian of the state function is concerned, it essentially enables the controller to process information more quickly, resulting in a situation when it can apply to real-time constraints.

The controller performance simulations further testify to the system configured developed. The computer-simulated trajectory, which operates with closed-loop control, leads the UAV, in particular, to trace a complicated closed form with a designated function of reference over a 10-second interval. The findings emphasize that while the control approaches the x and z trajectories with lower errors, the error in the y trajectory is larger in magnitude, yielding a higher error.

The overall structure and simulation result platform pointed out the capabilities of nonlinear model predictive control, especially in tackling the UAV control problems with fine trajectory chasing and disturbance rejection. Further work involves applying the adaptive control strategy to analyze and optimize the disturbances and nonlinearities.

References

1. Gao, Y., Deng, X., and Fang, W.: 'Dynamic modeling and flight performance analysis of quadrotor UAV under wind field disturbances', in 2023 IEEE International Conference on Unmanned Systems (ICUS), 2023, pp. 240 – 245, DOI: 10.1109/ICUS58632.2023.10318328.
2. Chovancová, A., Fico, T., Chovanec, E., and Hubinský, P.: 'Mathematical Modelling and Parameter Identification of Quadrotor (a survey)', Procedia Engineering, vol. 96, Jan. 2014 pp. 172 – 181,
3. Kamel, B., Yasmina, B., Laredj, B., Benaoumeur, I., and Zoubir, A.-F.: 'Dynamic modeling, simulation, and PID controller of unmanned aerial vehicle UAV', in 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Luton, UK, 2017, pp. 64 – 69, DOI: 10.1109/INTECH.2017.8102445.
4. Li, J., and Li, Y.: 'Dynamic analysis and PID control for a quadrotor', in 2011 IEEE International Conference on Mechatronics and Automation, 2011, pp. 573 – 578.
5. Chen, C.-C., and Chen, Y.-T.: 'Feedback linearized optimal control design for quadrotor with multi-performances', IEEE Access, vol. 9, pp. 26 674 – 26 695, 2021.
6. Mofid, O., Mobayen, S., Zhang, C., and Esakki, B.: 'Desired tracking of delayed quadrotor UAV under model uncertainty and wind disturbance using adaptive super-twisting terminal sliding mode control', ISA Transactions [online], vol. 123, 2022, pp. 455 – 471.
7. Wang, D., Pan, Q., Shi, Y., Hu, J., and Zhao, C.: 'Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment', IEEE Transactions on Cybernetics, vol. 51, no. 10, pp. 5057 – 5068, 2021.
8. Boivin, E., Desbiens, A., and Gagnon, E.: 'UAV collision avoidance using cooperative prediction control', 16th Medit. Conf. Cont. Autom. Cong., Center, Ajaccio, France, 2008, pp. 682 – 688.
9. Gagnon, E., Rabbath, C. A., and Lauzon, M.: 'Heading and position receding horizon control for trajectory generation', Amer. Cont. Conf., Portland, Oregon, 2005, pp. 134 – 139.
10. Gagnon, E., Rabbath, C. A., and Lauzon, M.: 'Receding horizons with heading constraints for collision avoidance in unknown environments', AIAA Guid., Nav. Cont. Conf., San Francisco, California, 15 – 18 August 2005, AIAA 2005-6369, pp. 1 – 15.

11. Islam, M., Okasha, M., and Sulaeman, E.: 'A model predictive control (MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking', International Journal of Control, Automation and Systems, vol. 17, no. 11, pp. 2819 – 2832, 2019.
12. 'Quadrotor UAV' [online image], (n.d.), retrieved 22 March 2025, from https://uk.images.search.yahoo.com/yhs/search;_ylt=AwrIfVuTz95nRnYRTCl3Bwx.;_ylu=Y29sbwMEcG9zAzIEdnRpZAMEc2VjA3Nj?p=quadrotor+UAV+WHITE+BACKGROUND&type.
13. Ali, A. M., Hashim, H. A., and Shen, C.: 'MPC Based Linear Equivalence with Control Barrier Functions for VTOL-UAVs', in 2024 American Control Conference (ACC), Toronto, ON, Canada, 2024, pp. 1 – 6, DOI: 10.23919/ACC60939.2024.10644310.
14. Lee, J. D., Kang, M., Han, S., and Bang, H.: 'Terrain Following of Tiltrotor UAV using MPC and Nonlinear Disturbance Observer-Based Backstepping', in 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, Republic of, 2022, pp. 852 – 857.
15. Mehrotra, S.: 'On the implementation of a primal-dual interior point method', SIAM Journal on Optimization, 1992, 2(4), pp. 575 – 601.
16. Kong, X., and Liu, X.: 'Efficient nonlinear MPC on permanent magnet synchronous motor', in 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 2017, pp. 4644 – 4647, DOI: 10.1109/CCDC.2017.7979317.