Predictive Control of Aerial Robots for Safety critical operations using Barrier Lyapunov function

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering
by Research

by

Vedant Mundheda
2018112006
`vedant.mundheda@research.iiit.ac.in`

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
June, 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Predictive Control of Aerial Robots for Safety critical operations using Barrier Lyapunov function” by Vedant Mundheda, has been carried out under my supervision and is not submitted elsewhere for a degree.

June 26, 2023	Adviser: Dr. Harikumar Kandath

Think like a Controller, One step ahead

Acknowledgments

I want to thank my advisor, Dr. Harikumar Kandath, for his constant support and guidance in my research. Dr. Kandath's passion for drones has been pivotal in defining my research problem and pushing me to explore varied ideas. I am thankful for the chance to work under his mentorship and learn from his vast knowledge and experience. I aim to make him proud as I move on to the next step of my career.

Secondly, I would like to thank the professors working in the Robotics Research Center (RRC) for designing courses that helped me gain advanced knowledge of robotics concepts. Dr. Harikumar Kandath, Prof. Madhava Krishna, Dr. Nagamanikandan Govindan, and Dr. Spandan Roy have been great sources of knowledge and guidance throughout my years at RRC.

Moreover, I want to thank Karan Mirakhor, Rahul Swayampakula and Damodar Datta for being the best project partners and helping me with my research and courses. I also want to thank Dipanwita Guhathakurta for her help in formulating my research problem, completing my coursework and being an awesome sister. I also express my gratitude to Sanskar Tibrewal, Dhruv Arya, Shivaan Seghal, Nishant Sachdeva, KV Aditya, Nomaan Qureshi, Gunjan Gupta, Aman Singh, Abhijit Srivastava, Parth Shah, Sanket Kalwar, Rishabh Dev Yadav, and Bitla Bhanu Teja for supporting my research work, giving technical advice and believing in me.

Lastly, I want to thank my sister Juhi and my parents for their constant love, moral support, and valuable life advice, which helped me cope with my studies and research.

Abstract

The issue of ensuring safety for aerial robots when navigating near obstacles remains a significant concern. There are three distinct aspects to addressing these safety concerns: the controller, which must provide assurances of safety while following a prescribed trajectory; the trajectory planner, which must generate a trajectory that avoids all obstacles; and the perception module, which must accurately localize the robot and obstacles with minimal deviation from actual ground truth. This research study focuses specifically on the first aspect, which pertains to the safe control of aerial robots when maneuvering near static obstacles. The study considers two types of aerial robots, namely an aerial manipulator and a quadrotor unmanned aerial vehicle.

The AM considered here is a 2-DOF (degrees-of-freedom) manipulator rigidly attached to a UAV. Our proposed controller structure follows the conventional inner loop PID control for attitude dynamics and an outer loop controller for tracking a reference trajectory. The outer loop control is based on the Model Predictive Control (MPC) with constraints derived using the Barrier Lyapunov Function (BLF) for the safe operation of the AM. BLF-based constraints are proposed for two objectives, viz. 1) To avoid the AM from colliding with static obstacles like a rectangular wall, and 2) To maintain the end effector of the manipulator within the desired workspace. The proposed BLF ensures that the above-mentioned objectives are satisfied even in the presence of unknown bounded disturbances. The capabilities of the proposed controller are demonstrated through high-fidelity non-linear simulations with parameters derived from a real laboratory scale AM. We compare the performance of our controller with other state-of-the-art MPC controllers for AM.

To address the problem of flying a UAV inside a tunnel, we propose the implementation of a Model Predictive Control (MPC) framework with constraints based on Control Barrier Function (CBF). The thesis approaches the issue in two distinct ways; first, by maintaining a safe distance from the tunnel walls to avoid the effects of both the walls and ceiling, and second, by minimizing the distance from the walls to effectively manage the nonlinear forces associated with close proximity tasks. Finally, the paper demonstrates the effectiveness of its approach through testing on simulation for various close proximity trajectories with the realistic model of aerodynamic disturbances due to the proximity of the ceiling and boundary walls.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

1.1 Scope of the Thesis

The focus of this thesis is centered on control systems utilized in safety-critical operations of Aerial robots. Our discussion pertains specifically to the challenges encountered by aerial robots operating in close proximity to static obstacles, such as buildings. We provide an in-depth analysis of the challenges faced by current state-of-the-art control systems and motivate towards the development of control algorithms in this context.

1.1.1 Control Systems

The application of control systems has been a pivotal aspect in the development of automation and robotics since the inception of the first robot in the 1950s. These systems are responsible for executing both low-level movements and high-level instructions in various systems, ranging from chemical plants to sophisticated dynamical robots. In the context of a robotic system, the controller assumes responsibility for governing the actions of the actuators while concurrently mitigating the discrepancy between the desired trajectory or position. This controller can be bifurcated into two distinct modules: firstly, the high-level controller, which formulates the requisite actuator torques or angles aimed at minimizing the position error, and secondly, the low-level controller, which diligently follows the desired torques or angles to accurately navigate the robot. The components of a robotic system are explained in Fig. 1.1.

These components are summarized below:

1. The sensor stack supplies pertinent data regarding the environment to the processing unit, which in turn utilizes this information to develop a comprehensive comprehension of the surrounding which is leveraged by the planner.

2. The Planner generates smooth trajectory points for the robot by avoiding obstacles to achieve the goal. These tasks can include reaching a goal location, or navigating from one place to another.

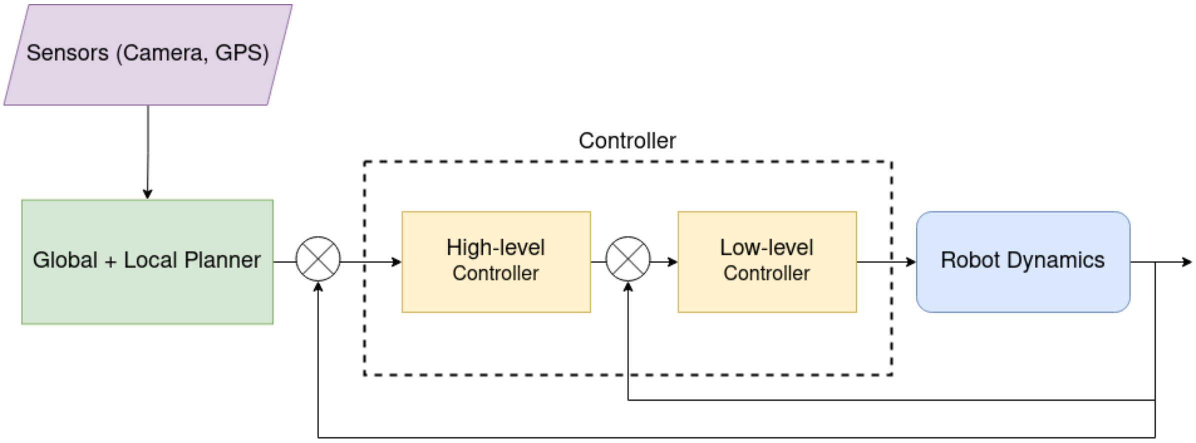

Figure 1.1: Layout for an autonomous robotic system: The Controller works on a feedback loop to achieve desired system state given by the planner.

3. The High-level controller generates the desired actuator states using the desired trajectory as provided by the Planner.

4. The Low-level controller tracks the desired actuator states using a feedback loop which provides with the current actuator information after the robot dynamic system performs the given input.

Although a planner defines a safe trajectory for a robot, a significant safety obstacle arises when the robot inaccurately tracks the trajectory or deviates from the intended path due to external disturbances. Consequently, ensuring stability and safety are primary concerns for any control system.

1.1.2 Aerial Robots: Safety Critical Operations

Unmanned Aerial Vehicles (UAVs) have been used for various tasks in the past decade. Present times have witnessed the widespread deployment of UAVs in numerous domains, ranging from delivery, search, and rescue to monitoring [2, 3]. Some other tasks include surveillance, object detection, and civil infrastructure inspection. Civil inspection and delivery tasks necessitate close-range operations near stationary obstructions, such as bridges and buildings. Similarly, when a floating-base manipulator is attached to these UAVs, it can be utilized for various manipulation tasks [4, 5, 6]. Aerial Manipulators (AMs) have gained much attention in recent years. The Unmanned Aerial Vehicle (UAV) acts as a floating base for the manipulator enabling it to conduct active operations in the 3D space. This combination of a UAV and a manipulator gives the system enough capabilities to perform complex operations where human access is limited (e.g., in disaster-affected zones) [7]. The other industrial and commercial applications of AMs are in the maintenance of power grids, the inspection of bridges, and canopy sampling.

Figure 1.2: Shows current use of UAVs for inspection tasks near bridges. Source: https://candrone.com/pages/inspection

Such close proximity operations require maneuvering near these static obstacles [8, 9]. It is unarguable that any system designed for these applications must be contructed with safety in mind [10].

In fact, safety has been a major hurdle in deploying such aerial manipulator systems in these applications [11]. The challenges faced in performing close proximity maneuvers are twofold: firstly, aerodynamic effects caused by ground, ceiling, and wall interference destabilize the drone and frequently result in collisions with obstacles [12, 13]; secondly, the highly coupled dynamics of the UAV and manipulator can generate undesirable forces and torques [14, 15]. Wind disturbances further exacerbate the situation by destabilizing the UAV and increasing the likelihood of collisions [16].

1.2 Research problems tackled

A *To design a control algorithm to incorporate safety guarantees for trajectory tracking of aerial robots.*

A safety constraint needs to be imposed on the underlying controller to provide safety guarantees while flying in outdoor areas. These constraints should also tackle obstacle avoidance even when the desired trajectory is close to the obstacles. Hence, the controller needs to predict the external forces and the state of the robot in the next time steps to avoid obstacles safely. Hence, we develop a model predictive controller with constraints from Barrier lyapunov function which incorporates the predictive nature of the controller with obstacle avoidance constraints.

B *To efficiently tackle non-linear dynamics of complex systems as an aerial manipulator in real-time.*

Aerial manipulator is a combination of a floating manipulator attached on a UAV base. The coupled dynamics of such systems is non-linear which exerts many unwanted torques on the UAV making it unstable. To tackle this, the aerial manipulator model is given to the predictive controller which predicts these forces and gives control inputs. Similarly, modelling of robotic system enhances understanding of the system which in turn improves the stability.

C *To incorporate constraints to tackle external disturbances in the form of aerodynamic effects near static objects and strong winds in the high-level controller.*

While operating near static obstacles, the aerial robot faces a lot of forces and torques due to the aerodynamic interactions of the rotor and the walls. These usually pull the robot towards the wall leading to collision and/or instability. Additional constraints to handle these forces must be incorporated to the underlying controller with disturbance rejection properties to handle these forces. This would also make the system disturbance tolerant for a bounded unmeasured disturbances.

1.3 Motivation

In this section, we discuss the various methods problems in earlier approaches of solving the research threads addressed in this thesis, which inspire certain aspects of our approach. We also mention alternative methods that might share similar inspiration or approaches but do not necessarily produce successful outcomes, and provide reasons for their shortcomings.

1.3.1 Control of an Aerial Manipulator near static obstacles

While the literature is sufficiently populated with novel design approaches of AMs, prior work involving safe control of the AM has been sparse [17, 18, 19]. Adaptive controllers [20, 21] have been used to tackle torques due to the highly coupled dynamics of the AM by using an outer loop adaptive control over the proportional–derivative (PD) inner loop of the UAV. Though such strategies provide computational efficiency, they are incapable of incorporating constraints to avoid obstacles. Model Predictive Control (MPC) [22, 23, 24] significantly reduces the abruptness in control inputs and tracks the desired trajectory while anticipating future dynamic interactions of the AM. AMs also need to tackle impulsive torques whenever the arm tries to manipulate a object which further might lead to instability.

Barrier Lyapunov functions (BLFs) [1, 25] have been combined with MPC for obstacle avoidance of UAV as they can provide safety guarantees even in 3-dimensions. Safety is ensured by guaranteeing forward invariance of a safe region. BLF-based MPC [26, 27] for a non-linear system have been proposed as stabilizing controllers to guarantee avoidance of a set of states associated with the unsafe region.

For operation near static obstacles, MPC [28, 29, 30] has been utilized with hard constraints on the optimizer to avoid states which might lead to collision. These kinds of constraints don't incorporate velocity of the UAV and are not able to enforce the desired bound. Certain MPC algorithms have been formulated by incorporating these as soft constraints inside the MPC cost, which work satisfactorily when the constraint weight is high, but restrict the motion of the AM hindering effective trajectory tracking.

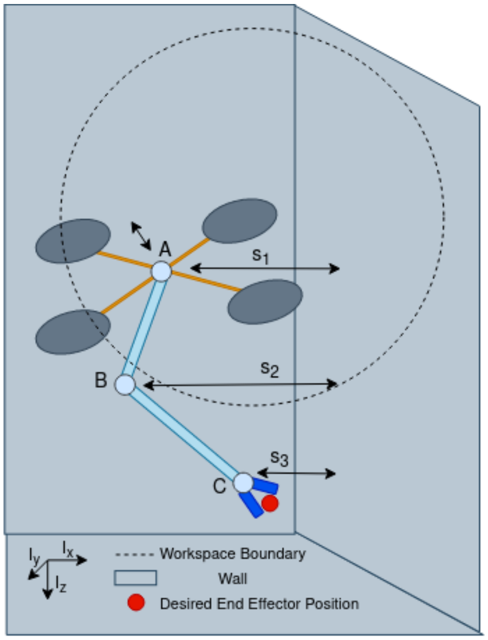

Figure 1.3: Image shows the Operation of an AM near 3 walls and the desired workspace for the AM.

We propose an MPC with constraints using BLF [31, 32] , which provides safety guarantees while giving leeway to the MPC for effective trajectory tracking. It is also utilized to bind the AM into a safe region whenever obstacle positions are unknown. We modify the BLF earlier used for obstacle avoidance into a constraint for bounding the AM into a safe region. We also introduce a novel disturbance rejection term which prevents the AM from leaving the safe region and colliding with obstacles under external disturbances. The problem statement can be better understood as given in Fig. 1.3.

1.3.2 Safe Maneuver of an Unmanned Aerial Vehicle inside a Tunnel

Indoor operations of UAVs have been restricted due to the problems faced by UAVs near walls. Motion of a UAV in a tunnel has various challenges, primarily the aerodynamic effects due to the turbulent forces produced by the rotor and wall interactions. To account for such disturbances from all directions, we demonstrate our controller for operating inside a tunnel.

Nonlinear Model Predictive Control (MPC) [24] has been used for navigation and obstacle avoidance of UAVs for real-time utilities. MPC provides the predictive ability [23] which aids in performing agile maneuvers with high precision and smooth control actions. [26] tries to limit the risk of unsafety by formulating a probabilistic guarantee, but fails to provide a rigid safety guarantee to avoid obstacles. [28] utilises partial sensor information to navigate through unknown environments by providing partial safety guarantees.

Control Barrier Function (CBF) [1] is used to guarantee safety-critical control for various domains, including dynamic robotic systems. [1] introduces safety, safety sets, and describes using CBF to enforce safety in a minimally invasive fashion by not increasing the control effort or trajectory cost. CBF has been used as a constraint to MPC [29] to provide safety guarantees while addressing the case of conflict between safety and performance. This provides improved performance to MPC while providing safety guarantees. [27] shows collision avoidance for multi UAV swarm to reach desired locations and providing safety guarantees. [31] utilizes MPC while handling external wind disturbances. Although nonlinear controllers have been tried separately for ground and ceiling effects, no effort has been made to minimize the impacts of these disruptions using a disturbance resistive barrier function.

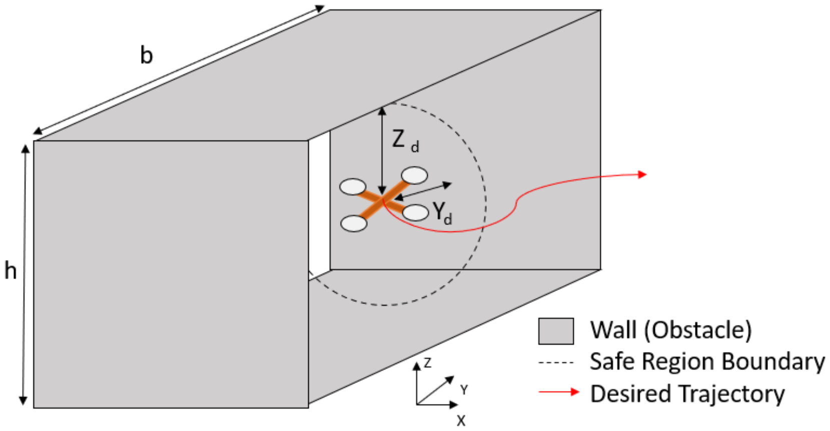

Figure 1.4: Image shows the trajectory tracking of a UAV inside a tunnel while handling aerodynamic effects from the walls.

We tackle disturbances due to aerodynamic interactions with wall by modifying the BLF. This modified constraint is added to the MPC controller which provides smooth trajectories even in the presence of unknown bounded disturbances. The problem statement can be better understood as given in Fig. 1.4.

1.4 Contributions

This thesis provides the following contributions:

1. To the best of the author's knowledge, this is the first attempt to design a predictive controller with constraints for Barrier Lyapunov Function (BLF) for an aerial manipulator.

2. The proposed controllers provide obstacle avoidance to the aerial robots while achieving tangible performance gain over existing predictive controllers.

3. We exploit the BLF to create a novel constraint to bound the UAV inside inside a defined safe boundary, which is contrary to it's earlier avoidance utility.

4. For a UAV, the proposed controller is the first attempt to handle all three effects (ground, ceiling, and wall).

5. We also introduce changes to the BLF to tackle external bounded wind disturbances by a distance rejection term.

1.5 Thesis Layout

C1 This chapter discusses the scope of this thesis describing the research problem tackled in terms of control systems for aerial robots. We outline the motivation behind the research problem and techniques others use for similar problems.

C2 In this chapter, we present the background on the dynamics of aerial robots, state space models, Model Predictive Control, and Barrier Lyapunov function. We explain how to formulate these for our proposed controller and provide details on their use.

C3 This chapter is the first contribution of this thesis. We demonstrate MPC-BLF as predictive control algorithms for an aerial manipulator with the ability to avoid obstacles and enclose the aerial manipulator in a confined safe space.

C4 In this chapter, we contribute to algorithm development for maneuvering a UAV through a tunnel and handling additional aerodynamic torques from interactions with the tunnel walls.

C5 In this chapter, we present the conclusion for the thesis with future prospects and endeavors. The thesis also proposes directions for newer research and practical use cases of the proposed algorithms.

Chapter 2

Background

In this chapter, we introduce dynamic modelling and control techniques such as Model Predictive Control (MPC) for aerial robots. We also mention known techniques and constraints which form the prerequisite for our work in the subsequent chapters.

Firstly, we present the known dynamic and kinematic model of a UAV and a manipulator. We convert the dynamic model of the manipulator into a floating base manipulator and combine the above dynamics to to formulate the dynamic and kinematic model of an Aerial Manipulator (AM).

Secondly, we explain a general state-space model of a UAV and demonstrate how to run an MPC on the general state-space model. We demonstrate how to form a cost function and utilize this optimal control technique by optimizing this cost and finding optimal control inputs.

Lastly, we explain the definition of a Barrier Lyapunov function (BLF) and it's various forms. Subsequently, we elaborate on how to utilize the BLF for obstacle avoidance and how to incorporate these conditions on an MPC.

2.1 Dynamic Model of Aerial Robots

Dynamic models of robots pertains to the relationship between the forces which act upon a robotic system and the acceleration they produce. They model the robot as a rigid-body system and define rigid-body dynamics to understand the robot's behaviour to varied forces and inputs.

Dynamic modelling problem can be divided into two points, firstly, forward dynamics which calculates the force when given acceleration, and secondly, inverse dynamics which calculates the acceleration when given the force.

Hence, the dynamic model consists of the following:

1. kinematic model for the mechanism of the robot

2. inertial parameters of the robot

2.1.1 Dynamic Model of a UAV

In this section we present the Dynamics Model of a quadcopter, floating base 2-DoF robotic arm and combined dynamics of Aerial Manipulator. Similar versions of the mathematical model using Newton-Euler Dynamics can be found in the paper[].

We denote the Inertial frame $G = [G_x, G_y, G_z]$ with the centre at O_I, the Body frame fixed to the drone is represented as $b = [b_x, b_y, b_z]$ centred at O_B. The Manipulator link frames $l = 1, 2$ are denoted as $M_l = [M_{x_l}, M_{y_l}, M_{z_l}]$ centred at O_l. The position of centre of Drone in inertial frame is $p_I = [x_I, y_I, z_I]$ and its orientation in Euler angle convention is $\Phi = [\phi, \theta, \psi]$. Similarly the 2-DoF manipulator joint angle are $\Theta = [\theta_1, \theta_2]$ defined about the positive direction of B_y. These are defined as in Fig. 2.1.

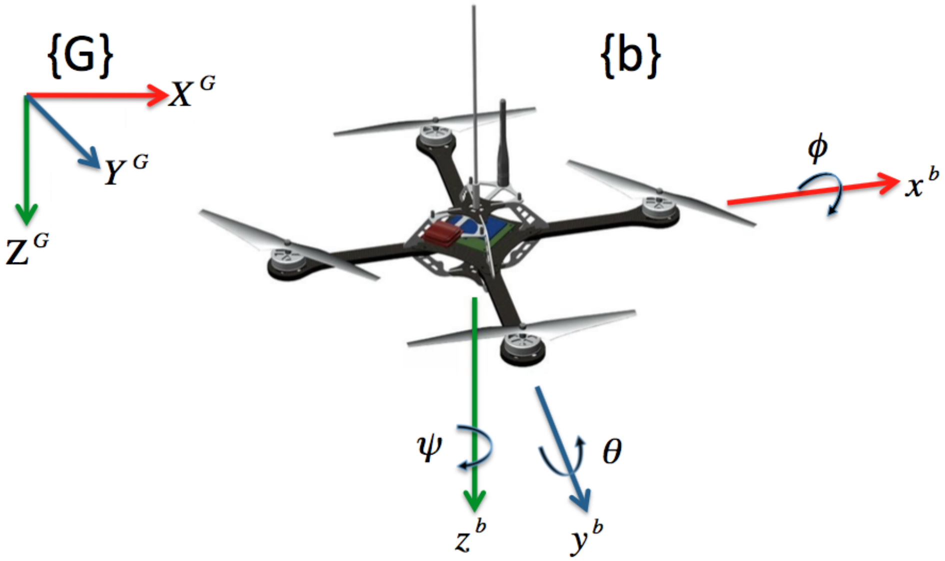

Figure 2.1: UAV showcasing roll, pitch and yaw angles with inertial frame {b} and ground frame {G}

Source: https://wilselby.com/research/arducopter/modeling/

Consider the angular speed of each rotor as ω_i, where $i \in \{1, 2, 3, 4\}$. k_f is the motor force constant which converts angular speed to thrust. The thrust vector T_B on the drone in the Body frame is given in 2.1.

$$T_B = \begin{bmatrix} 0 \\ 0 \\ -k_f \sum_{i=1}^{4} \omega_i^2 \end{bmatrix} \tag{2.1}$$

For transforming vectors in Body frame {b} to World frame {G} we use the standard rotation matrix ${}^{G}R_b \in SO(3)$. Now, The acceleration of the drone in the inertial frame is given in 2.2.

$$\begin{aligned} \ddot{p}_I &= \frac{1}{m_B}(G + T) \\ G &= m_B g e_3 \\ T &= m_B {}^{G}R_b T_B \end{aligned} \tag{2.2}$$

where m_B is the mass of drone, g is the gravity and $e_3 = [0, 0, 1]^T$ is the unit vector along inertial z-axis I_z.

The distance between the centre of Drone and the rotor is L in drone frame B and k_m is the motor moment constant. The torque on the drone in Body frame is given in 2.3.

$$\tau_B = \begin{bmatrix} Lk_f(\omega_1^2 - \omega_3^3) \\ Lk_f(\omega_4^2 - \omega_2^3) \\ k_m(\omega_1^2 - \omega_2^3 + \omega_3^3 - \omega_4^2) \end{bmatrix} \tag{2.3}$$

Using the torque τ_B and Moment of Inertia Matrix of the drone I_B, the angular acceleration $\dot{\omega}_B$ in the body frame is given in 2.4.

$$\dot{\omega}_B = I_B^{-1}(\tau_B - \omega_B \times I_B \omega_B). \tag{2.4}$$

Thus, the overall drone dynamics can be written as in 2.5

$$\begin{bmatrix} m_B I_3 & 0_{3\times3} \\ 0_{3\times3} & I_B \end{bmatrix} \begin{bmatrix} \ddot{p}_I \\ \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} 0_{3\times3} \\ \omega_B \times I_B \omega_B \end{bmatrix} = \begin{bmatrix} G + T \\ \tau_B \end{bmatrix} \tag{2.5}$$

where I_3 is a 3×3 identity matrix, $0_{3\times3}$ is a 3×3 null matrix. For converting ω_B to $\dot{\Phi}$ we use the transformation T as in 2.6.

$$T = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi sec\theta & cos\phi sec\theta \end{bmatrix} \tag{2.6}$$

Hence, the final equation for $\dot{\Phi}$ is given in 2.7.

$$\dot{\Phi} = T\omega_B \tag{2.7}$$

2.1.2 Dynamic model of a manipulator

For our use case we are using an RR open chain Manipulator consisting of 3 links numbered from 0 to 2 where 0 is the base link i.e. the quadcopter itself in this case and 2 is the end effector i.e. the

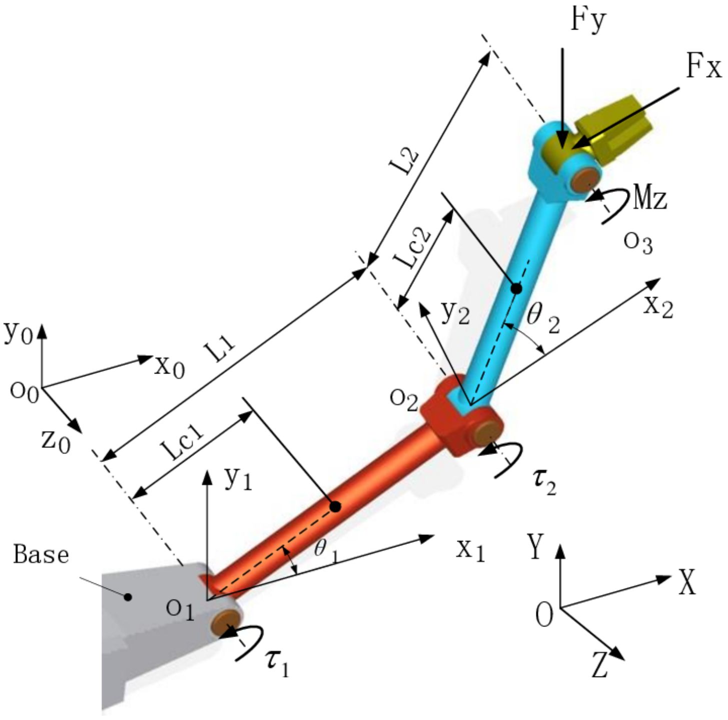

Figure 2.2: A 2R Manipulator Source: https://www.semanticscholar.org/paper/Dynamic-responses-of-a-2R-manipulator-in-state-by-Han-Zhang/a359ae06e7f1a9566c95a511e3f3928ea71f27e7/figure/0

gripper. Connecting two links are 2 joints numbered from 1 to 2 as in Fig. 2.2.

We follow a standard DH(Denavit-Hartenberg) convention in which link i-1 is connected to link i by joint i. Any link i is moved by joint i. Length a_i and twist α_i describes a link. Joints are described using the link offset d_i and joint angle θ_i. We have the coordinate frame i attached to the far end of the link i and the axis of joint i is aligned with the z_i axis.

In this convention the transformation from link coordinate frame i-1 to frame i is defined by rotation and translation as given in 2.8.

$$\begin{aligned} {}^{i-1}T_i(\theta_i, d_i, a_i, \alpha_i) &= T_{R_z}(\theta_i)T_z(d_i)T_x(a_i)T_{R_x}(\alpha_i) \\ {}^{i-1}T_i(\theta_i, d_i, a_i, \alpha_i) &= \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}t_i \\ 0_{3\times1} & 1 \end{bmatrix} \end{aligned} \tag{2.8}$$

As part of the Newton-Euler formulation we do forward recursion to find the velocities and acceleration of each link and then backward recursion to find the total force and torque experienced by each joint.

2.1.2.0.1 Forward Recursion We calculate the acceleration and velocities of each link recursively starting from base link to end-effector.

Some notations used in Forward recursion are :

$$
\begin{aligned}
\theta_i &= \text{the joint angle of joint i} \\
l_i &= \text{link lengths of link i} \\
m_i &= \text{mass of link i} \\
I_i &= \text{Inertia of link i} \\
r_{i,ci} &= \text{the vector from joint i to the CoM of link i} \\
r_{i+1,ci} &= \text{the vector from joint i+1 to the CoM of link i} \\
r_{i,i+1} &= \text{ the vector from joint i to joint i+1} \\
a_{c,i} &= \text{the acceleration of the center of mass of link i} \\
a_{e,i} &= \text{the acceleration of the end of link i} \\
\omega_i &= \text{the angular velocity of frame i w.r.t. frame i} \\
\alpha_i &= \text{the angular acceleration of frame i w.r.t. frame i} \\
z_{i-1} &= \text{the axis of rotation of joint i w.r.t. frame 0} \\
{}^{i}R_{i+1} &= \text{the rotation matrix from frame i + 1 to frame i}
\end{aligned}
$$

To start solving the forward recursion we begin with the initial conditions i.e. the base link velocities and accelerations. In our case the link 0 i.e. the base link is the drone itself and hence $a_{e,0} = \ddot{p}_B$, $\omega_0 = \omega_B$ and $\alpha_0 = \alpha_B$ where $\ddot{p}_B$, ω_B and α_B are linear acceleration, angular velocity and angular acceleration respectively of the drone in Body frame.
The equations of Forward recursion are given in 2.9.

$$
\begin{aligned}
\omega_i &= {}^{i}R_{i-1}\omega_{i-1} + {}^{i}R_0 z_{i-1}\dot{\theta}_i \\
\alpha_i &= {}^{i}R_{i-1}\alpha_{i-1} + {}^{i}R_0 z_{i-1}\ddot{\theta}_i + \omega_i \times {}^{i}R_0 z_{i-1}\dot{\theta}_i \\
a_{c,i} &= {}^{i}R_{i-1}a_{e,i-1} + \dot{\omega}_i \times r_{i,ci} + \omega_i \times (\omega_i \times r_{i,ci}) \\
a_{e,i} &= {}^{i}R_{i-1}a_{e,i-1} + \dot{\omega}_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1})
\end{aligned}
\tag{2.9}
$$

2.1.2.0.2 Backward Recursion With the computed accelerations and velocities we can now recursively compute the total joint force exerted on each link starting from end-effector to base link. Some notations used for Backward recursion are :

$$
\begin{aligned}
g_i &= \text{the acceleration due to gravity(expressed in frame i)} \\
f_i &= \text{the force exerted by link i-1 on link i} \\
\tau_i &= \text{the torque exerted by link i-1 on link i}
\end{aligned}
$$

To begin solving the Backward recursion we start with the terminal conditions i.e. f_{n+1} = 0 and τ_{n+1} = 0.

The equations of Backward recursion are given in 2.10.

$$\begin{aligned} f_i &= {}^{i}R_{i+1} f_{i+1} + m_i a_{c,i} - m_i g_i \\ \tau_i &= {}^{i}R_{i+1} \tau_{i+1} - f_i \times r_{i,ci} + {}^{i}R_{i+1} f_{i+1} \times r_{i+1,ci} \\ &\quad + I_i \alpha_i + \omega_i \times (I_i \times \omega_i) \end{aligned} \tag{2.10}$$

2.1.3 Dynamic model of an Aerial Manipulator (AM)

Finally we can write the Aerial Manipulator Dynamics by adding the forces and torques from the Manipulator dynamics to the Quadcopter dynamics. We can easily find the mass m_{am} and Moment of Inertia I_{am} terms of the combined system.

Hence, The coupled equations of motion for the quadrotor with manipulator is given in 2.11.

$$\begin{bmatrix} m_{am} I_3 & 0_{3\times 3} \\ 0_{3\times 3} & I_{am} \end{bmatrix} \begin{bmatrix} \ddot{p}_I \\ \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} 0_{3\times 3} \\ \omega_B \times I_B \omega_B \end{bmatrix} = \begin{bmatrix} G + T \\ \tau_B \end{bmatrix} + \begin{bmatrix} f \\ \tau \end{bmatrix} \tag{2.11}$$

here f and τ is the force and moment that is exerted by the manipulator at the base of the quadrotor.

2.2 State Space model

In the area of control engineering, a state-space model is a quantitative model of a physical system. It is characterised by a collection of input, output, and variable factors linked by differential equations or difference equations without second-order derivatives. The model's variables, known as state variables, change over time depending on their present values and the values of the input variables. The outcome variables' values are determined by the state variables' values.

A simple linear state space model are given in 2.12 and 2.13.

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{2.12}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{2.13}$$

where:
$\mathbf{x}(\cdot)$ is the "state vector", $x(t) Rn \mathbf{x}(t) \in \mathbb{R}^n$;
$\mathbf{y}(\cdot)$ is the "output vector", $y(t) Rq \mathbf{y}(t) \in \mathbb{R}^q$;
$\mathbf{u}(\cdot)$ is the "input (or control) vector", $u(t) Rp \mathbf{u}(t) \in \mathbb{R}^p$;
$\mathbf{A}(\cdot)$ is called the "state (or system) matrix", $\dim[\mathbf{A}(\cdot)] = n \times n$,
$\mathbf{B}(\cdot)$ is called the "input matrix", $\dim[\mathbf{B}(\cdot)] = n \times p$,
$\mathbf{C}(\cdot)$ is called the "output matrix", $\dim[\mathbf{C}(\cdot)] = q \times n$,
$\mathbf{D}(\cdot)$ is called the "feedthrough (or feedforward) matrix" , $\mathbf{D}(\cdot)$ is the zero matrix), $\dim[\mathbf{D}(\cdot)] = q \times p$,
$\dot{\mathbf{x}}(t) := \dfrac{d}{dt}\mathbf{x}(t)$.

2.2.1 Controllability

A continuous linear state-space model is controllable only when condition in 2.14 is satisfied.

$$\operatorname{rank}\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B}\end{bmatrix} = n \tag{2.14}$$

2.2.2 State Space of a UAV

The state space model of a 6 Degree-of-Freedom UAV is presented in 2.18.

The state vector are given below where Θ, Φ, Ψ depict the Pitch, Roll, Yaw respectively are given in 2.15.

$$x = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \Theta \\ \Phi \\ \Psi \\ \dot{\Theta} \\ \dot{\Phi} \\ \dot{\Psi} \end{bmatrix} \tag{2.15}$$

The input matrix is given in 2.16.

$$u = \begin{bmatrix} \mathbf{U_1} \\ \mathbf{U_2} \\ \mathbf{U_3} \\ \mathbf{U_4} \end{bmatrix} \tag{2.16}$$

Where,

- U_1 - Total upward force on the UAV $(T - mg)$
- U_2 - Pitch Torque (about x-axis)
- U_3 - Roll Torque (about y-axis)
- U_4 - Yaw Torque (about z-axis)

The output vector is given in 2.17.

$$y = \begin{bmatrix} x \\ y \\ z \\ \Theta \\ \Phi \\ \Psi \end{bmatrix} \tag{2.17}$$

The state space model is given in 2.18.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.18}$$

We utilize the inertia matrix I_x, I_y, I_z to get $\mathbf{A}$ and $\mathbf{B}$ as in 2.19.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{2.19}$$

The output equation (in matrix form) is given in 2.20:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{2.20}$$

And $\mathbf{C}$ and $\mathbf{D}$ matrix are given in 2.21.

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

From 2.18 and 2.19, we completely define the state space model of a 6DOF UAV.

2.3 Model Predictive Control

Model Predictive Control (MPC) is an approach to control a system by predicting the system's behaviour at future time steps. This future time horizon is called a prediction horizon and is computed over N time steps. MPC optimizes a performance index (cost) over the prediction horizon while handing non-linear constraints for a robust closed-loop behavior. The Performance index for the MPC can also handle multi variable input and output efficiently contrary to classical strategies like PID (Proportional Derivative and Integral Controller).

The optimization problem for a model predictive controller [33] for a UAV for trajectory tracking can be described as in 2.22. We use the discrete-time version of the state space equations hence t translates to k which is a single time step.

$$\min_{\mathbf{u}} l(\mathbf{x_k}, \mathbf{u_k}, t_k) \tag{2.22a}$$

$$\text{s.t. } \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{2.22b}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{2.22c}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max} \tag{2.22d}$$

where l is the cost function dependant on the state space model of the UAV. We can use any optimization technique for getting an optimal input u^{opt} like ADAM [34], SQP [35] etc. k depicts the time steps and N denotes the time horizon of the Model Predictive Control algorithm with is a tunable parameter.

2.4 Barrier Lyapunov Function

This section explains the various theory, applications, and uses of the Barrier Lyapunov Function. Barrier Lyapunov Functions (BLFs) are a type of Lyapunov function used to ensure constraints on a system while maintaining the stability of the system. Lyapunov functions are scalars that measure the energy of a system and are used to prove the stability properties of the system. At the same time, BLFs can enforce constraints on the system states and inputs.

BLFs are defined as negative functions inside the constraint set and positive outside of it, creating a ”barrier” that prevents the system from violating the constraints. They also satisfy the conditions of a Lyapunov function, including being positive definite, decreasing along the system trajectories, and having a global minimum at the equilibrium point.

Using BLFs in control design allows a systematic way to handle constraints in various applications, such as robotics, aerospace, and power systems. They have proven effective in theoretical analysis and practical implementation, making them valuable tools in control engineering. Below we present different types of Barrier Lyapunov Function: Zeroing Control Barrier Function and Control Barrier Functions.

2.4.1 Zeroing Control Barrier Function

In this section we would briefly introduce the fundamentals for Zeroing Control Barrier Function (ZCBF). For ZCBF we need to define a set of states which are safe i.e. all state space values which are permissible and then prove the forward invariance of the set i.e. if the system starts in the safe state initially we ensure that it remains there.

We first present some of the basics of the ZCBFs and later use them to ensure that the desired position of the end-effector is always in the workspace of the Aerial manipulator. We use the State space dynamics of a continuous time control affine form is given in 2.23

$$\dot{x} = f(x) + g(x)u \tag{2.23}$$

where f and g are locally Lipschitz continuous functions that describe the dynamics of the system and x $\in \mathbb{R}^n$ defines the state of the system, and u $\in \mathbb{R}^m$ is the control input to the system.

Now we define a set of $\zeta \in \mathbb{R}^n$ which describes the safe set of the system. We want to guarantee the forward invariance of the system i.e. if $x(0) \in \zeta$ then $x(t) \in \zeta, \forall t \geq 0$ by using the ZCBF function $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$.

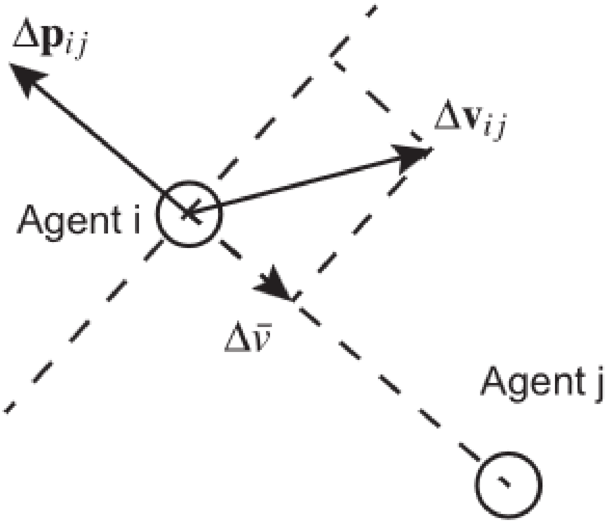

Figure 2.3: Object avoidance using a Control Barrier Function in a multi-agent setting Source: ”Control Barrier Functions: Theory and Applications” *2019 18th European Control Conference (ECC)* [1]

Here, $h(x) = 0$ for $x \in X$ at the boundary of the safe set ζ and $h(x) > 0$ for any $x \in X$ inside the safe set. Mathematically we represent it as in 2.24

$$\zeta = \{x \in \mathbb{R}^n \mid h(x) \geq 0\} \tag{2.24}$$

Now, formulation for the time derivative of $h(x)$ is given in 2.25

$$\frac{dh(x)}{dt} = L_f h(x) + L_g h(x) u \tag{2.25}$$

where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$.

We define the extended class K function $\kappa : \mathbb{R} \to \mathbb{R}$ which is continuous, strictly increasing and $\kappa(0) = 0$.

Definition 1 A continuous differentiable function $h(x) : X \to \mathbb{R}$ is a Zeroing Control Barrier function if there exists an extended class K function κ as in 2.26.

$$\sup_{u \in \mathbb{R}^m} \{L_f h(x) + L_g h(x) u + \kappa(h(x)) \geq 0\}, \tag{2.26}$$

We now define a set of control inputs $U(x)$ which satisfies (15) for all $x \in \zeta$ as in 2.27.

$$U(x) = \{u \in \mathbb{R}^m \mid L_f h(x) + L_g h(x) u + \kappa(h(x)) \geq 0\}, \tag{2.27}$$

We use the agent structure as given in 2.3 to form the Zeroing Control Barrier Function. The set of all state values which render ζ are considered safe. To guarantee that ζ is forward invariant we use the following theorem.

Theorem 1. Consider a set ζ that is a subset of R_n. This set is defined as the superlevel set of a continuously differentiable function h, which also functions as a control barrier function. Given this, it follows that any Lipschitz continuous controller implemented for the system ensures that the set ζ is secure.

2.4.2 Control Barrier Function

Dynamics of a UAV in control affine form are given in (2.28):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.28}$$

where A is a square matrix of dimension 4x4 and B is a matrix of dimension 4x2.
$h(\mathbf{x})$ is a valid CBF if it is differentiable and follows the conditions in (2.29).

$$\begin{cases} h(\mathbf{x}) > 0, \; \forall \, \mathbf{x} \in \zeta \\ h(\mathbf{x}) = 0, \; \forall \, \mathbf{x} \in \delta\zeta \end{cases} \tag{2.29}$$

where ζ is the set of all states of the UAV which lie in the safe region and $\delta\zeta$ are the states of the UAV on the boundary of the safe region.

If the UAV is initially located within the secure area ζ, the principle of forward invariance can be applied by verifying that $\dot{h}(\mathbf{x}) \geq 0$. This principle ensures that the UAV remains within the safe region if it commences within it. To enhance optimization for ideal trajectory tracking while also providing safety guarantees, this principle can be extended to an invariance condition where $\dot{h}(\mathbf{x}) \geq -\gamma h(\mathbf{x})$. This invariance condition induces the asymptotic convergence of $h(\mathbf{x})$ to 0. The condition for invariance is presented in equation (2.30).

$$\frac{\partial h(\mathbf{x})}{\partial x}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) + \gamma h^{z}(\mathbf{x}) \geq 0 \tag{2.30}$$

where $\gamma > 0$ is the relaxation coefficient and $z > 0$ is the exponential limit of convergence for the CBF. We define CBF to avoid point obstacles as h in (2.31).

$$h(\mathbf{x}) = \sqrt{2a_{max}(||\mathbf{p}|| - d_s)} + \frac{\mathbf{p}^T}{||\mathbf{p}||}\dot{\mathbf{p}} \tag{2.31}$$

The expression a_{max} represents the highest possible acceleration value of the UAV, whereas d_s is the secure distance that separates the obstacle from the UAV. Additionally, $\mathbf{p}$ denotes the vector from the obstacle's location to the UAV center, while $\dot{\mathbf{p}}$ represents the velocity of the UAV at a particular time instant k. Similarly, $h(\mathbf{x})$ can be defined as a discrete-time control barrier function (CBF). The final invariance condition can be found in equation (2.32).

$$\begin{aligned} \frac{a_{max}\,\dot{\mathbf{p}}^T\mathbf{p}}{\sqrt{2a_{max}(||\mathbf{p}||-d_s)}} - \left(\frac{\mathbf{p}^T}{||\mathbf{p}||}\dot{\mathbf{p}}\right)^2 + ||\dot{\mathbf{p}}||^2 + \mathbf{p}^T\mathbf{u} \\ + \gamma h^z(\mathbf{x})||\mathbf{p}|| \;\geq\; 0 \end{aligned} \tag{2.32}$$

In this chapter, we discussed the methods for trajectory tracking for aerial manipulators and UAVs. We also reviewed some existing control frameworks for aerial manipulators, identified their limitations and gave backgrounds on the necessary aerial robot dynamics and Barrier Lyapunov Function. In the next chapter, we propose a novel controller framework that addresses these limitations and provides trajectory tracking while ensuring safe operation of the system under unknown bounded disturbances.

Chapter 3

Predictive Barrier Lyapunov Function Based Control for Safe Trajectory Tracking of an Aerial Manipulator

This chapter proposes a novel controller framework that provides trajectory tracking for an Aerial Manipulator (AM) while ensuring the safe operation of the system under unknown bounded disturbances. The AM considered here is a 2-DOF (degrees-of-freedom) manipulator rigidly attached to a UAV. Our proposed controller structure follows the conventional inner loop PID control for attitude dynamics and an outer loop controller for tracking a reference trajectory. The outer loop control is based on the Model Predictive Control (MPC) with constraints derived using the Barrier Lyapunov Function (BLF) for the safe operation of the AM. BLF-based constraints are proposed for two objectives, viz. 1) To avoid the AM from colliding with static obstacles like a rectangular wall, and 2) To maintain the end effector of the manipulator within the desired workspace. The proposed BLF ensures that the above-mentioned objectives are satisfied even in the presence of unknown bounded disturbances. The capabilities of the proposed controller are demonstrated through high-fidelity non-linear simulations with parameters derived from a real laboratory scale AM. We compare the performance of our controller with other state-of-the-art MPC controllers for AM.

Work has been accepted for publication at: *European Control Conference (ECC) 2023*

3.1 Introduction

Aerial Manipulators (AMs) have gained much attention in recent years [17]. The Unmanned Aerial Vehicle (UAV) acts as a floating base for the manipulator enabling it to conduct active operations in the 3D space, see Fig. 1.3. This combination of a UAV and a manipulator provides the system with enough capabilities to perform a range of complex operations where human access is limited (e.g., in disaster-affected zones). The other industrial and commercial applications of AMs are in the maintenance of power grids, the inspection of bridges, and canopy sampling [17].

Such applications involve trajectory tracking maneuvers by the AM near static objects like bridges, trees, and buildings. It is unarguable that any system should be designed to be safe. In fact, safety has been a major hurdle in deploying such AM systems in these applications [36]. Three major issues are faced with said close maneuvers. Firstly, hovering close to such objects leads to ground, ceiling, and wall effects, causing immeasurable turbulent disturbances [36]. Secondly, the AM encounters disturbances in the form of forces and torques due to the highly coupled dynamics of the UAV, and manipulator [19]. These disturbances can lead to instability of the AM and cause a collision with the obstacles. E.g., external factors in the form of wind disturbances can lead to instability [37]. Thirdly, it is only sometimes possible to accurately model the obstacles around a trajectory due to their irregular shape, lack of visibility, or uncertainty associated with obstacle locations. This can occur when the AM maneuvers through a dark or uneven tunnel incapacitating it to determine a bound across the obstacles. Keeping this in mind, the AM must operate while keeping a safe distance from obstacles and maintaining stability. The AM movement can be bound in a desired workspace around the desired trajectory. This will prevent any possible collisions with obstacles.

3.2 Related Work

Adaptive controller[20] tackles torques due to the highly coupled dynamics of the AM by using an outer loop adaptive control over the proportional–derivative (PD) inner loop of the UAV. Though it provides computational efficiency, it is incapable of incorporating constraints to avoid obstacles. Model Predictive Control (MPC) [22] significantly reduces the abruptness in control inputs and tracks the desired trajectory while anticipating future dynamic interactions of the AM. PID and traditional adaptive controllers lack this predictive ability. MPC is applied to open a hinged door [30]. Considering the coupled dynamics of an AM and a hinged door, an MPC in the framework of a Linear Quadratic Regulator is designed.

Barrier Lyapunov Function (BLF) [1] is used as a tool to enforce the safety of non-linear dynamical systems. Barrier certificates [38] are established considering a safe region of operation defined as ζ. While guaranteeing forward invariance of ζ, safety is ensured. BLF-based MPC for a non-linear system described in [32] proposes a stabilizing controller to ensure avoidance of a set of states associated with the unsafe region for a chemical process. MPC combined with constraints using BLF [27] is used for distributive multi-UAV avoidance. An MPC scheme for safety planning [27] using BLF demonstrates the trade-off between the safety and performance of a UAV.

3.3 Contributions

The key contributions of this chapter are the following:

1. To the best of the author's knowledge, this is the first attempt to incorporate safe operation for AM maneuvers amidst unknown disturbances and boundary conditions.

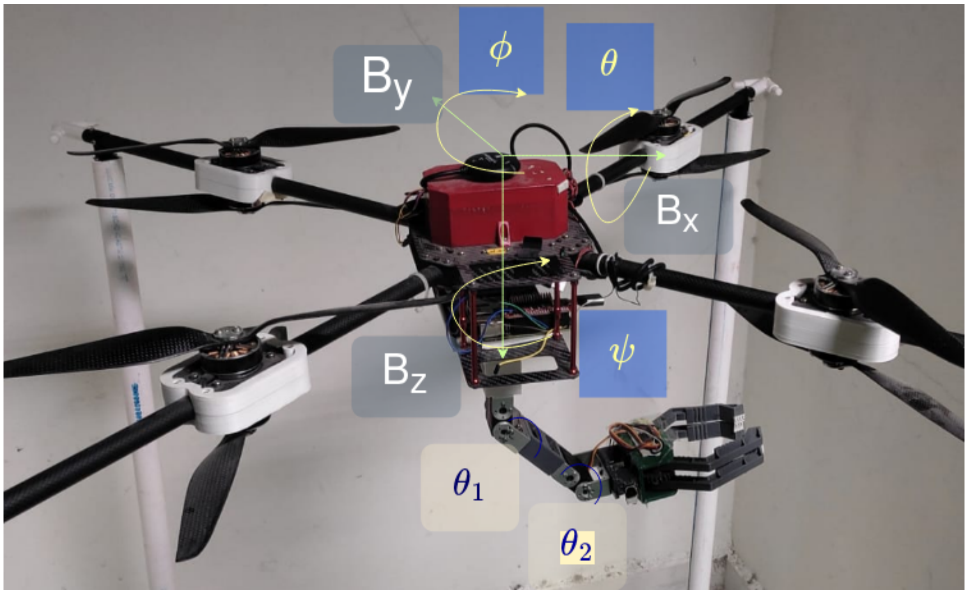

Figure 3.1: Aerial Manipulator in RRC, IIIT Hyderabad

2. We introduce BLF-based constraints over an MPC controller to include obstacle avoidance and achieve tangible performance gain in trajectory tracking for the end-effector of the AM in comparison to prior MPC controllers [22].

3. We exploit BLF to create a novel constraint for bounding the AM inside a defined boundary, contrary to it's collision avoidance utility.

4. A disturbance resistivity term is introduced in the BLF for guaranteeing safety under bounded random disturbances.

The chapter is structured as follows, Section II provides the dynamics model of the AM and BLF forward invariance constraint for collision avoidance. Section III provides the problem statement, while Section IV proposes a control architecture for the safe operation of AM. Section V discusses the simulation and benchmark results.

3.4 Problem Formulation

The primary aim of the chapter is to design a controller to follow the desired trajectory ($\mathbf{p}^d$) for the end-effector of the AM i.e. to minimize at any time t the trajectory error ($e_e(\mathbf{x})$)

$$\min_{\mathbf{u}} e_e(\mathbf{x}) = ||\mathbf{p}_e(\mathbf{x}) - \mathbf{p}^d|| \; \forall \, t > 0 \tag{3.1}$$

where $\mathbf{p}_e(\mathbf{x})$ is the position of the end-effector and $\mathbf{p}^d$ is the desired position of the end-effector at time

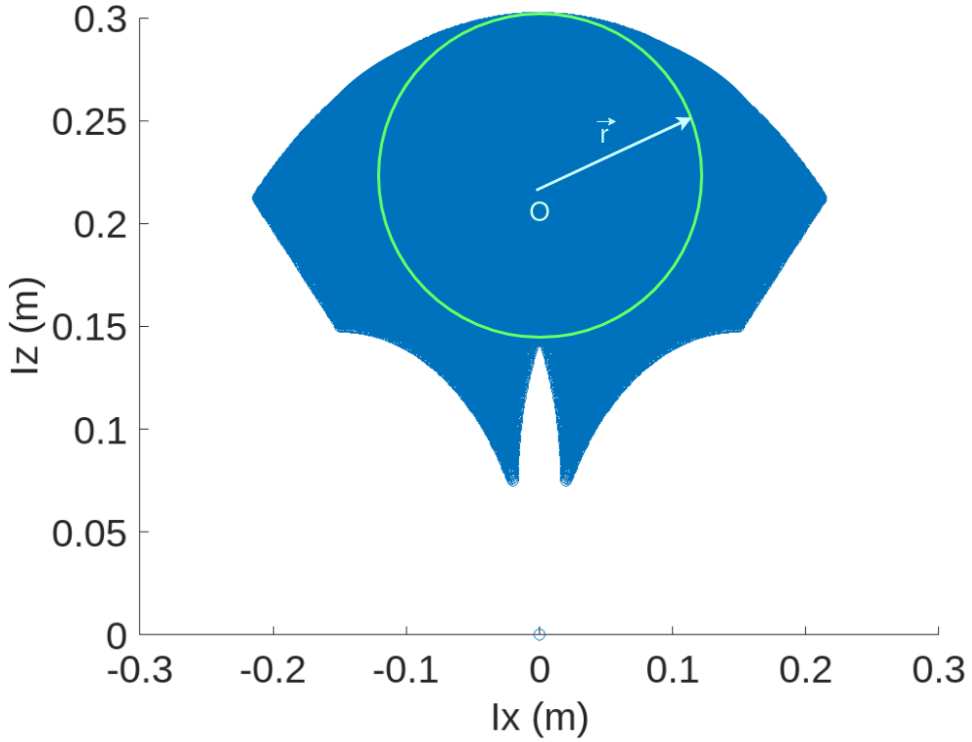

Figure 3.2: Inverse Workspace of a floating base manipulator wrt to a fixed end-effector point (0,0). This is a 2-D representation of a 3-D workspace. The 3-D workspace can be visualized by rotating 360°around line x = 0. The trajectory bound shown as a circle is a sphere $||p_r - O|| = r$ where p_r is a point on the sphere, O is the center of the sphere in 3D space and r is the radius of the sphere.

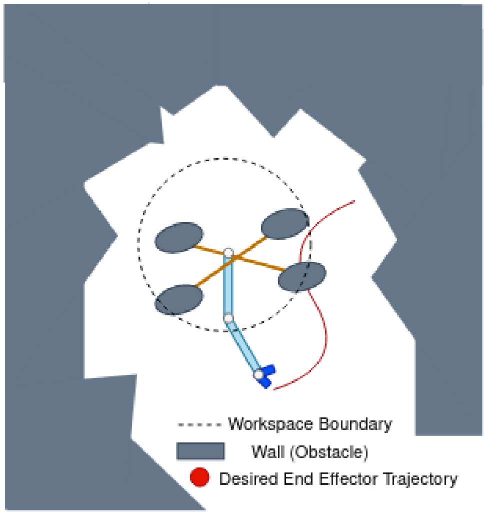

Figure 3.3: **Setup for AM simulation:** (a) Barrier avoidance: Shows close operation of an AM with rectangular walls on the three sides with a known position. (b) Free space workspace tracking: Shows bounded maneuver of the AM in the desired workspace within an allowed free space of the operation (here, the obstacle position information is not available).

t. This trajectory is defined for the end-effector, while the trajectory of the UAV is left to be determined by the controller, such as to reduce $\mathbf{e}_e(\mathbf{x})$ given in (3.1).

There are multiple ways to reach a desired point on the trajectory by placing the UAV in the inverse workspace of the floating base manipulator. This inverse workspace, Fig. 3.2, is defined by taking the inverse kinematics of the floating base manipulator considering it's end-effector to be fixed at the desired trajectory point. E.g., Fig. 3.2 shows the inverse workspace of a 2R manipulator in which manipulator joint angles are constrained.

Two cases are considered here, **Case I**, the location of the obstacle is known, and **Case II**, the allowed free space for operation is given without specifying the location of the obstacle. For **Case I** shown in Fig. 3.3(a), the end-effector of the AM has to follow the desired trajectory close to static obstacles. The condition for safety is given in (3.2).

$$\begin{aligned} ||\vec{\mathbf{s}}_j(\mathbf{x}_i)|| \geq s_j^{min} \,\forall\, j \in \{1, ..., C\} \\ \vec{\mathbf{s}}_j \times [a, b, c]^T = 0 \end{aligned} \tag{3.2}$$

where $\vec{\mathbf{s}}_j$ is the perpendicular distance vector of the critical point j from the wall and s_j^{min} is the minimum safety distance from the wall. The number of critical points is denoted by C. In Fig. 3.3(a), critical points on the AM are (A) the base of the manipulator with safety radius s_1, (B) joint 2 of the manipulator with safety radius s_2 and (C) end-effector of the manipulator with safety radius s_3. Equation of the wall is given by $ax + by + cz + d = 0$ where a,b,c,d are parameters of the plane and x, y, z are points on the plane.

For **Case II**, Fig. 3.3(b), a safe free space of operation around the trajectory is provided, and we provide a guarantee that the AM will stay in that safe bounded region. We choose a spherical differentiable part Fig. 3.2 of the non-differentiable inverse workspace with radius r. The condition for bounding in the desired workspace is given in (3.4).

$$||\vec{\mathbf{d}}(\mathbf{x})|| \leq r, \;\; \vec{\mathbf{d}}(\mathbf{x}_i) = \mathbf{p}_I(\mathbf{x}) - (\mathbf{p}^d - d_{iw}) \tag{3.3}$$

where r is the radius of the desired workspace and d_{iw} is the deviation of the workspace center from the desired trajectory point. As the bound for the allowable free space is within the inverse workspace, the end-effector would be able to reach the desired point in at least one orientation of the AM.

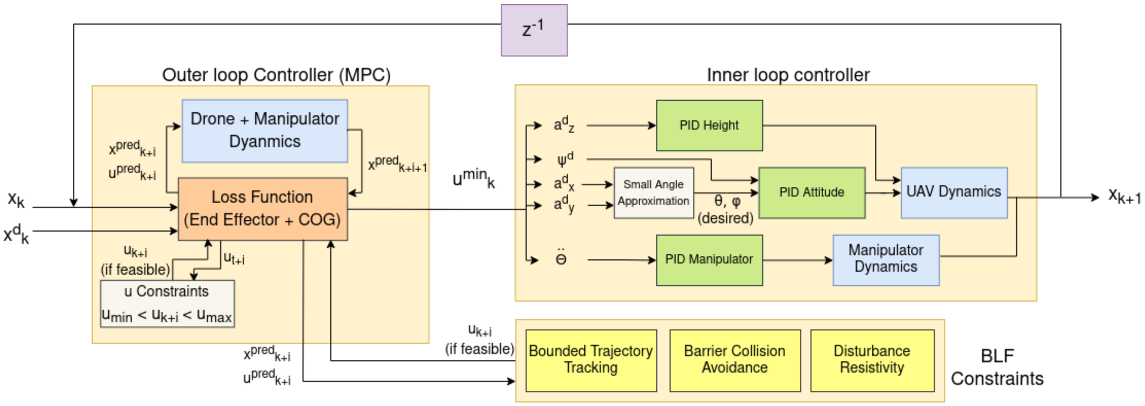

Figure 3.4: **Control Architecture of the AM.** Outer loop controller denotes the MPC optimization problem while the inner loop controller denotes the on-board PID control of the AM. Constraints on the optimizer are denoted by input constraints ($\mathbf{u}$) an Constraints.

3.5 PROPOSED CONTROLLER

The architecture of the proposed controller is shown in Fig. 3.4. It consists of an outer loop MPC with BLF constraints. The inner loop is driven by a conventional PID controller. The outer loop provides the desired acceleration and yaw angle references to the inner loop controller.

3.5.1 MPC (Outer loop control)

The state vector of the AM is defined as $\mathbf{x} = [\mathbf{p}_I, \dot{\mathbf{p}}_I, \psi, \dot{\psi}, \Theta, \dot{\Theta}]$, where $\mathbf{p}_I$ and $\dot{\mathbf{p}}_I$ is the position and velocity of the center of mass of the UAV in inertial frame respectively, ψ and $\dot{\psi}$ is the yaw and yaw rate, and $\Theta = [\theta_1, \theta_2]$ and $\dot{\Theta} = [\dot{\theta}_1, \dot{\theta}_2]$ is the joint angle and angular rate of the manipulator's joints. The control input for MPC is $\mathbf{u} = [\mathbf{a}_d, \ddot{\psi}, \ddot{\Theta}_d]$ where $\mathbf{a}_d = [a_x, a_y, a_z]$ is the acceleration of the

UAV center in the inertial frame, $\ddot{\psi}$ is the yaw angular acceleration and $\ddot{\Theta}$ is the angular acceleration of manipulator's joints. The state space model used within the MPC is formulated as given in (3.4).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.4}$$

$$\mathbf{A} = \begin{bmatrix} 0_{3x3} & I_{3x3} & 0_{3x6} \\ 0_{3x3} & 0_{3x3} & 0_{3x6} \\ 0_{1x7} & 1 & 0_{1x4} \\ 0_{1x3} & 0_{1x3} & 0_{1x6} \\ 0_{2x3} & 0_{2x7} & I_{2x2} \\ 0_{2x3} & 0_{2x3} & 0_{2x6} \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0_{3x3} & 0 & 0_{3x2} \\ I_{3x3} & 0 & 0_{3x2} \\ 0_{1x3} & 0 & 0_{1x2} \\ 0_{1x3} & 1 & 0_{1x2} \\ 0_{1x3} & 0 & 0_{1x2} \\ 0_{1x3} & 0 & I_{1x2} \end{bmatrix} \tag{3.5}$$

In (3.5), I is a diagonal matrix such that, $I(i,j) = 1, \forall\, i = j$ and $I(i,j) = 0, \forall\, i \neq j$ and 0 is zero matrix such that, $0(i,j) = 0, \forall\, i,j$ where $i,j \in \mathbb{N}$.

For Aerial Manipulator dynamics with state variables as $\mathbf{x}$ and control variables as $\mathbf{u}$, the optimal control problems at every time instant t_k where $t_k = kT$ and T is a time step is given in (3.6).

$$\min_{\mathbf{u}} l(\mathbf{x_k}, \mathbf{u_k}, t_k) \tag{3.6a}$$

$$\text{s.t. } \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{3.6b}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{3.6c}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max} \tag{3.6d}$$

where $\mathbf{u} = [\mathbf{u}_k{}^\top, \mathbf{u}_{k+1}{}^\top, ..., \mathbf{u}_{k+N-1}{}^\top]^\top$ denotes the vector of control variables and f is the dynamic model of the system. $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ represent the bounds on $\mathbf{u}$ and $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ represent the bounds on $\mathbf{x}$. $\mathbf{x}_k$ denotes the state of the system at k^{th} time step, similarly $\mathbf{u}_k$ denotes the control input of the system at k^{th} time step. The cost l is the summation of multiple cost functions $l = \sum_{i=1}^{N_l} l_i$ explained in the section *Weighing strategy*.

3.5.1.1 Weighing strategy

The N_l costs are chosen to follow the trajectory and enhance stability. These are defined in (3.7), (3.8) and (3.9).

3.5.1.1.1 Tracking error for the End-Effector The primary task is the tracking of the end effector over a given trajectory. This is achieved by penalizing the difference between the current and desired position as given in (3.7)

$$l_1 = \sum_{i=0}^{n-1} (||\mathbf{e}_e(\mathbf{x}_{k+i})||^2_{W_1}) + ||\mathbf{e}_e(\mathbf{x}_{k+N})||^2_{W_{s_1}} \tag{3.7}$$

where $\mathbf{e}_e(\mathbf{x}_{k+i}) = \ \mathbf{p}_e(\mathbf{x}_{k+i}) - \mathbf{p}^d_{k+i}$.

is the end-effector tracking error calculated at each time step of the prediction horizon. To decrease error from the desired trajectory, we penalize higher velocities of the manipulator end-effector as given in (3.8).

$$l_2 = \sum_{i=0}^{n-1}(||\mathbf{v}_e(\mathbf{x}_{k+i})||^2_{W_2}) + ||\mathbf{v}_e(\mathbf{x}_{k+N})||^2_{W_{s_2}} \tag{3.8}$$

where $\mathbf{v}_e$ is the velocity of the End-Effector of the Manipulator.

3.5.1.1.2 COG Allignment Error As the manipulator moves in the $\mathbf{B_x B_z}$ plane of the UAV (refer to Fig. 4.1) while the UAV changes its attitude (θ, ϕ, ψ), the Center of Gravity of the system moves in the $\mathbf{B_x B_y}$ direction of the UAV due to which undesirable torques appear which destabilize the UAV. The following cost given in (3.9) is introduced considering this factor.

$$l_3 = \sum_{i=0}^{n-1}(||\mathbf{p}_{G_{XY}}(\mathbf{x}_{k+i})||^2_{W_3}) + ||\mathbf{p}_{G_{XY}}(\mathbf{x}_{k+N})||^2_{W_{s_3}} \tag{3.9}$$

where $\mathbf{p}_{G_{XY}}$ is the Center of Gravity of the Manipulator in the $\mathbf{B_x B_y}$ plane.

MPC to reach desired points using this approach was used in [24], and we would call this technique **Naive MPC**.

3.5.2 PID (Inner Loop Control)

From the control input ($\mathbf{u}_k^{min}$) obtained from the outer loop controller, the desired values of θ, ϕ are estimated using small angle approximation (refer to Fig. 3.4) and along with ψ_d are tracked by the PID controller namely PID Attitude. Similarly, PID Height regulates the height, and PID Manipulator angle controls the joint angles θ_1 and θ_2 [20].

3.5.3 Safe operation near known barriers

For **Case I**, the priority for safety is the highest. The AM has to avoid collisions for both the UAV and the manipulator simultaneously while tracking the desired trajectory. The critical points are chosen such that if we can guarantee collision avoidance for these points, the entire system can safely perform desired maneuvers with collision. BLF used for collision avoidance along the radial direction of the Wall is given in (3.10).

$$h_1(\mathbf{x}_k) = \sqrt{\frac{2\vec{\mathbf{s}}_j(\mathbf{x}_{k+i})^T\alpha_{max}}{||\vec{\mathbf{s}}_j(\mathbf{x}_{k+i})||}(||\vec{\mathbf{s}}_j(\mathbf{x}_{k+i})|| - s_j^{min})} + \frac{\vec{\mathbf{s}}_j(\mathbf{x}_{k+i})^T}{||\vec{\mathbf{s}}_j(\mathbf{x}_{k+i})||}\vec{\mathbf{v}}_{k+i} \tag{3.10}$$

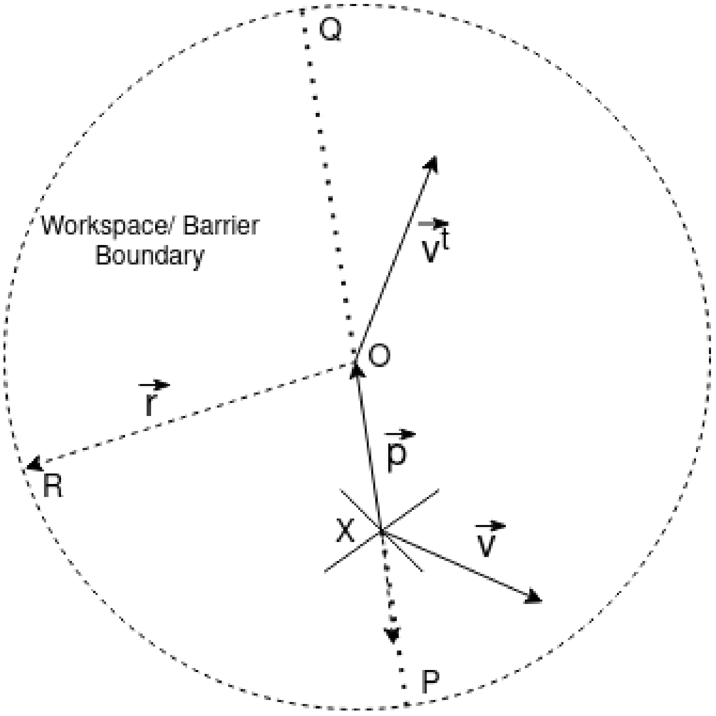

Figure 3.5: Desired workspace for the UAV (refer to the circle in Fig. 3.2). X denotes the Center of the UAV, $\vec{p}$ denotes vector from X to O, $\vec{v}$ denotes instantaneous velocity of X and $\vec{v}_k$ denotes instantaneous velocity of desired trajectory of the end-effector

3.5.4 Bounded Trajectory Tracking

For **Case II**, one of our major contributions in this section is the usage of BLF for the bounding of the UAV in the desired workspace shown in Fig. 3.5. We exploit the property that a particle can exit a sphere only through its motion in the radially outwards direction. Hence, the UAV can only escape the boundary if it is provided with a high velocity in the radial direction (along $\vec{\mathbf{p}}$ or $-\vec{\mathbf{p}}$). Due to this restricted movement in the radial directions on both sides, the UAV does not leave the workspace at that particular instant k. BLF (h_{2_a}) along $\vec{\mathbf{p}}$ is given in (3.11) and a similar BLF (h_{2_b}) along $-\vec{\mathbf{p}}$ can be found by replacing $\vec{\mathbf{p}}$ with $-\vec{\mathbf{p}}$.

$$h_{2_a}(\mathbf{x}_k) = \sqrt{\frac{2(\vec{\mathbf{p}}_{k+i})^T \alpha_{max}}{||\vec{\mathbf{p}}_{k+i}||}(||\vec{\mathbf{p}}_{k+i}|| - r)} + \frac{\vec{\mathbf{p}}_{k+i}^T}{||\vec{\mathbf{p}}_{k+i}||}(\vec{\mathbf{v}}_{k+i} - \vec{\mathbf{v}}_{k+i}^t) \tag{3.11}$$

h_{2_b} can be formulated as in 3.14.

$$h_{2_b}(\mathbf{x}_k) = \sqrt{\frac{2(-\vec{\mathbf{p}}_{k+i})^T \alpha_{max}}{||\vec{\mathbf{p}}_{k+i}||}(||\vec{\mathbf{p}}_{k+i}|| + r)} + \frac{(-\vec{\mathbf{p}}_{k+i})^T}{||\vec{\mathbf{p}}_{k+i}||}(\vec{\mathbf{v}}_{k+i} - \vec{\mathbf{v}}_{k+i}^t) \tag{3.12}$$

3.5.5 BLF with Disturbance Rejection

Another contribution of this chapter involves handling unseen disturbances, such as constant wind or impulses acting at random intervals in the framework of BLF. The condition in (2.32) need not be satisfied for the given $h(\mathbf{x})$ in the presence of unmeasured disturbances. If the AM encounters a sudden disturbance, it might result in a collision with obstacles. The condition $\dot{h}(\mathbf{x}) \geq -\gamma h^z(\mathbf{x})$ is tightened while maintaining relaxation for smooth operation as given in (3.15) with $\lambda > 0$.

$$\dot{h}_i(\mathbf{x}) + \gamma(h_i^z(\mathbf{x}) - \lambda) \geq 0 \tag{3.13}$$

Theorem 1: For the state-space model of the dynamical system with a bounded disturbance $\mathbf{d}$ given in (3.14),

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} + \mathbf{d},\ \forall\, |\mathbf{d}| \leq d_m \tag{3.14}$$

the forward invariance condition for $h(\mathbf{x})$ is ensured by the condition given in (3.15).

$$\dot{h}_i(\mathbf{x}) + \gamma(h_i^z(\mathbf{x}) - \lambda) \geq 0 \tag{3.15}$$

with $\lambda \geq |\frac{1}{\gamma}\frac{\partial h}{\partial \mathbf{x}} d_m|$.

Proof:

Expanding (3.15) and using (3.16), the following result is obtained.

$$\frac{\partial h(\mathbf{x})}{\partial x}(f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \gamma h^z(\mathbf{x}) + \frac{\partial h(\mathbf{x})}{\partial x}\mathbf{d} - \gamma\lambda \geq 0 \tag{3.16}$$

The term $\frac{\partial h(\mathbf{x})}{\partial x}(f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \gamma h^z(\mathbf{x})$ is greater than 0. Hence, the condition $\lambda \geq |\frac{1}{\gamma}\frac{\partial h}{\partial \mathbf{x}} d_m|$ yields the desired result.

$$\frac{\partial h(\mathbf{x})}{\partial x} d - \gamma\lambda \geq 0 \tag{3.17}$$

The BLF functions h_1, h_{2_a} and h_{2_b} are put in (3.15) and are added to the MPC optimizer as constraints. The resultant controller is termed as **MPC-BLF**. To the best of the author's knowledge, there have been no attempts to handle disturbance forces or torques for an AM using a BLF.

3.5.6 BLF for bounding box (Variation-2)

We formulate a safety constraint in the spherical region of $r = 0.075m$ by ensuring that the system stops before it crosses the region boundary. The region is defined with respect to the desired end-effector position at time t which is the the position $p_t \in \mathbb{R}^3$ from the trajectory at time t. The centre of the region is defined as :

$$p_o = p_t + \begin{bmatrix} 0 \\ 0 \\ 0.225 \end{bmatrix} \tag{3.18}$$

As the centre of the region is moving with the same velocity $v_t \in \mathbb{R}^3$ and acceleration $a_t \in \mathbb{R}^3$ as that of trajectory at time t, we take into account the relative velocity between the drone and trajectory as this helps in ensuring that we remain within the dynamically moving workspace. We define the relative position and velocity between the drone and centre of the region as $\Delta p = p_d - p_o$ and $\Delta v = v_d - v_t$.

To strictly enforce the safety constraint, we consider that $||\Delta v||$ is acting along the direction of the vector Δp as $(r - ||\Delta p||)$ is the shortest distance from the boundary region. We use the maximum deceleration value α to check whether we will be able to stop the system before it crosses the boundary. The safety constraint can be formulated as follows :

$$\begin{aligned} ||\Delta p|| + \frac{||\Delta v||^2}{2\alpha} \leq r \\ \frac{||\Delta v||^2}{2\alpha} \leq r - ||\Delta p|| \end{aligned} \tag{3.19}$$

The above inequality has the following solutions :

$$-\sqrt{2\alpha(r - ||\Delta p||)} \leq ||\Delta v|| \leq \sqrt{2\alpha(r - ||\Delta p||)} \tag{3.20}$$

Since, norm of the vector i.e $||\Delta v|| \geq 0$, the final solution is :

$$0 \leq ||\Delta v|| \leq \sqrt{2\alpha(r - ||\Delta p||)} \tag{3.21}$$

The safe set ζ for the trajectory tracking Aerial Manipulator can be defined as :

$$\begin{aligned} \zeta = \{(p_d, v_d) \in \mathbb{R}^6 \mid h(x) \geq 0\} \\ h(p_d, v_d) = \sqrt{2\alpha(r - ||\Delta p||)} - ||\Delta v|| \end{aligned} \tag{3.22}$$

where $h(p_d, v_d)$ is the level set function of the set ζ which is used to define the forward invariance of the safe set i.e. the ZCBF candidate. For proving the forward invariance of the safe set ζ, the constraint in (15) needs to be satisfied. Writing equation of the form (15) from eq. (32), we get :

$$\gamma h^3 - \frac{\alpha \Delta v^T \Delta p}{||\Delta p||\sqrt{2\alpha(r - ||\Delta p||)}} - \frac{\Delta u^T \Delta v}{||\Delta v||} \geq 0 \tag{3.23}$$

Here $\Delta u = u - u_t$ is the relative acceleration between the Drone and the trajectory at any time t. The constraint can be expressed in Linear form $Au \leq b$ as :

$$A = \frac{\Delta v^T}{||\Delta v||} \tag{3.24}$$

$$b = \gamma h^3 - \frac{\alpha \Delta v^T \Delta p}{||\Delta p||\sqrt{2\alpha(r - ||\Delta p||)}} + \frac{u_t^T \Delta v}{||\Delta v||} \tag{3.25}$$

The Safety Barrier Certificate defines the admissible control space U for the system as :

$$U(x) = \{u \in \mathbb{R}^3 \mid Au \leq b\} \tag{3.26}$$

This version of BLF to bind the AM inside a safe region did not work as expected, hence was scarped.

3.6 Simulation Results

The algorithm is implemented on Python 3 on an Intel® Core™ i7-8550U CPU PC running at 1.80 GHz. We use the 'SLSQP' method from scipy as the non-linear optimizer for MPC. The AM used for simulation is a mathematical replica of a laboratory scale AM shown in Fig. 4.1 with the following specifications given in Table 3.1.

Parameter	Value
Mass	3.5 kg
Arm length	0.3 m
Propeller Diameter	0.33 m
Moment of Inertia - UAV	$I_x = 0.3\ kg\ m^2$, $I_y = 0.3\ kg\ m^2$, $I_z = 0.6\ kg\ m^2$
Manipulator length	$link1 = 0.15\ m, link2 = 0.15\ m$
Moment of Inertia - Manipulator	$I_1 = 4.256 \times 10^{-5}\ kg\ m^2$, $I_2 = 8.321 \times 10^{-5}\ kg\ m^2$
Moment of Inertia - Manipulator	$I_1 = 4.256 \times 10^{-5}\ kg\ m^2$, $I_2 = 8.321 \times 10^{-5}\ kg\ m^2$
UAV attitude constraints	$\lvert\theta\rvert \leq \pi/10\ rad$, $\lvert\phi\rvert \leq \pi/10\ rad$
Manipulator joint angle constraints	$\lvert\theta_1\rvert \leq \pi/3\ rad$, $\lvert\theta_1 + \theta_2\rvert \leq \pi/2\ rad$

Table 3.1: Specifications of the Aerial Manipulator

We perform simulations separately for two cases.

Case I: Avoiding Wall on three sides of the AM (The position of the obstacle is known).

Case II: Free space for the operation of AM is given (The position of obstacles is unknown, a desired spherical workspace for the UAV is created.)

For both of the simulations, the task is to follow a desired trajectory by the end-effector. A uniformly random disturbance with an amplitude of $d_m = 0.8\, m/s^2$ is used to evaluate the performance in the presence of external disturbances.

3.6.1 Modified Naive MPC for comparison

The Naive MPC is modified as given below to compare the performance with the proposed controller.

3.6.1.1 Hard Constraint (MPC-HC)

For **Case I**, a hard constraint on the critical points to avoid collision with the Wall is enforced as in (3.2) $\forall\ i\ \in [0, n]$. For **Case II**, a hard constraint on the relative position of UAV center (Fig. 3.5) as in (3.4)

$\forall\ i\ \in [0,n]$ is added. Similar to **Case I**, this is the intuitive constraint for bounding. These additional constraints are put on the MPC optimizer in (3.6).

$$\mathbf{r} - ||\vec{\mathbf{p}}_{k+i}|| \geq 0\ \forall i \in [0,n] \tag{3.27}$$

3.6.1.2 Soft Constraint (MPC-SC)

For **Case I**, a cost function for safe operation near walls is given in (3.28). This cost increases as the Wall is approached, penalizing the critical points going near the Wall.

$$l_4 = \sum_{i=0}^{n-1} \frac{1}{(||\vec{\mathbf{p}}_j(\mathbf{x}_{k+i})|| - s_j)^2_{W_4}} + \frac{1}{(||\vec{\mathbf{p}}_j(\mathbf{x}_{k+n})|| - s_j)^2_{W_{s_4}}} \tag{3.28}$$

For **Case II**, a cost function that penalizes the UAV movement outside the workspace is given in (3.29) and is added to the MPC cost.

$$l_5 = \sum_{i=0}^{n-1} ||\mathbf{p}(\mathbf{x}_{k+i})||^2_{W_5} + ||\mathbf{p}(\mathbf{x}_{k+n})||^2_{W_{s_5}}\ \forall\ |\mathbf{p}(\mathbf{x}_{k+i})| \geq r \tag{3.29}$$

Parameters and weights for the different MPC versions are given in Table 3.2.

Parameter	**Value**
MPC Weights	$w_1 = 10 \times \mathbf{I}_{3\times3}$, $w_{s_1} = 50 \times \mathbf{I}_{3\times3}$, $w_2 = 2 \times \mathbf{I}_{3\times3}$, $w_{s_2} = 10 \times \mathbf{I}_{3\times3}$
	$w_3 = \mathbf{I}_{3\times3}$, $w_{s_3} = 5 \times \mathbf{I}_{3\times3}$, $w_4 = 5 \times \mathbf{I}_{3\times3}$, $w_{s_4} = 20 \times \mathbf{I}_{3\times3}$
	$w_5 = 5 \times \mathbf{I}_{3\times3}$, $w_{s_5} = 20 \times \mathbf{I}_{3\times3}$
s_j^{min}	0.1 m
u_k Initialization	$\mathbf{0}_{1\times6n}$
d_{iw}	0.225 m
γ	3
Sampling time (t_s)	0.1 s
Total time (t)	120 s
Max disturbance amplitude	$d_m = 0.8\,m/s^2$

Table 3.2: Weights and Parameters for MPC

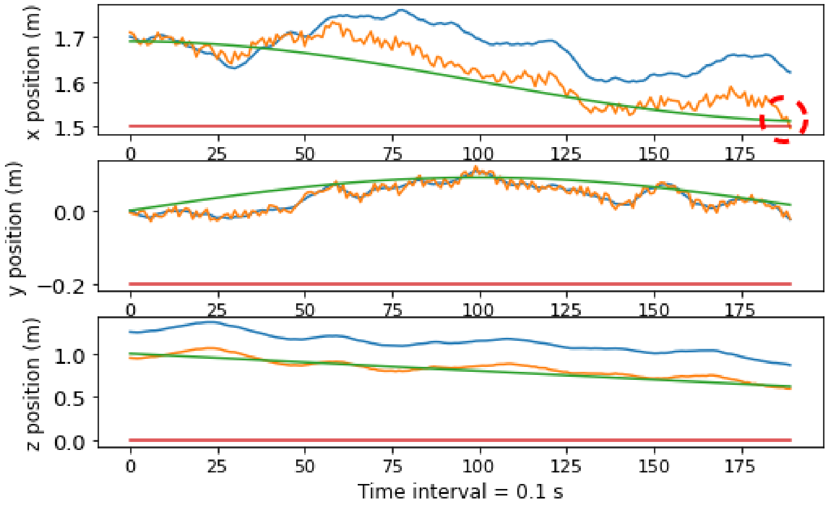

Figure 3.6: **AM position for case I with disturbances (Naive MPC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

3.6.2 Metrics for performance comparison

The performance of the controller is evaluated by the following metrics for N time steps ($N = t_f/t_s$). Here, t_f is the final time, and t_s is the sampling time in seconds.

- Maneuver completion without collision or escaping the desired workspace
- Manipulator end-effector root mean square error from the desired trajectory, TE = $\sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(\mathbf{e}_e(\mathbf{x}_k))^2}$
- Control effort, $c_e = \sum_{k=0}^{N-1} ||\mathbf{u}_k||^2$
- Control Smoothness, $c_s = \sum_{k=0}^{N-1} |\Delta \mathbf{u}_k|$

3.6.3 Performance comparison for Case II

Selection of the prediction horizon (n) and the parameter λ determines the computational load and disturbance rejection capabilities, respectively. An analysis was conducted by varying n and λ and the results are given in Table 3.3. We chose n = 5 and $\lambda = 5$ as it shows low TE while maintaining the workspace bound. These values are used for all subsequent simulations. $n = 10$ does not show significant improvement (¡5%) in TE compared to $n = 5$ and is rejected because of a very high computational time compared to $n = 5$ (¿600%). $\lambda = 5$ shows high disturbance resistivity given its low TE and low c_s. T_C denotes the computational time for one step.

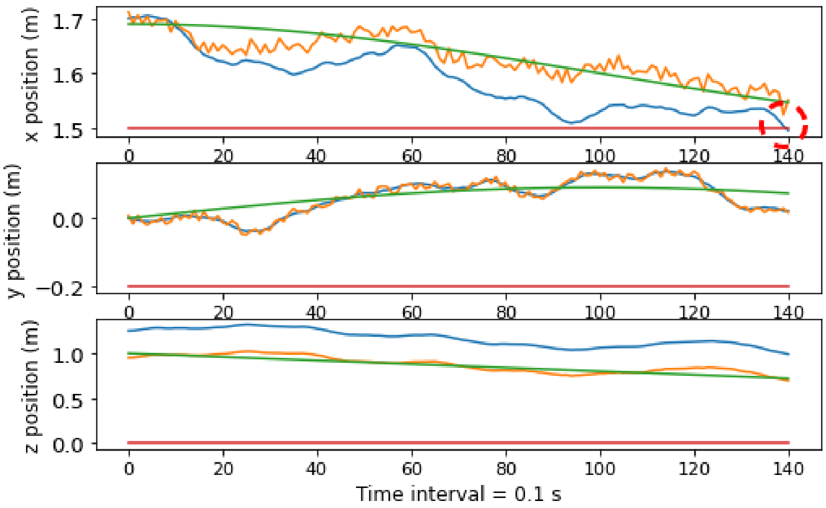

Figure 3.7: **AM position for case I with disturbances (MPC - HC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

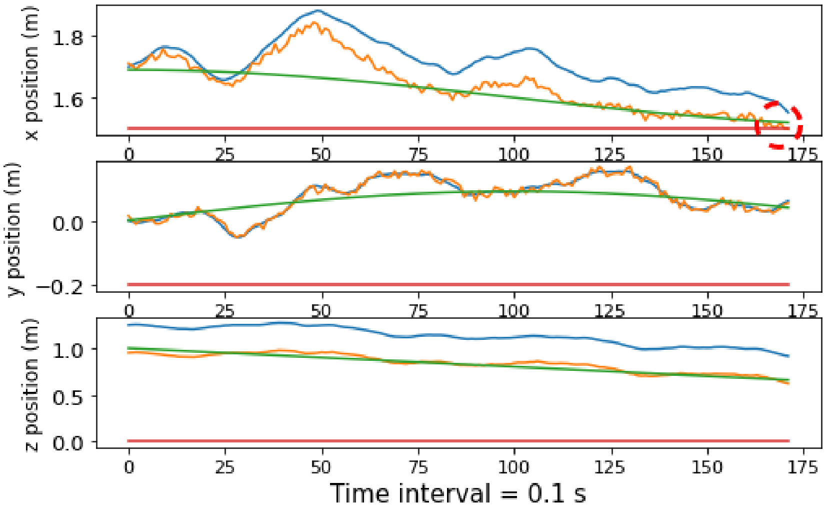

Figure 3.8: **AM position for case I with disturbances (MPC-SC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

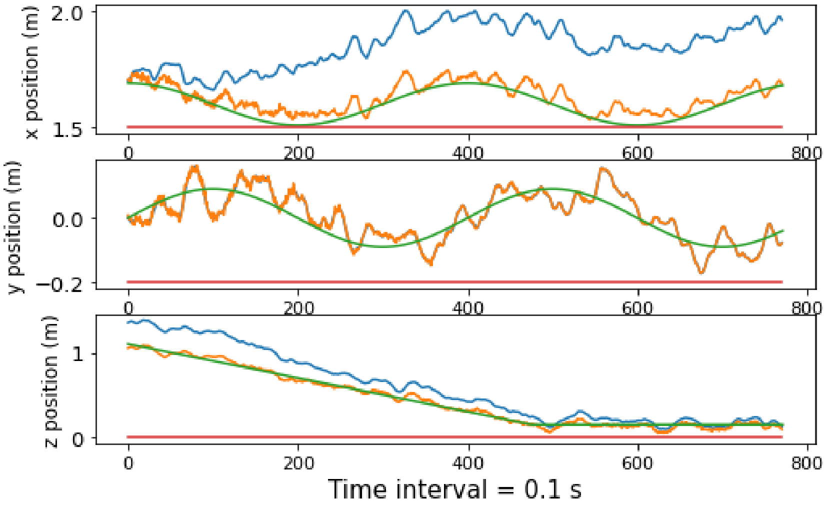

Figure 3.9: **AM position for case I with disturbances (MPC-BLF):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

Parameter	n = 1		n = 5		n = 10	
	$\lambda = 1$	$\lambda = 5$	$\lambda = 1$	$\lambda = 5$	$\lambda = 1$	$\lambda = 5$
T_C (s)	0.0313	0.0309	0.1712	0.1603	1.370	1.232
TE (m)	0.1943	0.1839	0.0738	0.0658	0.0701	0.0623
c_s	0.1593	0.1603	0.0746	0.0772	0.0716	0.0694
c_e	0.1432	0.1248	0.0801	0.0788	0.0939	0.0942

Table 3.3: Ablation showing the effect of parameters on the proposed MPC-BLF method (units mentioned in brackets)

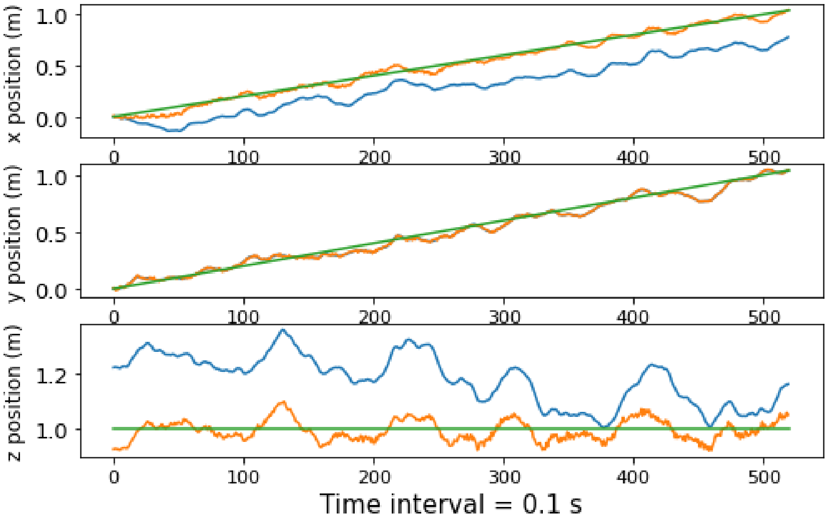

Figure 3.10: **AM position for case I with disturbances (Naive MPC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

The results given in Table 3.4 show that none among the Naive MPC, MPC-HC, and MPC-SC is able to restrict the UAV in the desired workspace, either in the presence or absence of external disturbances. MPC-HC has no penalization for the velocity of the UAV; hence when a large control input is provided, it exits the bounds. MPC-SC has to trade-off between TE and bounding cost, hence compromising on one of the factors. The proposed MPC-BLF is able to restrict the UAV within the desired workspace with the lowest control effort, highest control smoothness, and lowest tracking error. The inference is more evident in the presence of external disturbances. The trajectory of the AM is shown in Fig. 3.10 for all four methods in the presence of external disturbances. The proposed MPC-BLF method is the only one to confine the UAV within the safe boundary (Fig. 3.17). Though MPC-SC is able to avoid the Wall during its maneuver, it is not possible for the algorithm to give higher priority to wall avoidance. Hence, it tries to converge for both tasks (Wall avoidance and End-effector tracking) leading to high TE. Similarly, if weights for bounding the UAV in the desired workspace are increased, the TE shoots up and vice versa. As MPC-BLF is able to bind the UAV in the desired workspace, it accounts for less TE compared to MPC-HC or MPC-SC as the end effector reaches the desired trajectory point with lesser UAV movement reducing the control effort significantly.

For Case I, Fig. 3.10 shows collision with the Wall for Naive MPC, MPC-HC and MPC-SC.

3.6.4 Performance comparison for Case I

As shown in Fig. 3.13, **Case I** shows similar behaviour as **Case II**. Fig. 3.13 shows trajectory tracking in only I_x, a similar result is obtained in I_y, I_z. As shown in Table 3.4, MPC-SC shows a very

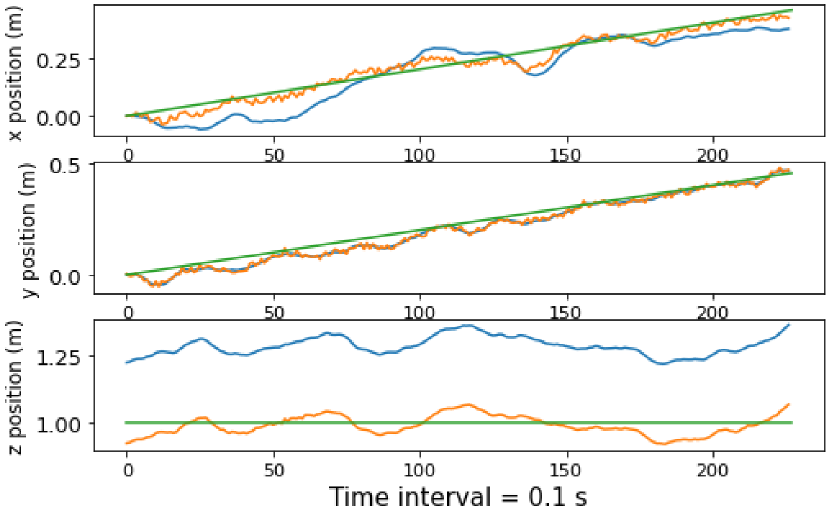

Figure 3.11: **AM position for case I with disturbances (MPC - HC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

high TE compared to MPC-BLF even in the absence of disturbances. MPC-BLF is able to avoid the walls contrary to any other method in the presence of disturbances.

As shown in Table 3.4, AM with Naive MPC collides for Case I as the algorithm has no constraint for avoidance. MPC-HC is not able to work for safe operation as there is no penalization for high velocities. If the AM is far from the barrier, but the velocity in the radial direction is high, it will still collide.

3.7 Chapter Conclusion

This chapter presented a BLF-based Model predictive controller with a primary objective of safe operation in the proximity of static objects. Our approach shows how BLF, formulated for barrier avoidance and free space tracking objectives, shows robust behavior for the safe operation of an Aerial Manipulator in the presence of external disturbances. A state-of-the-art MPC-based method is modified using two types of constraints and compared with the proposed method, which shows significant improvement in two cases i.e with and without external disturbances. This is validated in simulation using parameters from a real laboratory scale AM.

This chapter focuses on flying an aerial manipulator near static obstacles and handles external wind disturbances. We extend this work in the next chapter for a different application of a UAV flying inside a tunnel. The new problem has nonlinear torques while flying inside the tunnel which creates a different set of difficulties to be tackled by the proposed controller.

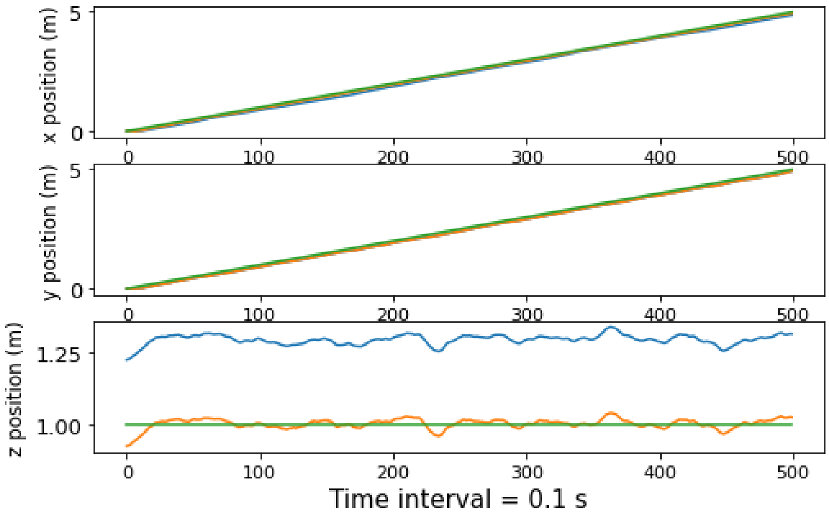

Figure 3.12: **AM position for case I with disturbances (MPC-SC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

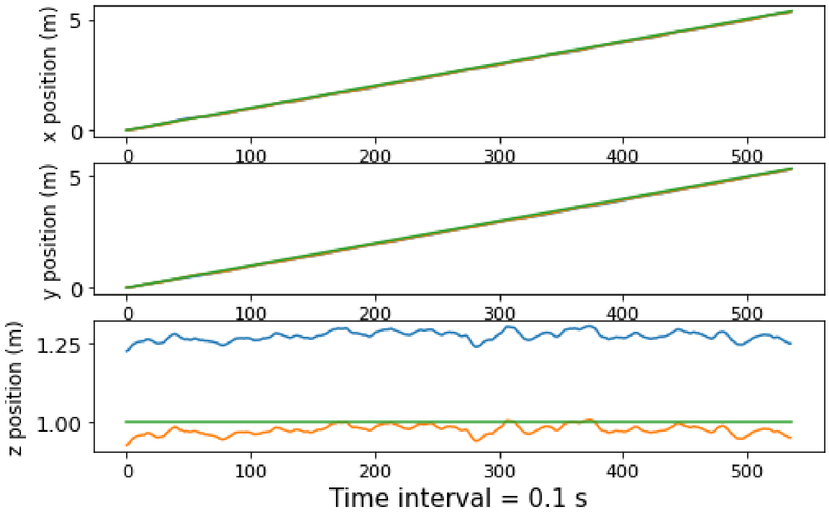

Figure 3.13: **AM position for case I with disturbances (MPC-BLF):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

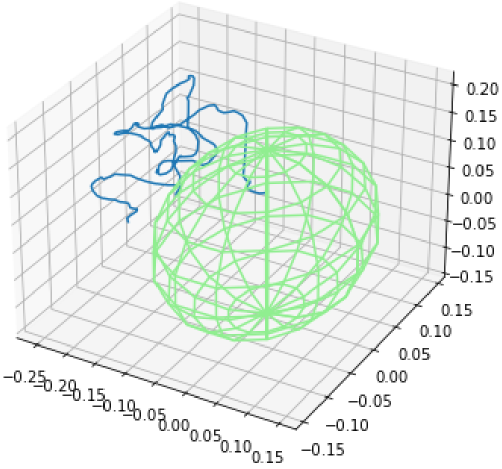

Figure 3.14: **AM position for case I with disturbances (Naive MPC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

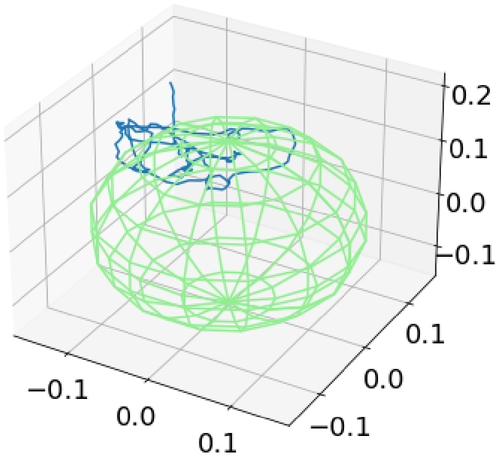

Figure 3.15: **AM position for case I with disturbances (MPC - HC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

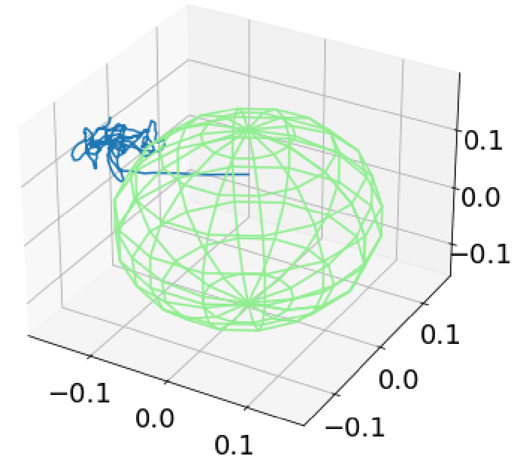

Figure 3.16: **AM position for case I with disturbances (MPC-SC):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

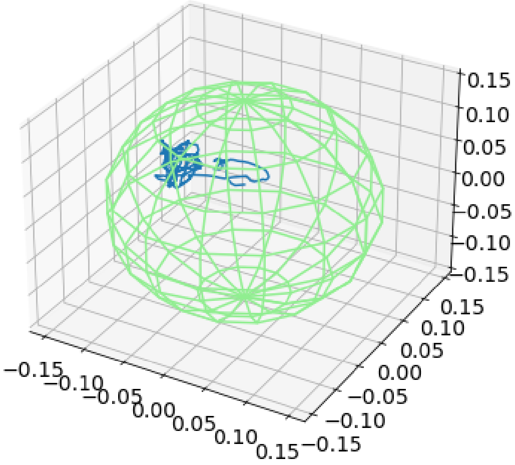

Figure 3.17: **AM position for case I with disturbances (MPC-BLF):** Red circle depicts collision with any wall. Green - Desired trajectory, Blue - UAV Center, Orange - End-effector position, Red - Wall. The UAV and manipulator have to avoid collision with the walls.

		Naive MPC		MPC - HC		MPC- SC		MPC- BLF	
		Case I	**Case II**	**Case I**	**Case II**	**Case I**	**Case II**	**Case I**	**Case II**
Without Disturbance		×	×	×	×	✓	×	✓	✓
	TE (m) '↓'	0.02432∗	0.02429	0.02657∗	0.02386	0.04482	0.09731	**0.02629**	**0.04750**
	c_s '↓'	0.00557∗	0.00612	0.02880∗	0.04981	**0.00443**	0.00698	0.00732	**0.03212**
	c_e '↓'	0.01427∗	0.01489	0.04787∗	0.06572	0.02326	0.02229	**0.022771**	**0.04531**
With Disturbance		×	×	×	×	×	×	✓	✓
	TE (m) '↓'	0.07366∗	0.06735	0.05430∗	0.05490∗	0.07708∗	0.11384	**0.07164**	**0.06589**
	c_s '↓'	0.08933∗	0.08942	0.10546∗	0.11304∗	0.08933∗	0.07915	**0.15592**	**0.07720**
	c_e '↓'	0.17720∗	0.20924	0.18921∗	0.18301∗	0.21039∗	0.08328	**0.23040**	**0.07887**

Table 3.4: **Algorithm Benchmarking for simulations:** For Wall Avoidance (**Case I**), '×' signifies that the AM collided with the wall. For workspace bound (**Case II**), '×' signifies that the UAV was unable to maintain its position inside the desired workspace. '*' denotes that the maneuver was incomplete due to collision with the Wall or the inability of the MPC optimizer to find a valid control input to satisfy safe operation conditions. '↓' denotes that a lower value of the performance metric is desired.

Chapter 4

Control Barrier Function-based Predictive Control for Close Proximity operation of UAVs inside a Tunnel

We augment the previous chapter [31] by including extended disturbance handling in the form of ground, ceiling and wall effects and apply these changes on a more challenging task which are the key contributions to this chaper.

This chapter introduces a method for effectively controlling the movement of an Unmanned Aerial Vehicle (UAV) within a tunnel. The primary challenge of this problem lies in the UAV's exposure to nonlinear distance-dependent torques and forces generated by the tunnel walls, along with the need to operate safely within a defined region while in close proximity to these walls. To address this problem, the chapter proposes the implementation of a Model Predictive Control (MPC) framework with constraints based on Control Barrier Function (CBF). The chapter approaches the issue in two distinct ways; first, by maintaining a safe distance from the tunnel walls to avoid the effects of both the walls and ceiling, and second, by minimizing the distance from the walls to effectively manage the nonlinear forces associated with close proximity tasks. Finally, the chapter demonstrates the effectiveness of its approach through testing on simulation for various close proximity trajectories with the realistic model of aerodynamic disturbances due to the proximity of the ceiling and boundary walls.

4.1 Introduction

Present times have witnessed the widespread deployment of Unmanned Aerial Vehicles (UAVs) in a variety of domains, ranging from delivery, search, and rescue to monitoring [39]. Certain civil inspection and delivery tasks necessitate close-range operations near stationary obstructions, such as bridges and buildings [40]. Furthermore, UAV based indoor missions involving inspection of tunnels, rooms, aircraft fuel tanks, coal mines and AC ducts, offer significant advantages over traditional manual methods by reducing the time and effort required while also minimizing risks to human safety. Nonetheless, when conducting inspection tasks in close proximity to obstacles or walls, the UAV's aerial dynamics are subject to various force and torque disturbances, leading to potential instability and safety concerns. To

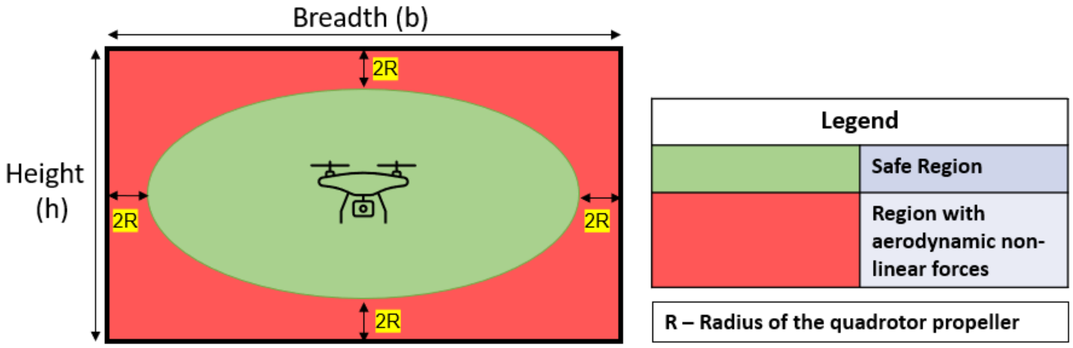
Breadth (b)
2R
Height (h)
2R
2R
2R
Legend
Safe Region
Region with aerodynamic non-linear forces
R – Radius of the quadrotor propeller

Figure 4.1: Depicts operation of the UAV in a safe region with minimal aerodynamic effects from the wall. If the UAV goes closer than 2× R from the walls, it experiences turbulent forces, which tend to destabilize the UAV and cause collision.

account for such disturbances from all directions, we demonstrate our controller for operating inside a tunnel.

The behavior of a UAV as it approaches the walls of a tunnel is characterized by nonlinear variation in its thrust, attributable to the intricate aerodynamic interactions at play [12]. As a result, a region of operation that is deemed unsafe can be identified in the vicinity of the wall or obstacle, necessitating the confinement of the UAV to a remaining safe region. Nonetheless, certain inspection tasks may require the UAV to operate in close proximity to the wall. Consequently, the controller must be designed to facilitate stability in the presence of such nonlinear disturbances. [36] demonstrates the safe distance for operation is beyond 2× *Radius of Propeller* from the obstruction or wall.

4.2 Related Work

The literature is sufficiently populated with efforts to model ceiling and ground effects [12, 36, 41, 42], but there is a clear gap in formulating control algorithms to tackle these effects in a combined fashion. Nonlinear Model Predictive Control (MPC) [24] has been used for navigation and obstacle avoidance of UAVs for real-time utilities. MPC provides the predictive ability [23] which aids in performing agile maneuvers with high precision and smooth control actions. [26] tries to limit the risk of unsafety by formulating a probabilistic guarantee, but fails to provide a rigid safety guarantee to avoid obstacles. [28] utilises partial sensor information to navigate through unknown environments by providing partial safety guarantees.

Control Barrier Function (CBF) [1] is used to guarantee safety-critical control for various domains, including dynamic robotic systems. [1] introduces safety, safety sets, and describes using CBF to enforce safety in a minimally invasive fashion by not increasing the control effort or trajectory cost. CBF has been used as a constraint to MPC [29] to provide safety guarantees while addressing the case of conflict between safety and performance. This provides improved performance to MPC while providing safety guarantees. [27] shows collision avoidance for multi UAV swarm to reach desired locations and providing safety guarantees. [31] utilizes MPC while handling external wind disturbances. Although nonlinear controllers have been tried separately for ground and ceiling effects, no effort has been made to minimize the impacts of these disruptions using a disturbance resistive barrier function.

4.3 Contributions

The chapter contributes in the following ways:

1. To the best of the author's knowledge, this chapter marks the initial endeavor to address the challenges of ground, ceiling, and wall effects simultaneously in a closed space for a UAV via the utilization of a model predictive controller.

2. This chapter also proposes the use of CBF as a bounding function to bound the UAV into the Safe region (in Fig. 4.1) to prevent interaction with the aerodynamical forces of tunnel effect.

3. The contemporary CBF function is modified to tackle disturbances and provide safety guarantees in the presence of bounded external disturbances.

The chapter follows the structure with Section IV describing the UAV dynamic, a conventional CBF, and different aerodynamic effects acting on the UAV. Section V provides the problem formulation, and Section VI describes the outer loop MPC with CBF constraints and inner loop PID. Section VII explains the simulation results for different cases, and Section VIII concludes the chapter.

4.4 Aerodynamic Ceiling, Ground and Wall effect

When a UAV's rotors start rotating, depending on whether it is near a vertical or horizontal surface, different aerodynamic forces start acting on it. These aerodynamic forces start affecting the UAV by pulling or pushing from the expected trajectory. It is crucial to understand where these forces originate and how they affect to tackle their effects.

4.4.1 Ground Effect

When a UAV flies over a horizontal surface, ground effects (GE) occur. GE is an aerodynamic effect that has been studied extensively and seen to push UAVs away from the ground [36, 41]. The theoretical model of GE presented by Cheeseman and Bennet [43] is a widely accepted thurst ratio approximation of GE as given in (4.1).

$$\text{Ground Effect: } \left[\frac{T_{GE}}{T_{\infty}}\right] = \frac{1}{1-(\frac{R}{4z})^2} \tag{4.1}$$

where T_{GE} is the Thrust into the Ground, T_{∞} is the Thrust baseline, R is the radius of the propeller and z is the distance from the ground.

4.4.2 Ceiling Effect

When a UAV flies underneath a horizontal surface, nonlinear disturbances in the form of ceiling effect (CE) acts on the UAV. Contrary to GE, CE pulls the UAV towards the surface [36]. The mathematical approximation is found as a curve in (4.2)

$$\text{Ceiling Effect: } \left[\frac{T_{CE}}{T_{\infty}}\right] = \frac{1}{1-(\frac{1}{a_1})(\frac{R}{a_2+z})^2} \tag{4.2}$$

where T_{CE} is the Thrust into the ceiling, a_1 and a_2 are coefficients obtained through an experimental least square approach.

4.4.3 Sidewall Effect

When a UAV flies close to a vertical surface, it experiences a pull toward the wall. This force is smaller than GE and CE forces and acts on the rotors randomly while pulling toward the wall, destabilizing the UAV. In [44], the chapter tried to model this effect and found it to be yaw invariant, but it could not model the effect as it could not detect the wall effect reliably. According to their experiments, the force along the X-Y axis varied by up to 0.052 N with a standard deviation of up to 0.022 N, and along the Z-axis varied by up to 0.062 N with a standard deviation of 0.065 N. Hence, these forces act randomly with these parameters.

4.4.4 Combined Tunnel effect

The combined tunnel effect refers to the two possible combinations of aerodynamic forces acting in corners. They are near the ceiling (Ceiling effect and sidewall effect) and ground (ground effect and sidewall effect). These effects were studied in [45], as In Low Corner Effect (ILoCE) and In Upper Corner Effect (IUpCE). It tries to analyse the effects and concluded that a source and drain vortex depicting the combined forces is formed in the corners. These vortexes are shown as Particle image velocimetry (PIV) images, and the force diagrams shows a higher combined force in the corners than the individual forces.

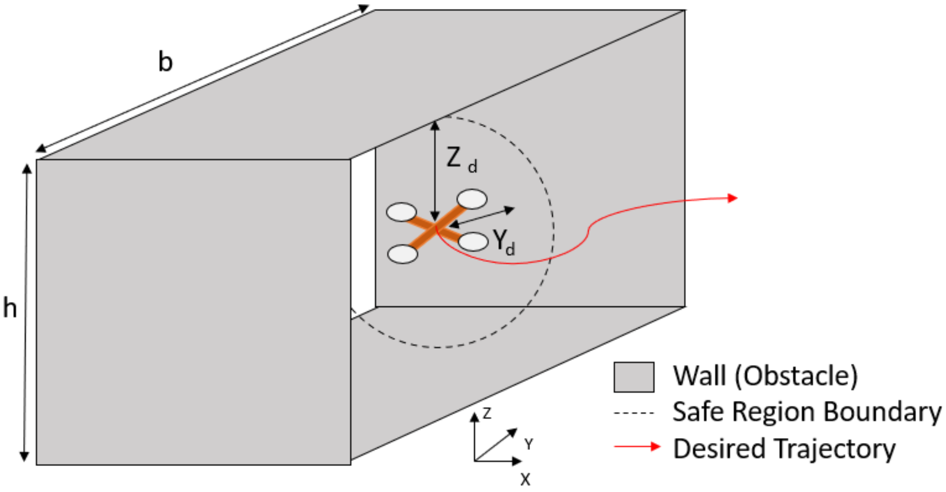

Figure 4.2: Image shows the trajectory tracking of a UAV inside a tunnel while handling effects from tunnel effects.

4.5 PROBLEM FORMULATION

The aim of the chapter is to provide a control strategy to avoid tunnel effects (combined, ceiling, sidewall and ground effects) and tackle their disturbance to provide safety guarantees when the UAV is close proximity to the tunnel walls as shown in Fig. 4.1 and Fig. 4.2. These tasks have been defined in 3 cases. We consider UAV center to be same as the UAV center of gravity.

4.5.0.1 Case I

To follow trajectory inside a tunnel while maintaining a minimum distance of 2 × Radius of Propeller (Fig. 4.1) to avoid aerodynamic interactions with the wall. The UAV will be bound inside a safe region of operation (Negligible Aerodynamic interactions) in the presence of external disturbances in the form of wind.

4.5.0.2 Case II

Minimize the safe distance of operation (z_d, y_d, $h-z_d$ and $b-y_d$) (Fig. 4.2) from the tunnel walls for close proximity operations. We minimize the safe hovering distance from the walls even in the presence of external disturbances.

4.5.0.3 Case III

To follow a trajectory with close proximity to the wall, ceiling and ground and tackle the combined tunnel aerodynamic effect.

Primary objective in trajectory tracking and hovering tasks is defined as the error ($e(\mathbf{x})$) in (4.3).

$$\min_{\mathbf{u}} e(\mathbf{x}) = ||\mathbf{p}(\mathbf{x}) - \mathbf{p}^d|| \; \forall \; k > 0 \tag{4.3}$$

where $p(\mathbf{x})$ is the position of UAV center, p^d is the desired position of the UAV center and k is the discreet time step.

4.6 PROPOSED CONTROLLER

The control architecture of the proposed controller is presented in Fig. 4.3. The control loop consists of an outer loop Model Predictive Control (MPC) with safety constraints derived from a modified Control Barrier Function (CBF). The modifications to CBF are made to restrict the UAV inside a desired safe region contrary to its earlier collision avoidance utility. A disturbance rejection term is also introduced to the conventional CBF to handle the Tunnel effect and other wind disturbances in the tunnel. The inner loop control is comprised of thrust and attitude PID control. We present our main contributions in this section. We begin by writing the discrete-time dynamics of the UAV for calculating the cost inside MPC outer loop. The state vector of the UAV is defined as $\mathbf{x_k} = [\mathbf{p_k}, \dot{\mathbf{p}}_\mathbf{k}, \Psi_k, \dot{\Psi}_k]$ where $\mathbf{p_k}$ is the position of the UAV center in the inertial frame and Ψ_k is the yaw angle of the UAV. The control input is $u_k = [\ddot{\mathbf{p}}_\mathbf{k}, \ddot{\Psi}_k]$ at time interval k. The state space model for the UAV utilized by the model MPC is given in (2.28).

4.6.1 Model Predictive Control (Outer loop)

The optimal control problem for each time step k is given in (4.4) for the UAV dynamics.

$$u_k^{opt} = \min_{\mathbf{u}} g(\mathbf{x_k}, \mathbf{u_k}, t_k) \tag{4.4a}$$

$$\text{s.t. } \dot{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{4.4b}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max} \tag{4.4c}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{4.4d}$$

u_k^{opt} depicts the optimal input by the optimizer which is then given to the inner loop controller for tracking. The cost g is the weighted sum of N_g cost functions $g = \sum_{i=1}^{N_g} g_i$ given below. N_g is the number of cost functions and N denotes the prediction horizon of the MPC.

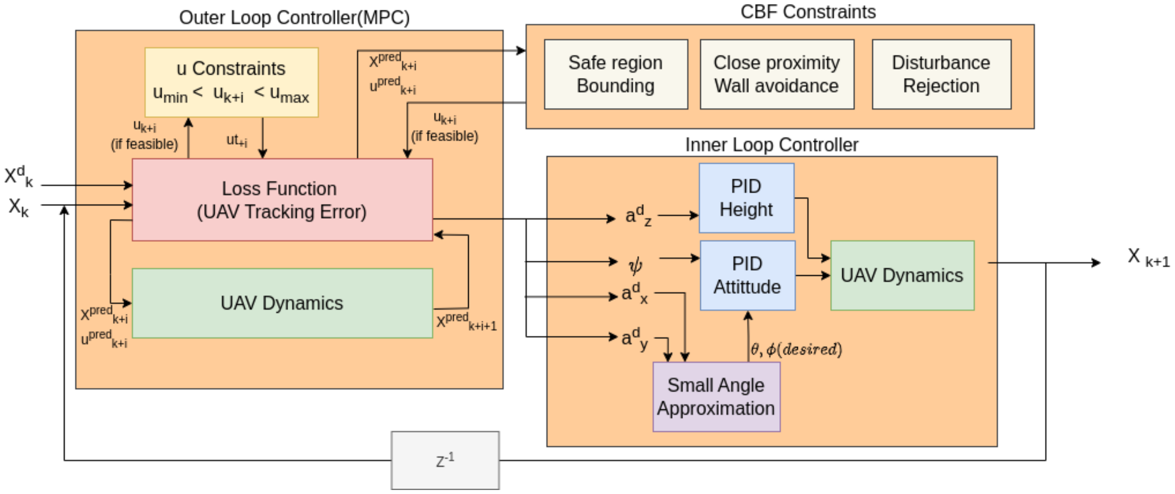

Figure 4.3: **Control Architecture:** The Outer loop control for the UAV is a Model Predictive controller which provides the optimal control input to the Inner loop control (PID) to track while additional constraints to the MPC are derived from the CBF. x_k^d is the desired state of the UAV.

4.6.1.1 UAV center tracking error

To account for the penalization of drift from the desired position or trajectory, we add a cost to the MPC optimizer as in (4.5)

$$g_1 = \sum_{i=0}^{N-1} (\left|\left|\mathbf{p}(\mathbf{x_{k+i}}) - \mathbf{p}^d_{k+i}\right|\right|^2_{W_1}) + \left|\left|\mathbf{p}(\mathbf{x_{k+N}}) - \mathbf{p}^d_{k+N}\right|\right|^2_{W_{s_1}} \tag{4.5}$$

where W_1 and W_{s_1} are weight matrices.

4.6.1.2 UAV center velocity error

To penalize the higher velocity of the UAV, we add a cost to the MPC optimizer as in (4.6).

$$g_2 = \sum_{i=0}^{N-1} (||\dot{\mathbf{p}}(\mathbf{x}_{k+i})||^2_{W_2}) + ||\dot{\mathbf{p}}(\mathbf{x}_{k+N})||^2_{W_{s_2}} \tag{4.6}$$

where W_2 and W_{s_2} are weight matrices.

The controller with only MPC as the outer loop and PID as the inner loop is referred to as **Naive MPC** in the following sections. **MPC-HC** is demonstrated as Naive MPC with hard constraints on the optimizer, not in the form of CBF. These algorithms would be utilized to compare the performance of the proposed controller. Additional constraints for **MPC-HC** are: For **Case I**, $||\mathbf{p} - d^s|| \leq r$ and for **Case III**, $||\mathbf{d}|| \geq d_s$ which have been explained in Section VI part C.

4.6.2 PID (Inner loop)

The inner loop PID receives a u_k^{opt} as the optimal u_k from the MPC optimizer. Desired roll Θ and pitch Φ angles are calculated using small angle analysis, and the desired thrust and attitude are tracked by PID Thrust and Attitude Controllers.

4.6.3 CBF Constraints

4.6.3.1 Bounding UAV in safe region (Bounding condition)

For **Case I**, We give our primary contribution to bound the UAV inside the safe region where aerodynamic effects do not hamper the stability of the UAV. To assume a continuous differentiable bounding area, we choose the safe region to be a spherical boundary similar to Fig. (4.1), as the tunnel effect and other effects together form a region where the safe region can be simplified to a sphere. The UAV can only leave the safe region in a radial direction. We constrain the movement of the UAV for a high velocity motion using CBF. The CBF for one direction is given in 4.7, and we replace $\mathbf{p}$ with $-\mathbf{p}$ to get CBF in the opposite direction.

$$h_1(\mathbf{x}_k) = \sqrt{\frac{2(\mathbf{p}_{k+i})^T a_{max}}{||\mathbf{p}_{k+i}||}(||\mathbf{p}_{k+i}|| - r)} + \frac{\mathbf{p}_{k+i}^T}{||\mathbf{p}_{k+i}||}(\dot{\mathbf{p}}_{k+i} - \dot{\mathbf{p}}_{k+i}^t) \tag{4.7}$$

Where r is the radius of the safe region.

4.6.3.2 Minimize safe distance of operation from tunnel Walls (Disturbance Rejection)

For **Case II**, we can tighten the bound of the CBF using an additional disturbance rejection parameter λ to tackle aerodynamic disturbances from various effects. Hence we change the earlier invariance condition in (2.30) to the condition in (4.8).

$$\dot{h}(\mathbf{x}) + \gamma(h^z(\mathbf{x}) - \lambda) \geq 0 \tag{4.8}$$

4.6.3.3 Trajectory tracking for close proximity flights

For **Case III**, the CBF is modified to avoid walls and the CBF condition for this task is given in (4.9).

$$h_2(\mathbf{x}_k) = \sqrt{\frac{2\mathbf{d}(\mathbf{x}_{k+i})^T a_{max}}{||\mathbf{d}(\mathbf{x}_{k+i})||}(||\mathbf{d}(\mathbf{x}_{k+i})|| - d^s)} + \frac{\mathbf{d}(\mathbf{x}_{k+i})^T}{||\mathbf{d}(\mathbf{x}_{k+i})||}\dot{\mathbf{p}}_{k+i} \tag{4.9}$$

where $d(\mathbf{x}_k)$ is the perpendicular distance from the wall at time instance k and d^s is the minimum safe distance from the wall. The combination

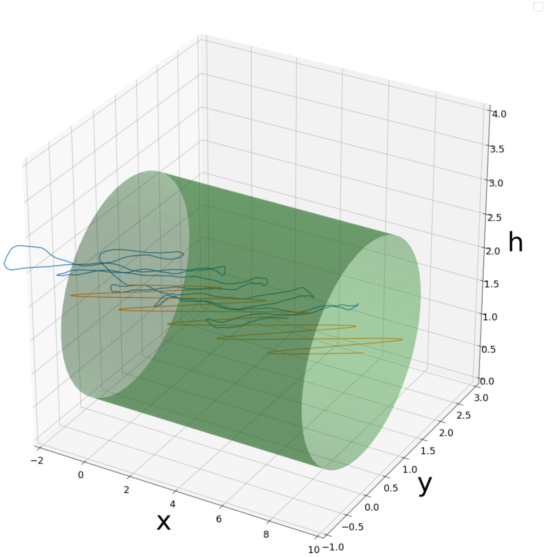

Figure 4.4: **UAV position to maintain safe region (Case I) (Naive MPC)** It shows that the UAV leaves the safe region for Naive MPC. Green - Safe Region Boundary, Blue - UAV Center, Orange - Desired Trajectory.

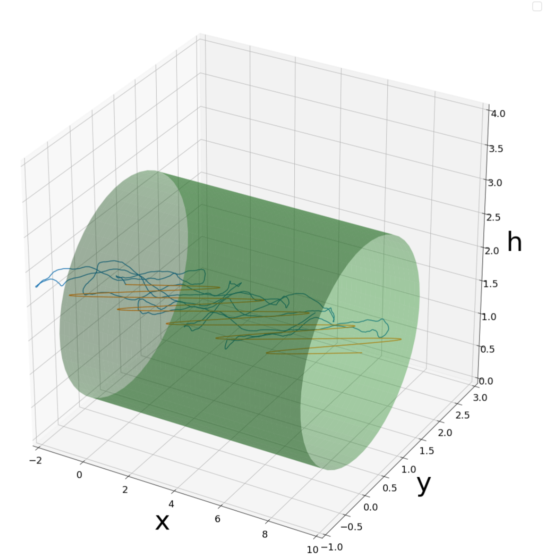

Figure 4.5: **UAV position to maintain safe region (Case I) (MPC-HC)** It shows that the UAV leaves the safe region for MPC-HC. Green - Safe Region Boundary, Blue - UAV Center, Orange - Desired Trajectory.

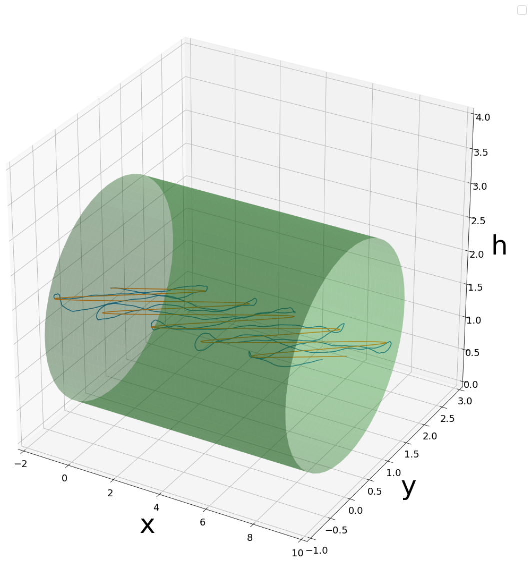

Figure 4.6: **UAV position to maintain safe region (Case I) (MPC-CBF)** It shows that the UAV maintains the safe region for MPC-CBF. Green - Safe Region Boundary, Blue - UAV Center, Orange - Desired Trajectory.

4.7 Simulation Results

This section presents the results of the performance of the algorithm on simulation. Python 3 was used to perform the scenario on an Intel® Core™ i7-8550U CPU desktop operating at 1.80 GHz. The optimizer used is the 'SLSQP' method provided in the scipy library [46]. The specifications of the UAV model used are given in Table 4.1 and the parameters used in **MPC-CBF** are given in Table. 4.2.

Parameter	**Value**
Mass	$1.5\ kg$
Arm length	$0.20\ m$
Propeller Diameter	$0.24\ m$
Moment of Inertia - UAV	$I_x = 0.1\ kg\ m^2$, $I_y = 0.1\ kg\ m^2$, $I_z = 0.2\ kg\ m^2$
UAV attitude constraints	$\|\theta\| \leq \pi/10\ rad$, $\|\phi\| \leq \pi/10\ rad$

Table 4.1: Specifications of the UAV: These parameters have been taken from the UAV used to define the Aerodynamic effects

Parameter	**Value**
MPC Weights	$w_1 = 10 \times \mathbf{I}_{3\times3}$, $w_{s_1} = 50 \times \mathbf{I}_{3\times3}$, $w_2 = 2 \times \mathbf{I}_{3\times3}$, $w_{s_2} = 10 \times \mathbf{I}_{3\times3}$
u_k Initialization	$\mathbf{0}_{1\times4n}$
γ	3
λ	8
z	3
Sampling step (t_s)	0.1 s
Total time (t)	100 s
Max wind disturbance	$d_m = 0.8\,m/s^2$

Table 4.2: Weights and Parameters for MPC and CBF

4.7.1 Metric for performance comparison

We measure the performance of the algorithm with the following matrices:

- Bounding inside Safe region
- Trajectory Tracking error, $T_e = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(\mathbf{p}(\mathbf{x}_k) - p_k^d)^2}$
- Control effort, $c_e = \sum_{k=0}^{N-1}||\mathbf{u}_k||^2$
- Control Smoothness, $c_s = \sum_{k=0}^{N-1}|\Delta\mathbf{u}_k|$

4.7.2 Results for Case I

For bounding the UAV inside the safe region, **Naive MPC** is unable to find the bounds and shows very high Trajectory tracking error in the presence of wind disturbances. **MPC - HC** is unable to maintain the bound when the UAV gets a high velocity input. **MPC-CBF** performs best compared to other algorithms because it incorporates obstacle avoidance and disturbance rejection using CBF. It shows a 30% decrease in the trajectory error and maintains the safe region's bound. Trajectory tracking results are shown in Fig. 4.12 depicting the trajectory in 3D. The performance matrices are mentioned in Table. 4.3.

4.7.3 Results for Case II

The shortest distance between the walls and the UAV depicts the extended stability zone of the UAV when deploying a new control algorithm. **Naive MPC** gives the minimal distance as 2× R while **MPC-CBF** shows a decrease in this distance by 45% as shown in Table. 4.3.

4.7.4 Results for Case III

When the UAV trajectory passes through the unsafe region, **Naive MPC** and **MPC-HC** are unable to maintain the trajectory and subsequently collide to the wall. Only **MPC-CBF** is able to maintain the trajectory while reducing the control effort by 1̃5% thus reducing the power consumed by the UAV.

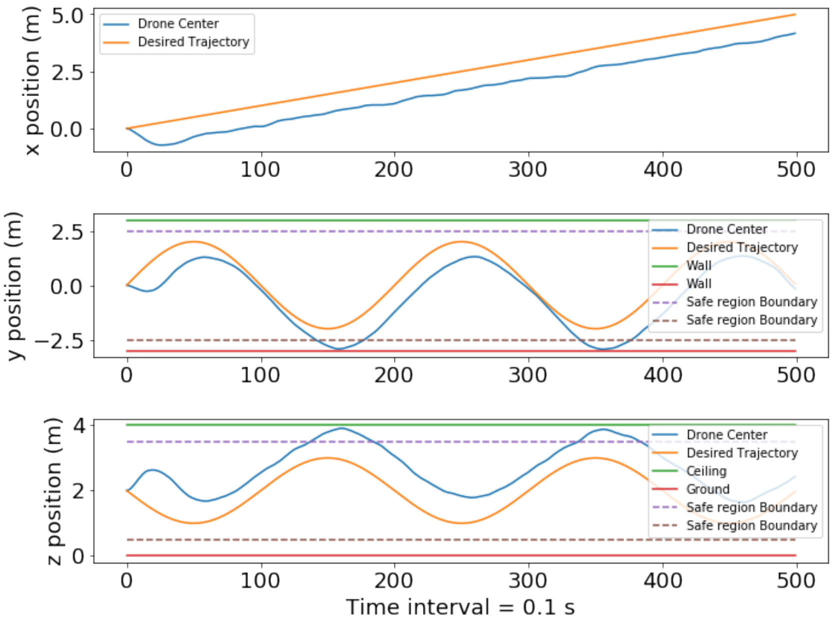

Figure 4.7: **UAV position to maintain safe region (Case I) (Naive MPC)** It shows that the UAV leaves the safe region for Naive MPC.

4.8 Chapter Conclusion

The chapter shows that a Model predictive controller, when combined with constraints using Control Barrier Function, can provide safety guarantees when flying inside a tunnel. The controller also reduces the safe hovering distance from the wall by 37% and incorporates high disturbance tolerance. It is also shown that flying near the ground and ceiling can reduce the UAV's power consumed (control effort) by 1̃5%. The algorithm's efficacy provides safety guarantees while travelling inside a tunnel using

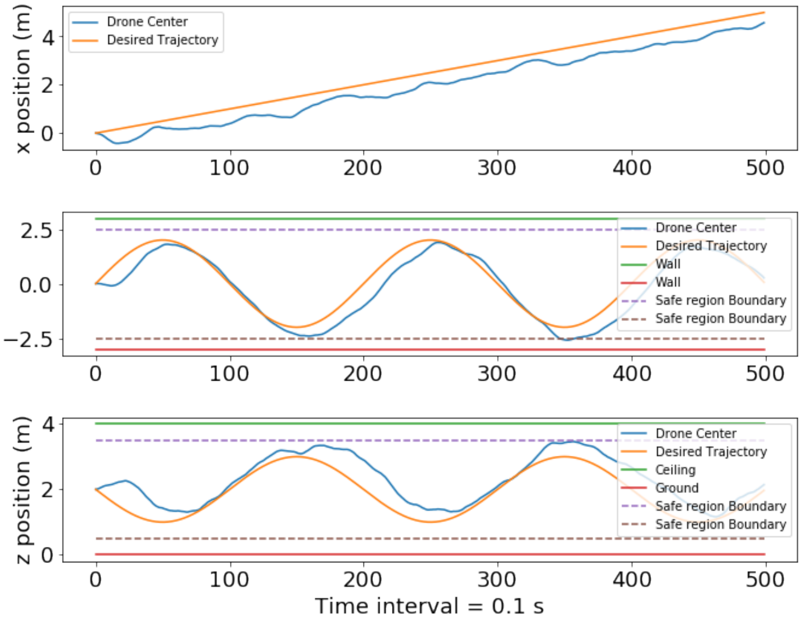

Figure 4.8: **UAV position to maintain safe region (Case I) (MPC-HC)** It shows that the UAV leaves the safe region for MPC-HC.

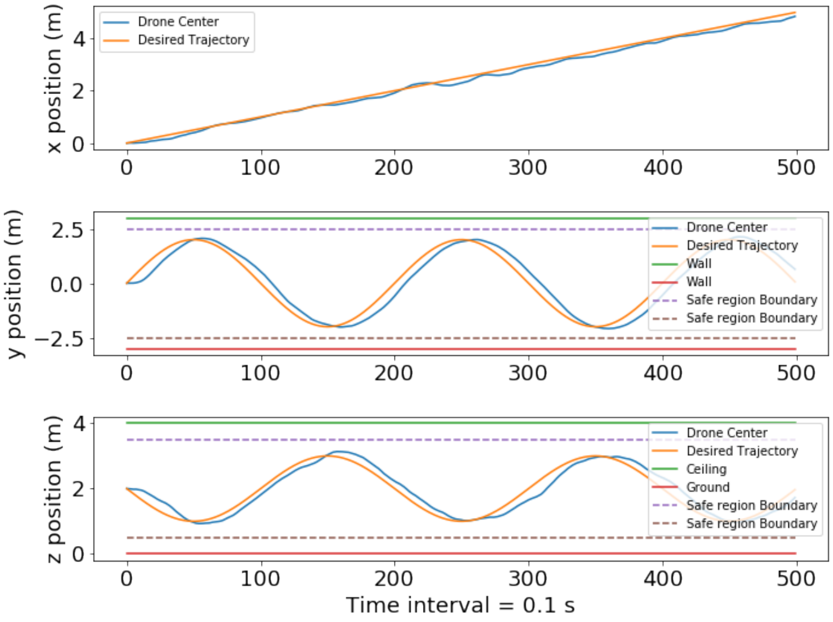

Figure 4.9: **UAV position to maintain safe region (Case I) (MPC-CBF)** It shows that the UAV maintains the safe region for MPC-CBF.

UAV Tasks		Naive MPC	MPC - HC	MPC - CBF
Bounding in safe region (**Case I**)	Maintain Boundary	×	×	✓
	$T_e(m)$	1.16250	0.70152	**0.51087**
	c_e	0.95237	0.93959	**0.74246**
	c_s	0.09519	0.12497	**0.09190**
Minimum distance to wall (**Case II**)	Ground effect (m)	0.495	0.521	**0.312**
	Ceiling effect (m)	0.502	0.478	**0.298**
	Sidewall effect (m)	0.481	0.465	**0.138**
Close proximity trajectory tracking (**Case III**)	Collision	✓	✓	×
	$T_e(m)$	1.38771	0.8327	**0.56010**
	c_e	1.11998	0.99483	**0.89015**
	c_s	0.10604	0.09343	**0.06222**

Table 4.3: **Algorithm benchmarking:** We compare the Trajectory rms error, control effort and control smoothness of MPC-CBF while flying amidst external disturbances with other algorithms, and it performs substantially better than all other algorithms.

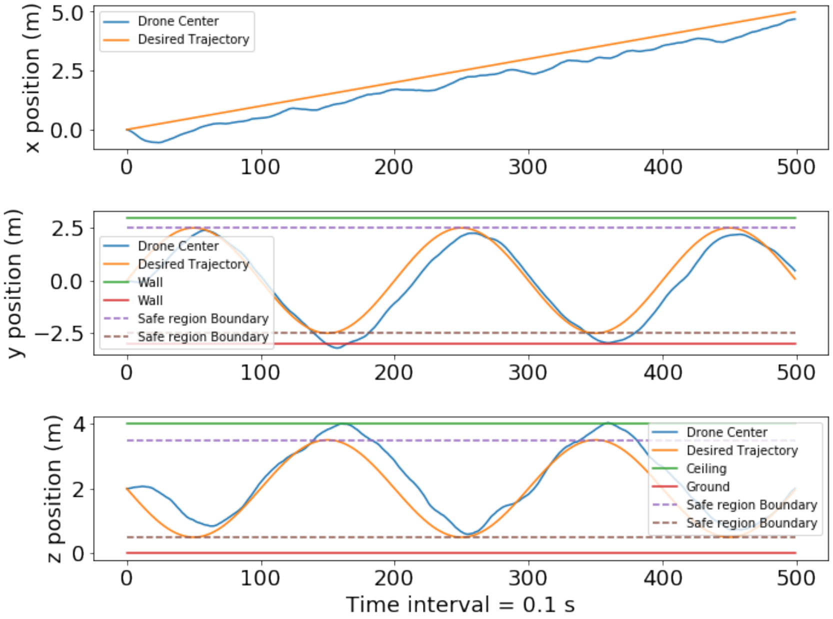

Figure 4.10: **UAV position while trajectory in unsafe region (Case III) (Naive MPC)** It shows that the UAV collides with the wall for Naive MPC.

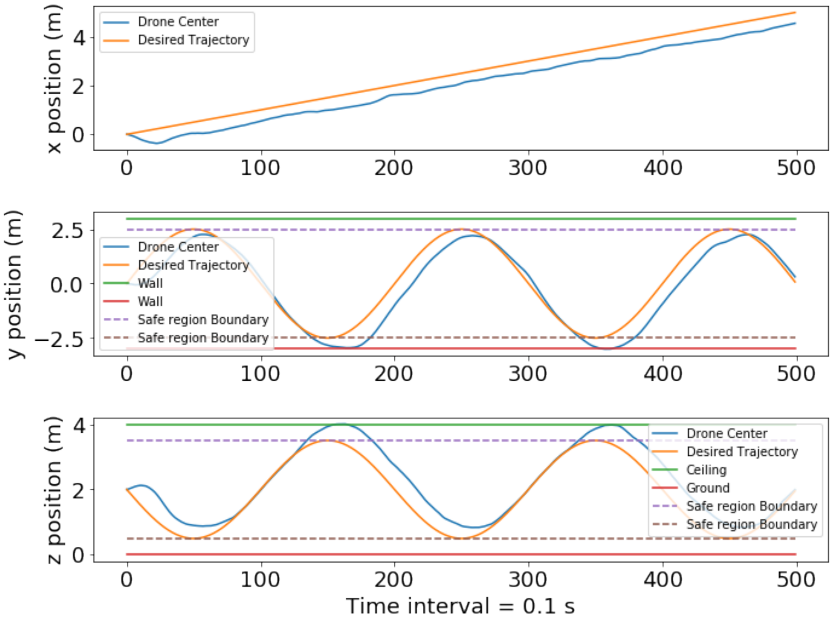

Figure 4.11: **UAV position while trajectory in unsafe region (Case III) (MPC-HC)** It shows that the UAV collides with the wall for MPC-HC.

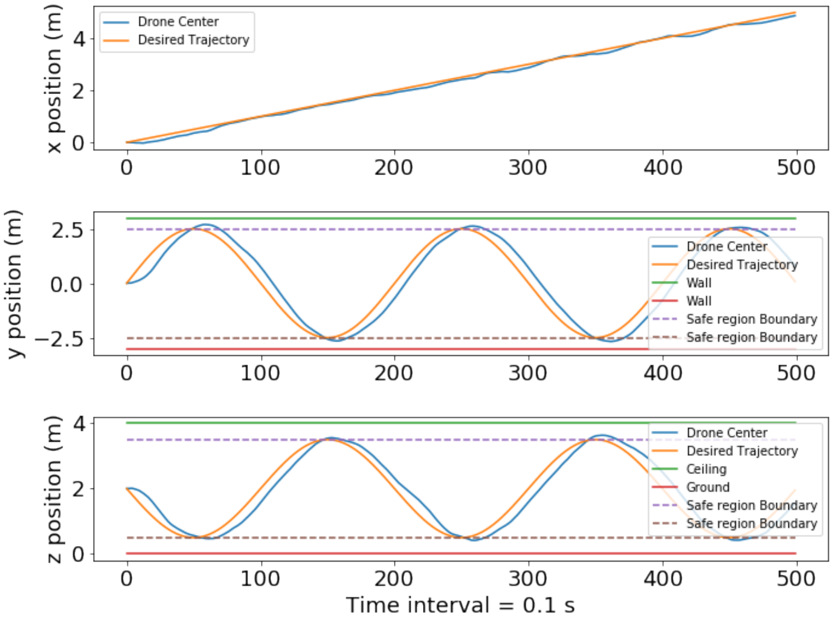

Figure 4.12: **UAV position while trajectory in unsafe region (Case III) (MPC-CBF)** It shows that the UAV maintains the trajectory for MPC-CBF.

parameters from a real UAV model. Future work shall include using vision based learning models to detect obstacles and create barrier functions through their understanding.

Chapter 5

Conclusion

The presented dissertation explains that Barrier Lyapunov Function when combined with optimal control strategies like the Model Predictive Controller incorporates safety and improvises the control algorithms stability and trajectory tracking abilities. MPC-BLF and MPC-CBF provide enhanced trajectory tracking for Aerial robots while providing safety guarantees. The proposed constraints are modifiable for various avoidance and bounding tasks, while providing no unnecessary hindrance to the MPC controller cost optimizer. This controller algorithm can also tackle external wind disturbances within a known bound to provide safety guarantees.

In MPC-BLF for an Aerial Manipulator, the BLF-based Model predictive controller with a primary objective of safe operation in the proximity of static objects. Our approach shows how BLF, formulated for barrier avoidance and free space tracking objectives, shows robust behavior for the safe operation of an Aerial Manipulator in the presence of external disturbances. A state-of-the-art MPC-based method is modified using two types of constraints and compared with the proposed method, which shows significant improvement in two cases i.e with and without external disturbances. This is validated in simulation using parameters from a real laboratory scale AM.

In MPC-CBF for maneuvering a UAV inside a tunnel, we present a control framework to handle aerodynamic torques by rotor and wall interactions. These forces are nonlinear and difficult to model, hence we define constraints to tackle these forces through a CBF. The MPC when combined with constraints using Control Barrier Function, can provide safety guarantees when flying inside a tunnel. The controller also reduces the safe hovering distance from the wall by 37% and incorporates high disturbance tolerance. It is also shown that flying near the ground and ceiling can reduce the UAV's power consumed (control effort) by approximately 15%. The algorithm's efficacy provides safety guarantees while travelling inside a tunnel using parameters from a real UAV model.

In it's entirety, the contributions of this dissertation are suitable for complex environments, making this approach a valuable tool for a variety of applications from search and rescue to manipulation. Overall, the proposed MPC-CBF and MPC-BLF controllers provide an efficient and robust control strategy for various aerial robots to ensure collision free trajectory tracking while handling various unknown external disturbances.

5.1 Future Work

Future work will include determining these CBF bounds with the help of vision algorithms and utilizing them for safe tracking without human intervention. This would lead to a pathway for fully autonomous high-level control systems.

Various sensor and calibration errors can be accounted for in the study to make it closer to the real world, which can further be deployed on the Aerial Manipulator and UAV hardware setup.

Techniques can also be developed for multi-UAV systems using MPC controllers with dynamic obstacle avoidance.

The proposed controllers will be deployed on actual Aerial manipulator and UAV hardware.

Moreover, the alogirhtm could be made more robust by handling uncertainty in the ”Prediction Model”.

Appendix A

Algorithm Details

A.1 MPC Implementation

We have provided the implementation of Model Predictive Control Algorithm.
The implementation of MPC algorithm has the following components:

- System Model
- Cost function
- Prediction Horizon
- Optimization problem solver

A.1.1 System Model

System Model has been explained in 3.5 and 2.20 for Aerial Manipulator and UAV respectively.

A.1.2 Cost Function

```
for i in range(n):

        cost_ee_x_error += (x_man_pred - x_man_des)
       @ w1 @ (x_man_pred - x_man_des).T
        cost_ee_v_error += (v_man_pred) @ w2 @ (v_man_pred).T

        cost_input += (x_input_i) @ wu @ (x_input_i).T

        [x_pred,
        xdot_pred,
        thetadesired_pred ,
        omega_pred ,
        theta_m_pred ,
        thetadot_m_desired_pred ,
        x_man_pred ,
        v_man_pred] = drone_dyanmics(
            adesired[i],
            yawdesired[i],
            thema_m_desired[i],
            x_pred ,
            xdot_pred ,
            omega_pred ,
            theta_m_pred ,
            thetadot_m_pred ,
        )

    cost_ee_x_error += (x_man_pred - x_man_des) @ ws1 @ (x_man_pred - x_man_des).T
    cost_ee_v_error += (v_man_pred) @ ws2 @ (v_man_pred).T

    cost += cost_ee_x_error + cost_ee_v_error + cost_input
```

A.1.3 Prediction Horizon

Prediction horizon has been mentioned for all MPC algorithms in the previous chapters.

A.1.4 Optimatizer

The algorithms are implemented on Python 3 on an Intel® Core™ i7-8550U CPU PC running at 1.80 GHz. We use the 'SLSQP' method from scipy as the non-linear optimizer for MPC.

Related Publications

1. Mundheda, V., Mirakhor, K., RahulK, S., Kandath, H., Govindan, N. (2022). Predictive Barrier Lyapunov Function Based Control for Safe Trajectory Tracking of an Aerial Manipulator. *2023 European Control Conference (ECC) (Accepted) ArXiv, abs/2212.04625*.

Bibliography

[1] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431, 2019.

[2] Francisco J. Perez-Grau, J. Ramiro Martinez-de Dios, Julio L. Paneque, J. Joaquin Acevedo, Arturo Torres-González, Antidio Viguria, Juan R. Astorga, and Anibal Ollero. Introducing autonomous aerial robots in industrial manufacturing. *Journal of Manufacturing Systems*, 60:312–324, 2021.

[3] Pedro J. Sanchez-Cuevas, Antonio Gonzalez-Morgado, Nicolas Cortes, Diego B. Gayango, Antonio E. Jimenez-Cano, Aníbal Ollero, and Guillermo Heredia. Fully-actuated aerial manipulator for infrastructure contact inspection: Design, modeling, localization, and control. *Sensors*, 20(17), 2020.

[4] Elena Ciampa, Luca De Vito, and Maria Rosaria Pecce. Practical issues on the use of drones for construction inspections. *Journal of Physics: Conference Series*, 1249(1):012016, may 2019.

[5] Parham Nooralishahi, Clemente Ibarra-Castanedo, Shakeb Deane, Fernando López, Shashank Pant, Marc Genest, Nicolas P. Avdelidis, and Xavier P. V. Maldague. Drone-based non-destructive inspection of industrial sites: A review and case studies. *Drones*, 5(4), 2021.

[6] Sreehari Sreenath, Haroon Malik, Narman Husnu, and Kanimozhi Kalaichelavan. Assessment and use of unmanned aerial vehicle for civil structural health monitoring. *Procedia Computer Science*, 170:656–663, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[7] Sharifah Mastura Syed Mohd Daud, Mohd Yusmiaidil Putera Mohd Yusof, Chong Chin Heo, Lay See Khoo, Mansharan Kaur Chainchel Singh, Mohd Shah Mahmood, and Hapizah Nawawi. Applications of drone in disaster management: A scoping review. *Science Justice*, 62(1):30–42, 2022.

[8] Hafiz Suliman Munawar, Fahim Ullah, Amirhossein Heravi, Muhammad Jamaluddin Thaheem, and Ahsen Maqsoom. Inspecting buildings using drones and computer vision: A machine learning approach to detect cracks and damages. *Drones*, 6(1), 2022.

[9] Adetayo Olugbenga Onososen, Innocent Musonda, Damilola Onatayo, Motheo Meta Tjebane, Abdullahi Babatunde Saka, and Rasaki Kolawole Fagbenro. Impediments to construction site digitalisation using unmanned aerial vehicles (uavs). *Drones*, 7(1), 2023.

[10] Idris Jeelani and Masoud Gheisari. Safety challenges of uav integration in construction: Conceptual analysis and future research roadmap. *Safety Science*, 144:105473, 2021.

[11] Jianjie Wu, Limei Peng, Jiawen Li, Xinyu Zhou, Jingbing Zhong, Cynthia Wang, and Jun Sun. Rapid safety monitoring and analysis of foundation pit construction using unmanned aerial vehicle images. *Automation in Construction*, 128:103706, 2021.

[12] Darius J Carter, Lauren Bouchard, and Daniel B Quinn. Influence of the ground, ceiling, and sidewall on micro-quadrotors. *AIAA Journal*, 59(4):1398–1405, 2021.

[13] Stephen A. Conyers, Matthew J. Rutherford, and Kimon P. Valavanis. An empirical evaluation of ceiling effect for small-scale rotorcraft. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 243–249, 2018.

[14] Zahra Samadikhoshkho and Michael Lipsett. Decoupled control design of aerial manipulation systems for vegetation sampling application. *Drones*, 7(2), 2023.

[15] R. Naldi, A. Macchelli, N. Mimmo, and L. Marconi. Robust control of an aerial manipulator interacting with the environment∗∗this work has been partially supported by the european project airborne (ict 780960). *IFAC-PapersOnLine*, 51(13):537–542, 2018. 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018.

[16] Bo Hang Wang, Dao Bo Wang, Zain Anwar Ali, Bai Ting Ting, and Hao Wang. An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control*, 52(7-8):731–739, 2019.

[17] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018.

[18] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Introduction to the special issue on aerial manipulation. *IEEE Robotics and Automation Letters*, 3(3):2734–2737, 2018.

[19] Xiangdong Meng, Yuqing He, and Jianda Han. Survey on aerial manipulator: System, modeling, and control. *Robotica*, 38:1288 – 1317, 2019.

[20] Somasundar Kannan, Marouane Alma, Miguel A. Olivares-Mendez, and Holger Voos. Adaptive control of aerial manipulation vehicle. In *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, pages 273–278, 2014.

[21] Dimitris Chaikalis, Farshad Khorrami, and Anthony Tzes. Adaptive control approaches for an unmanned aerial manipulation system. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 498–503, 2020.

[22] Dario Lunni, Angel Santamaria-Navarro, Roberto Rossi, Paolo Rocco, Luca Bascetta, and Juan Andrade-Cetto. Nonlinear model predictive control for aerial manipulation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–93, 2017.

[23] Tomas Baca, Daniel Hert, Giuseppe Loianno, Martin Saska, and Vijay Kumar. Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6753–6760, 2018.

[24] Björn Lindqvist, Sina Sharif Mansouri, Ali-akbar Agha-mohammadi, and George Nikolakopoulos. Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles. *IEEE Robotics and Automation Letters*, 5(4):6001–6008, 2020.

[25] Kapil Sachan and Radhakant Padhi. Barrier lyapunov function based output-constrained control of nonlinear euler-lagrange systems. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 686–691, 2018.

[26] Astghik Hakobyan and Insoon Yang. Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *IEEE Transactions on Robotics*, 38(2):939–957, 2022.

[27] Pravin Mali, K. Harikumar, Arun Kumar Singh, K. Madhava Krishna, and P.B. Sujit. Incorporating prediction in control barrier function based distributive multi-robot collision avoidance. In *2021 European Control Conference (ECC)*, pages 2394–2399, 2021.

[28] Danilo Saccani, Leonardo Cecchin, and Lorenzo Fagiano. Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, pages 1–16, 2022.

[29] Zahra Marvi and Bahare Kiumarsi. Safety planning using control barrier function: A model predictive control scheme. In *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, pages 1–5, 2019.

[30] Dongjae Lee, Dohyun Jang, Hoseong Seo, and H. Jin Kim. Model predictive control for an aerial manipulator opening a hinged door. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 986–991, 2019.

[31] Vedant Mundheda, Karan Mirakhor, Rahul K S, Harikumar Kandath, and Nagamanikandan Govindan. Predictive barrier lyapunov function based control for safe trajectory tracking of an aerial manipulator, 2022.

[32] Zhe Wu, Fahad Albalawi, Zhihao Zhang, Junfeng Zhang, Helen Durand, and Panagiotis D. Christofides. Control lyapunov-barrier function-based model predictive control of nonlinear systems. In *2018 Annual American Control Conference (ACC)*, pages 5920–5926, 2018.

[33] Moses Bangura and Robert Mahony. Real-time model predictive control for quadrotors. *IFAC Proceedings Volumes*, 47(3):11773–11780, 2014. 19th IFAC World Congress.

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[35] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E. O'Brien. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A*, 107(3), mar 2023.

[36] Pedro J Sanchez-Cuevas, Victor Martín, Guillermo Heredia, and Aníbal Ollero. Aerodynamic effects in multirotors flying close to obstacles: modelling and mapping. In *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pages 63–74. Springer, 2020.

[37] Nursultan Imanberdiyev, Sumil Sood, Dogan Kircali, and Erdal Kayacan. Design, development and experimental validation of a lightweight dual-arm aerial manipulator with a cog balancing mechanism. *Mechatronics*, 82:102719, 2022.

[38] Li Wang, Aaron D. Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[39] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018.

[40] Sina Tavasoli, Xiao Pan, and TY Yang. Real-time autonomous indoor navigation and vision-based damage assessment of reinforced concrete structures using low-cost nano aerial vehicles. *Journal of Building Engineering*, page 106193, 2023.

[41] Antonio Matus-Vargas, Gustavo Rodriguez-Gomez, and Jose Martinez-Carranza. Ground effect on rotorcraft unmanned aerial vehicles: A review. *Intelligent Service Robotics*, 14(1):99–118, 2021.

[42] Pedro Jesus Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Multirotor uas for bridge inspection by contact using the ceiling effect. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 767–774. IEEE, 2017.

[43] IC Cheeseman and WE Bennett. The effect of the ground on a helicopter rotor in forward flight. 1955.

[44] Christopher D McKinnon and Angela P Schoellig. Estimating and reacting to forces and torques resulting from common aerodynamic disturbances acting on quadrotors. *Robotics and Autonomous Systems*, 123:103314, 2020.

[45] S Prothin, C Fernandez Escudero, T Jardin, and N Doue. Archean: aerodynamics of rotors in confined environments study in ground and corner effect. In *10th international micro-air vehicles conference, Melbourne, Australia*, 2018.

[46] Pauli Virtanen, Ralf Gommers, and et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, feb 2020.