Trajectory Tracking by Quadcopter using MPC in Presence of Obstacles and External Disturbances

Abhinav Kumar
IIT Jodhpur
Jodhpur, India
kumar.288@iitj.ac.in

Shreyash Gupta
IIT Jodhpur
Jodhpur, India
gupta.50@iitj.ac.in

Somil Maheshwari
IIT Jodhpur
Jodhpur, India
maheshwari.7@iitj.ac.in

S. V. Shah
IIT Jodhpur
Jodhpur, India
surilshah@iitj.ac.in

Niladri Sekhar Tripathy
IIT Jodhpur
Jodhpur, India
niladri@iitj.ac.in

ABSTRACT

The capability of smooth trajectory tracking is essential for Unmanned Aerial Vehicles (UAVs) to perform several complex tasks. This paper proposes a collision-free framework for smooth trajectory tracking of rotary wing UAVs in the presence of any external disturbances in motion of the vehicle. For this, a simplified kinematic model is used which closely resembles the behaviour of rotary wing UAVs. A Model Predictive Control (MPC) framework together with on-demand collision avoidance constraints is proposed which penalizes rapid change in subsequent control inputs, thereby enabling trajectory tracking with minimum jerk, while also avoiding collision with any obstacle in the environment. Moreover, an Asynchronous Path Smoothing (APS) strategy is concatenated with the MPC framework to deal with the external disturbances in motion of the vehicle. Numerical simulations are presented to validate the efficacy of the proposed framework.

CCS CONCEPTS

• **Computer systems organization → Robotic control**.

KEYWORDS

Trajectory-tracking, MPC, quadcopter, obstacle-avoidance, APS

ACM Reference Format:

Abhinav Kumar, Shreyash Gupta, Somil Maheshwari, S. V. Shah, and Niladri Sekhar Tripathy. 2023. Trajectory Tracking by Quadcopter using MPC in Presence of Obstacles and External Disturbances. In *Advances In Robotics - 6th International Conference of The Robotics Society (AIR 2023), July 05–08, 2023, Ropar, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3610419.3610500

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have the ability to access areas that may be dangerous for humans or difficult for mobile robots to reach. They can be controlled remotely or autonomously and come in different shapes and sizes for various applications like search and rescue, aerial photography, scientific research, and military operations. They can be equipped with payloads and sensors like cameras, communication devices, and radar. They are also used for indoor warehouse or industry inspections. However, the biggest challenge in this type of task is avoiding obstacles in the limited space. Traditionally, UAV inspections are carried out manually with remote control. This process is challenging, time-consuming, and requires skilled or certified operators. The risk of human error is inevitable, which may lead to crashes and infrastructural damage. To overcome this issue, autonomous control strategies have been developed to ensure the drone follows a predetermined path or trajectory strictly. However, this approach creates new problems, such as jerky movements and inefficiency in energy consumption. The need is for a control strategy that can follow the trajectory effectively while minimizing energy consumption through minimal control inputs and also reducing the jerks in the trajectories. This study aims to address the same challenge.

There are many studies available in the literature which aim to track a trajectory by a UAV using different control strategies. For example, PID in [1], PD in [10], Feedback Linearisation Technique combined with PD in [2] all of them are using complex mathematical models of the quadcopter to develop the control algorithm. Some studies have also compared the effectiveness of different control strategies like in [9] the authors have performed the experimental comparison of PID, LQR and MPC over a Parrot Mambo Mini-Drone and have found out that LQR and MPC have performed far better than PID however LQR and MPC have no significant difference between them in practical results. Some work uses control strategies based on Predictive Control for trajectory tracking, in [6], [5], [12] the Model Predictive Control (MPC) is used, in [8] the Generalised Predictive Control is used which have some differences from traditional MPC, in [12] the data-driven MPC is used for trajectory tracking where the aerodynamic disturbances are first learned through the data and then modeled in the system, in [11], the authors have also done the same by adding the framework which that can reduce the learning time of disturbances, all are using complex mathematical models of the quadcopter for trajectory tracking and proved to be very effective. The predictive control strategies have clear advantages over all the previous control strategies of previewing the future states of the system and optimizing the control inputs

AIR 2023, July 05–08, 2023, Ropar, India

ACM ISBN 978-1-4503-9980-7/23/07…$15.00
https://doi.org/10.1145/3610419.3610500

accordingly but at the cost of computational power and processing time. In [8] and [9], authors either have to use a better computing unit or increase the sampling time to perform the experiments successfully. If the processing time needed for running a step is higher, then the sampling time is needed to be increased but that will restrict the control we have over the system. To counter this, systems using MPC run the calculations off-board and transmit the control inputs to the UAV, this creates new problems of communication delay or lag, which may be time-varying. Several literature used a mix of two control strategies like in [3], the MPC and PID are used simultaneously for trajectory tracking, where PID is used for altitude control. Most of the studies majorly focus on trajectory tracking strictly; few studies talk about obstacle avoidance along with trajectory tracking, in [7], the authors have successfully generated the collision-free trajectories for multiagent systems. In [4], the authors have successfully segregated multiple mobile robots into multiple groups by avoiding each other. The authors have also commented about the jerks in the trajectory.

In this study, we build upon [4], for trajectory tracking of Quadcopter in the presence of obstacles and external disturbances. The primary differences between our work and [4] are as follows:

(1) Use of a simplified linear discrete-time kinematic model which closely resembles the behavior of any rotary wing UAV.
(2) The cost function is modified by introducing a term which now penalizes rapid change in subsequent control inputs which ensures jerk minimization.
(3) The APS strategy is modified to update position as well as velocity when the error between desired and measured positions exceeds a specified limit, ensuring path smoothing.

2 PROBLEM STATEMENT & PROPOSED SOLUTION

In this section, the discrete-time linear kinematic model of the system used in this study is described, then the problem is discussed followed by its proposed solution.

2.1 System Model

The Quadcopter is assumed to be following the given discrete-time linear kinematic model,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] \tag{1}$$

where the states $\boldsymbol{x} \in \mathbb{R}^6$ are the position and velocity of the Quadcopter, the control input $\boldsymbol{u} \in \mathbb{R}^3$ is the acceleration of the Quadcopter, and the system matrices $\boldsymbol{A} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{B} \in \mathbb{R}^{6\times3}$ with sampling time h are given as

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & h & 0 & 0 \\ 0 & 1 & 0 & 0 & h & 0 \\ 0 & 0 & 1 & 0 & 0 & h \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{B} = \begin{bmatrix} 0.5h^2 & 0 & 0 \\ 0 & 0.5h^2 & 0 \\ 0 & 0 & 0.5h^2 \\ h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & h \end{bmatrix}. \tag{2}$$

The output equation can be represented as

$$\boldsymbol{y}[k+1] = \boldsymbol{C}\boldsymbol{x}[k+1] \tag{3}$$

where $\boldsymbol{y} \in \mathbb{R}^3$ is the position of the Quadcopter and $\boldsymbol{C} \in \mathbb{R}^{3\times6}$ is a matrix mapping states to output, given as

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

2.2 Problem Statement

The conventional control methods for rotary wing-type UAVs result in harsh control inputs that cause them to move erratically and consume a lot of power. The goal is to create a control strategy for rotary wing type UAVs for example quadcopters to carry out autonomous operations smoothly and effectively while minimizing power consumption and avoiding obstacles.

2.3 Overview of the Proposed Solution

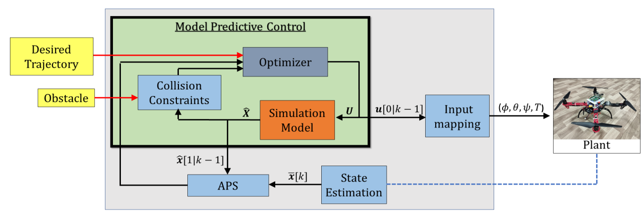


Figure 1: Schematic diagram of the proposed solution.

One way to approach the problem of generating aggressive control inputs in autonomous control strategies for UAVs is to use Model Predictive Control (MPC) with a simplified model and Asynchronous Path Smoothing (APS) technique. By using a model predictive control strategy, control inputs can be generated to effectively follow the desired trajectory. Many works in the literature use MPC with complex dynamic mathematical models of the quadcopter, In this work, we are using a simple kinematic mathematical model of a point object in 3D having accelerations in x, y and z as the control input. By using this simple model we can reduce the computational load and also the processing time over the onboard processor which can save battery power and leads to a longer time of flight. The reduced processing time will provide us the freedom to choose a shorter sampling time in MPC which can give us more control over UAV and the agility of the UAV can be increased. As shown in Fig. 1, we have desired trajectory and the position of the obstacle as our input. MPC is running an optimizer which is taking the desired trajectory collision constraints and current state of the system as its input and generating the sequence of control actions upto the prediction horizon of the MPC out of which the first control input is being fed to the plant through a mapping and all the control sequence is fed to the simulation model to obtain the future states of the system as a result of the series of control input provided. From these future states of the system the position of the system is extracted and the distance from the obstacle is measured, if the distance is found to be less than the safe limit then the collision constraints are generated and fed to the optimiser, all of this happen in the collision constraint block. One of the input of the optimiser is the measured state of the system which the APS block is deciding weather the first predicted state as is is given as the input or the actual measured state is given as the input. The first control input of MPC is mapped to the actual system by an input mapping that can vary among different types of rotary-wing UAVs. This mapping can be calculated for any rotary-wing aircraft or can be found experimentally. APS will further help in reducing the jerks in the trajectory and improving the energy efficiency.

3 TRAJECTORY TRACKING USING MPC WITH APS

3.1 The Prediction Model

The prediction model of the Quadcopter can be represented as

$$\hat{\boldsymbol{x}}[k_t+1|k] = \boldsymbol{A}\hat{\boldsymbol{x}}[k_t|k] + \boldsymbol{B}\hat{\boldsymbol{u}}[k_t|k] \tag{5}$$

$$\hat{\boldsymbol{y}}[k_t+1|k] = \boldsymbol{C}\hat{\boldsymbol{x}}[k_t+1|k] \tag{6}$$

where $\hat{(.)}[k_t|k]$ represents the predicted value of $(.)[(k_t+k)h]$ with the information available at time-step kh, $k_t \in \{0, 1, \cdot, N-1\}$, N is the length of prediction horizon and h is sampling time. The stacked predicted state equation for one prediction horizon can be represented as,

$$\hat{\boldsymbol{X}} = \boldsymbol{A'}\bar{\boldsymbol{x}}[0|k] + \boldsymbol{B'}\hat{\boldsymbol{U}} \tag{7}$$

where $\hat{\boldsymbol{X}} \in \mathbb{R}^{6N}$ and $\hat{\boldsymbol{U}} \in \mathbb{R}^{3N}$ are the stacked predicted states and inputs respectively, and $\bar{\boldsymbol{x}}[0|k]$ is the current measured state of the Quadcopter. The matrices $\boldsymbol{A'} \in \mathbb{R}^{6N\times 6}$ and $\boldsymbol{B'} \in \mathbb{R}^{6N\times 3N}$ are given as

$$\boldsymbol{A'} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ . \\ . \\ . \\ \boldsymbol{A}^N \end{bmatrix}, \boldsymbol{B'} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \cdot & \boldsymbol{0} \\ \boldsymbol{AB} & \boldsymbol{B} & \cdot & \boldsymbol{0} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \cdot & \boldsymbol{B} \end{bmatrix}. \tag{8}$$

The stacked predicted output equation for one prediction horizon can similarly be represented as,

$$\hat{\boldsymbol{Y}} = \boldsymbol{C'}\hat{\boldsymbol{X}} \tag{9}$$

where $\hat{\boldsymbol{Y}} \in \mathbb{R}^{3N}$ is the stacked predicted output and the output matrix $\boldsymbol{C'} \in \mathbb{R}^{3N\times 6}$ is given as,

$$\boldsymbol{C'} = [\boldsymbol{C}\ \boldsymbol{C}\ \cdots\ \boldsymbol{C}]^T \tag{10}$$

3.2 Cost Function

To ensure that the Quadcopter tracks the desired trajectory with a minimum jerk, we define a cost function with three terms, the first term ensures trajectory tracking by minimizing the error between the predicted positions of the Quadcopter and the desired trajectory, the second term ensures minimum input generation and the third term reduces the jerk by reducing the change in control input. Let, $\boldsymbol{Y_d} \in \mathbb{R}^{3N}$ be a vector of the desired trajectory, the cost function for trajectory tracking with a minimum jerk can be given as,

$$J = (\hat{\boldsymbol{Y}} - \boldsymbol{Y}_d)^T\boldsymbol{Q}(\hat{\boldsymbol{Y}} - \boldsymbol{Y}_d) + \hat{\boldsymbol{U}}^T\boldsymbol{R}\hat{\boldsymbol{U}} + \Delta\hat{\boldsymbol{U}}^T\boldsymbol{S}\Delta\hat{\boldsymbol{U}} \tag{11}$$

where $\boldsymbol{Q} \in \mathbb{R}^{3N\times 3N} \geq 0$, $\boldsymbol{R} \in \mathbb{R}^{3N\times 3N} > 0$ and $\boldsymbol{S} \in \mathbb{R}^{3N\times 3N} > 0$ are the respective weight matrices of the three terms of the cost function. The optimal control input calculated by minimizing the cost function J will result in minimum jerk trajectory tracking by the Quadcopter.

3.3 Constraints

To obtain optimal control input for trajectory tracking of Quadcopter with a minimum jerk, the following constraints are imposed on the optimization problem.

3.3.1 Obstacle Avoidance Constraints. To ensure obstacle avoidance while tracking trajectory with a minimum jerk, an on-demand collision avoidance method is used to develop obstacle avoidance constraints. The constraints developed by the on-demand collision avoidance method are active only when it detects a collision with an obstacle in future with information of last predictions, i.e. when the following condition,

$$d = \|\hat{\boldsymbol{y}}[k_c|k-1] - \boldsymbol{O_b}\| < r_{min} \tag{12}$$

holds true. Here, (k_c+k-1) is the first time-step when the collision is detected using the information of the predictions obtained at $(k-1)^{th}$ instant, $\boldsymbol{O_b}$ is the position of the static obstacle, d is the distance between the Quadcopter and the obstacle at the instant when the collision is detected and r_{min} is the minimum distance that needs to be ensured between the Quadcopter and the obstacle in order to avoid the collision. Therefore, to ensure obstacle avoidance, the Quadcopter must attain a position such that,

$$\|\hat{\boldsymbol{y}}[k_c-1|k] - \boldsymbol{O_b}\| \geq r_{min} \tag{13}$$

is satisfied. The above equation (13) is linearized using Multivariable Taylor Series Expansion and rearranged to derive the obstacle avoidance constraints in the following form,

$$\boldsymbol{a_{coll}}\hat{\boldsymbol{U}} \leq b_{coll} \tag{14}$$

where $\boldsymbol{a_{coll}} = -\mu^T\boldsymbol{C'}\boldsymbol{B'}$ and $b_{coll} = \mu^T\boldsymbol{C'}\boldsymbol{A'}\bar{\boldsymbol{y}}[0|k] - \rho$. Here,

$$\begin{aligned} &\mu = \begin{bmatrix} \boldsymbol{0}^T_{3(k_c-1)\times 1} & v^T & \boldsymbol{0}^T_{3(N-k_c)\times 1} \end{bmatrix}^T, \\ &\rho = r_{min}d - d^2 + v^T\hat{\boldsymbol{y}}[k_c|k-1] \text{ and} \\ &v = (\hat{\boldsymbol{y}}[k_c|k-1] - \boldsymbol{O}_b). \end{aligned} \tag{15}$$

3.3.2 Control Input Bounds. The upper and lower bounds ensure that the control inputs are generated within the physical limits of the Quadcopter, i.e.,

$$\boldsymbol{lb} \leq \hat{\boldsymbol{U}} \leq \boldsymbol{ub} \tag{16}$$

where $\boldsymbol{lb} \in \mathbb{R}^{3N}$ and $\boldsymbol{ub} \in \mathbb{R}^{3N}$ are the lower and upper bounds on the control input, respectively. The equation (17) can be written in the form of inequality constraints as,

$$\begin{bmatrix} -\boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix}\hat{\boldsymbol{U}} \leq \begin{bmatrix} -\boldsymbol{lb} \\ \boldsymbol{ub} \end{bmatrix} \tag{17}$$

where $\boldsymbol{I} \in \mathbb{R}^{3N\times 3N}$ is an identity matrix. The obstacle avoidance constraints (14) and the bounds (17) are concatenated to be written as following inequality constraints,

$$\boldsymbol{A_{ineq}}\hat{\boldsymbol{U}} \leq \boldsymbol{b_{ineq}} \tag{18}$$

where $\boldsymbol{A_{ineq}} \in \mathbb{R}^{(6N+1)\times 3N} = \begin{bmatrix} \boldsymbol{a}^T_{coll} & -\boldsymbol{I}^T & \boldsymbol{I}^T \end{bmatrix}^T$ and $\boldsymbol{b_{ineq}} \in \mathbb{R}^{(6N+1)\times 1} = \begin{bmatrix} b^T_{coll} & -\boldsymbol{lb}^T & \boldsymbol{ub}^T \end{bmatrix}^T$. The quadratic optimization problem with obstacle avoidance constraints and bounds as inequality constraints is defined as,

$$\begin{aligned} \min_{\hat{\boldsymbol{U}}} \quad & J \\ \text{s.t.} \quad & \boldsymbol{A_{ineq}}\hat{\boldsymbol{U}} \leq \boldsymbol{b_{ineq}} \end{aligned} \tag{19}$$

The solution of the quadratic optimization problem yields optimal control input which will ensure trajectory tracking with a minimum jerk.

3.4 Asynchronous Path Smoothing

Model Predictive Control predicts the optimal control inputs and future states of the system up to the prediction horizon based on the system model and current state. MPC keeps on generating the smooth trajectory up to the prediction horizon for the system but practically this doesn't show up on the hardware as there are random aerodynamic disturbances involved due to which the actual next state is generally different from the predicted one and thus when this actual state is fed back to the MPC. The MPC generates the different control inputs for that step which will be definitely more than the previously predicted one in order to attain the desired state back again. This will happen repeatedly because the disturbances will always be there on each step. This repeated generation of increased control inputs on every step will make the trajectory followed by the quadcopter jerky and also power-consuming. This can be eliminated with the help of APS in which we give the first predicted state as the actual state of the system until the error exceeds an error threshold. Since the first predicted state is being fed back to the MPC, the MPC will generate the control inputs according to this state only assuming that the first predicted state is acquired hence these control inputs will lead to a smooth trajectory, whenever the error becomes more than the specified limit, the correction is made by providing the real state of the system to the MPC. This can be written mathematically as shown below. The error can be defined as: -

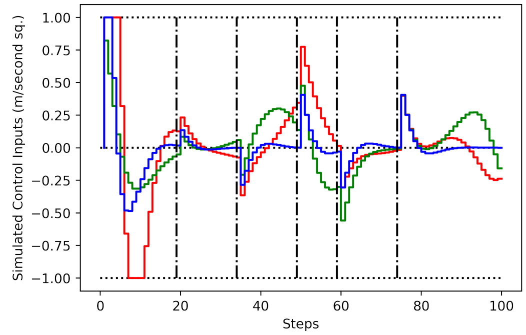
$$e[k] = \|\bar{\boldsymbol{y}}[k] - \hat{\boldsymbol{y}}[1|k-1]\| \tag{20}$$

$$\bar{\boldsymbol{x}}[0|k] = \begin{cases} \bar{\boldsymbol{x}}[k] & \text{if } e[k] \geq \bar{e} \\ \bar{\boldsymbol{x}}[1|k-1] & \text{else} \end{cases} \tag{21}$$

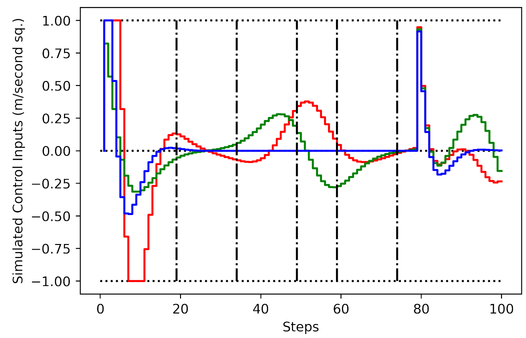
where $\bar{e}$ is error threshold. By using the above framework, simulation results are generated first without using APS or can be said by using Continuous Path Smoothing (CPS) and then by using APS over a lemniscate trajectory which is followed by a quadcopter. In Fig. 2, the vertical dash-dotted lines represent the steps at which the disturbances are provided. The red, green and blue line signifies the acceleration in x, y and z axis respectively. The disturbances are provided five times, the same in each of the two cases. In the case of CPS in Fig. 2a, we can see the control inputs changing drastically at the time when disturbances are provided but in the case of APS in Fig. 2b the control inputs have changed drastically only once in the five-time and that was the time when the error has exceeded the $\bar{e}$. Here, $\bar{e} = 0.3m$. The more drastic changes in the control inputs are there the more will the jerks be on the hardware and from both the Fig. 2a and Fig. 2b we can see the jerks have reduced significantly.

3.5 Acceleration to Plant Input Mapping

Most rotary wing-type UAVs use a Flight Control Unit (FCU) which is a low-level flight controller that decides the angular velocity of each rotor actively by this it manages thrust on each rotor actively. It also provides the flexibility to take high-level control inputs which are roll (ϕ), pitch (θ), yaw (ψ) and thrust (T). Where ϕ, θ & ψ are the ZYX Euler Angles which provide the rotational mapping



(a) Control Inputs vs Steps in case of CPS



(b) Control Inputs vs Steps in case of APS

Figure 2: Comparision of control inputs between CPS and APS. The ramp disturbances are provided for the duration of 0.2 seconds at times steps 20, 35, 50, 60 and 75 which are represented by Vertical dash-dotted lines. The red, green and blue line signifies the acceleration in x, y, and z, respectively.

between the ground or local coordinate frame and the body fixed frame. The rotation from ZYX Euler Angles can be obtained first rotating by an angle ψ about the current z-axis, then by an angle θ about the current y-axis and lastly by an angle ϕ about the current x-axis. By proving a set of (ϕ, θ, ψ, T) inputs the FCU actively tries to maintain a fixed orientation of the UAV according to (ϕ, θ, ψ) in space with respect to the ground frame of reference and a T amount of collective thrust. The framework that we have proposed can be applied to any kind of rotary wing type UAV having (ϕ, θ, ψ, T) or equivalent quaternions and thrust as its control input. One simple example of a linearised model to hardware mapping in the case of quadcopter can be seen below as: -

$$\psi = 0 \tag{22}$$

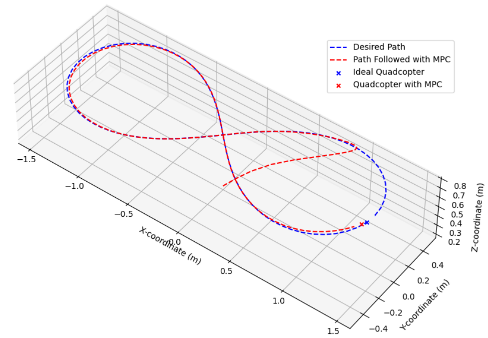
$$\boldsymbol{u} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \begin{bmatrix} \theta & \phi & T \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \tag{23}$$

In the above example, since the yaw (ψ) is always zero hence pitching will directly result in the acceleration in the x-axis and rolling will directly result in the acceleration in the y-axis with respect to the ground frame. Here the relation between the acceleration in the x-direction and pitch angle is fitted by a linear line, similarly in the y-direction. The relation between the acceleration in the z-direction and thrust is also fitted by a linear line as stated in the equation (23). This mapping can be done differently for different types of UAVs. For example, to fit a highly non-linear relation piece-wise linear fitting can be used and if the yaw angle is also varying while following the trajectory then the mapping used will be different.

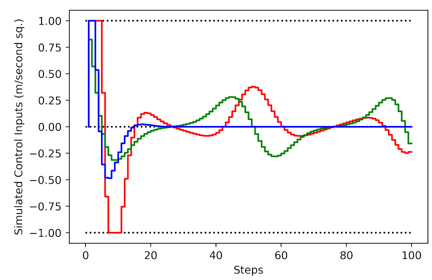
4 RESULTS & DISCUSSIONS

The proposed framework was validated by numerical simulations, assuming that the Quadcopter follows the model presented in Section 2.1 with $h = 0.2s$. The results are generated for a Quadcopter tracking a trajectory in presence of an obstacle and external disturbances using the following input parameters, $N = 20$, $\boldsymbol{Q} = \boldsymbol{I}$, $\boldsymbol{R} = 0.04\boldsymbol{I}$, $\boldsymbol{S} = 0.005\boldsymbol{I}$, where $\boldsymbol{I} \in \mathbb{R}^{3N \times 3N}$, $\bar{e} = 0.3m$, $r_{min} = 0.25m$ and the lower and upper bounds on the control inputs are $-1m/s^2$ and $1m/s^2$, respectively. We generated a lemniscate trajectory, which is considered to be a benchmark trajectory when it comes to trajectory tracking of autonomous vehicles, as the desired trajectory for $20s$, with the distance between the center of the trajectory and one of the foci as $1m$, Fig. 3a. In Fig. 3a, the blue dashed curve represents the desired trajectory and the red dashed curve represents the path traced by the Quadcopter, the Quadcopter takes off from $(0, 0, 0.2)$ and it is visible that the Quadcopter successfully tracks the desired trajectory. In Fig. 3b, the red, green, and blue colours represent the generated control inputs, i.e., acceleration in the x, y, and z direction, respectively.

To validate the robustness of the proposed framework in presence of obstacles, an obstacle was introduced in the simulation environment at $(0, 0, 0.75)$, represented by a black solid circle in Fig. 4a. The position of the obstacle was strategically chosen such that it forces the Quadcopter to perform obstacle avoidance twice while tracking the trajectory only once. The Quadcopter was able to track the lemniscate trajectory while avoiding collision with a static obstacle, as shown in Fig. 4a. In Fig. 4b, the red, green, and blue coloured, dotted and solid curves represent desired and measured positions in x, y, and z direction, respectively; here, it can be observed that the Quadcopter tries to maintain altitude while ensuring collision avoidance by deviating more in x and y direction and remaining close to the desired position in z direction. The results of the implementation of complete framework, i.e., trajectory tracking with APS in the presence of obstacle and external disturbances are presented in Fig. 5. In Fig. 5a, the black solid circle represents the static obstacle and the green circles represent the external disturbances, the Quadcopter tracks the lemniscate trajectory in the presence of obstacle while dealing with external disturbances using APS. Fig. 5b represents the position of the Quadcopter with respect to time-steps, in particular direction; here, the red, green, and blue coloured, dotted and solid curves represent desired and measured positions in x, y, and z direction, respectively. The Quadcopter experienced external disturbances at time-steps 20, 50, and 65, depicted by vertical black lines in Fig. 5b, it can be observed that the Quadcopter does not deviate from its desired trajectory when the error remains within the threshold limit, i.e., at time-steps 20 and 50, but deviates from its trajectory only when the error exceeds the threshold limit, i.e. at time-step 65, thereby performing path smoothing when required. In Fig. 5b, the green region represents the time-steps when the Quadcopter successfully avoids collision with the static obstacle. It is worth noting that the presented results confirms the efficacy of the proposed framework for trajectory tracking in the presence of obstacle and external disturbances.



(a) Motion Plot of the Quadcopter

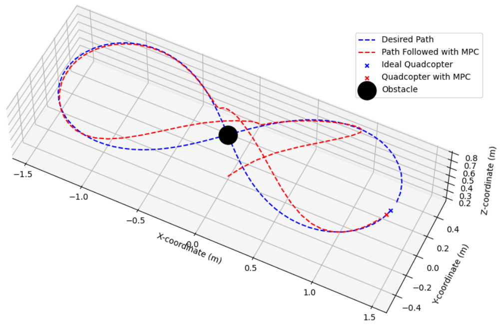


(b) Control Inputs of the Quadcopter with Steps

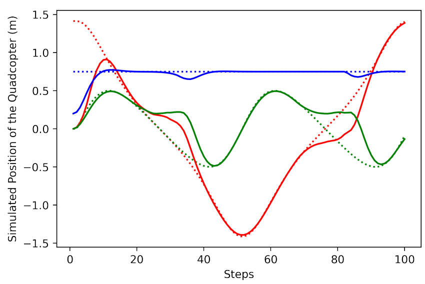
Figure 3: Quadcopter taking off and following the desired trajectory. In (a) the blue path represents our desired path and the red path represents the path followed by MPC.In (b) the red, green and blue line signifies the acceleration in x, y and z axis respectively.

5 CONCLUSION

In this work, a generalised MPC-based framework is developed using on-demand collision avoidance constraints and APS for smooth trajectory tracking of rotary wing UAVs, in presence of obstacles and external disturbances. The MPC uses a simplified linear discrete-time kinematic model of the system instead of the complex non-linear model thus, which is less computationally complex over other models. Also, the model is applicable to any rotary wing UAV having roll, pitch, yaw and collective thrust as the control input. An input mapping is required to be defined to fed the control inputs from MPC to the UAV. The results are simulated using linearised input mapping for the case of quadcopter. The cost function in MPC is designed with three terms in which the first term penalise the error in the trajectory tracking, the second term penalise the control input generation to ensure the minimal generation of control inputs and hence lower the power consumption of the system and the last term penalise the change in control input to reduce the

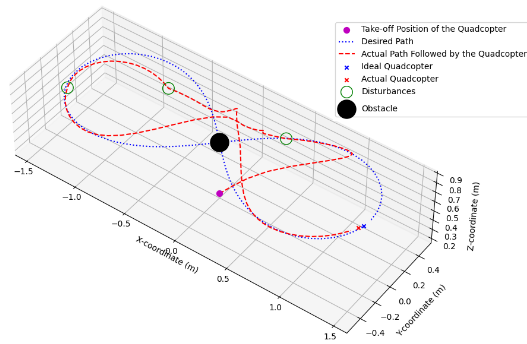


(a) Motion Plot of the Quadcopter with Static Obstacle

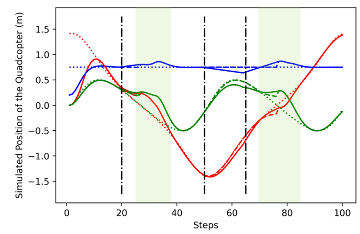


(b) Positions of the Quadcopter vs Steps

Figure 4: Quadcopter taking-off and tracking the lemniscate trajectory in the presence of an Obstacle. In (a) the black solid represents the obstacle, the blue path represents the path in which collision is happening and the red path represents the path in which collision is avoided using MPC.In (b) the red, green, and blue coloured, dashed and solid curves represent desired and measured positions in x, y, and z, respectively.



(a) Motion Plot of the Quadcopter with Static Obstacle and APS



(b) Positions of the Quadcopter vs Steps

Figure 5: Lemniscate trajectory tracking by Quadcopter in presence of an obstacle and asynchronous path smoothing. In (a), the black solid circle represents the static obstacle and the green circles represent the external disturbance. In (b) the red, green, and blue coloured, dotted and solid curves represent desired and measured positions in x, y, and z directions, respectively.

generation the aggressive control inputs. The simulation results are generated for lemniscate trajectory tracking problem for the case of quadcopter and observed that the quadcopter have successfully followed the trajectory. An obstacle is also introduced in the same trajectory to ensure the obstacle avoidance capability of the framework and also observed the obstacle has successfully avoided by keeping a safe distance. The results are also generated to prove the jerk reduction capability of the framework where the control inputs from CPS and APS are compared and the clear observation is made in the reduction of jerks. Then to check the efficacy of the framework the results for obstacle avoidance with external disturbance are generated on the lemniscate trajectory and observed the successful obstacle avoidance along with the reduction in jerks. Further studies can be carried out to test the framework on the hardware. The studies can be carried out on the multiple vehicles with static and dynamic obstacles.

REFERENCES

[1] Sabir Abdelhay and Alia Zakriti. 2019. Modeling of a quadcopter trajectory tracking system using PID controller. *Procedia Manufacturing* 32 (2019), 564–571.

[2] Ademola Abdulkareem, Victoria Oguntosin, Olawale M Popoola, Ademola A Idowu, et al. 2022. Modeling and Nonlinear Control of a Quadcopter for Stabilization and Trajectory Tracking. *Journal of Engineering* 2022 (2022).

[3] Huan Cheng and Yanhua Yang. 2017. Model predictive control and PID for path following of an unmanned quadrotor helicopter. In *2017 12th IEEE conference on industrial electronics and applications (ICIEA)*. IEEE, 768–773.

[4] Shreyash Gupta, Saurabh Chaudhary, Deepak Maurya, Shyam K Joshi, Niladri S Tripathy, and Suril V Shah. 2022. Segregation of Multiple Robots Using Model Predictive Control With Asynchronous Path Smoothing. In *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 1378–1383.

[5] M Islam, M Okasha, and MM Idres. 2017. Dynamics and control of quadcopter using linear model predictive control approach. In *IOP conference series: materials science and engineering*, Vol. 270. IOP Publishing, 012007.

[6] Maidul Islam, Mohamed Okasha, and Erwin Sulaeman. 2019. A model predictive control (MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking. *International Journal of Control, Automation and Systems* 17, 11 (2019), 2819–2832.

[7] Carlos E Luis and Angela P Schoellig. 2019. Trajectory generation for multiagent point-to-point transitions via distributed model predictive control. *IEEE Robotics and Automation Letters* 4, 2 (2019), 375–382.

[8] José Luis Mendoza-Soto and H Rodríguez Cortés. 2017. Generalized predictive control for trajectory tracking of a quadcopter vehicle. In *2017 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 206–212.

[9] Mohamed Okasha, Jordan Kralev, and Maidul Islam. 2022. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. *Aerospace* 9, 6 (2022), 298.

[10] Durgesh Haribhau Salunkhe, Siddhant Sharma, Sujal Amrit Topno, Chandana Darapaneni, Amol Kankane, and Suril V Shah. 2016. Design, trajectory generation and control of quadrotor research platform. In *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*. IEEE, 1–7.

[11] Niklas Schmid, Jonas Gruner, Hossam S Abbas, and Philipp Rostalski. 2022. A real-time GP based MPC for quadcopters with unknown disturbances. In *2022 American Control Conference (ACC)*. IEEE, 2051–2056.

[12] Guillem Torrente, Elia Kaufmann, Philipp Föhn, and Davide Scaramuzza. 2021. Data-driven MPC for quadrotors. *IEEE Robotics and Automation Letters* 6, 2 (2021), 3769–3776.