Control Barrier Functions in Dynamic UAVs for Kinematic Obstacle Avoidance: A Collision Cone Approach

Manan Tayal[1], Rajpal Singh[2], Jishnu Keshavan[2], Shishir Kolathaya[1]

***Abstract*— Unmanned aerial vehicles (UAVs), specifically quadrotors, have revolutionized various industries with their maneuverability and versatility, but their safe operation in dynamic environments heavily relies on effective collision avoidance techniques. This paper introduces a novel technique for safely navigating a quadrotor along a desired route while avoiding kinematic obstacles. We propose a new constraint formulation that employs control barrier functions (CBFs) and collision cones to ensure that the relative velocity between the quadrotor and the obstacle always avoids a cone of vectors that may lead to a collision. By showing that the proposed constraint is a valid CBF for quadrotors, we are able to leverage its real-time implementation via Quadratic Programs (QPs), called the CBF-QPs. Validation includes PyBullet simulations and hardware experiments on Crazyflie 2.1, demonstrating effectiveness in static and moving obstacle scenarios. Comparative analysis with literature, especially higher order CBF-QPs, highlights the proposed approach's less conservative nature.**

I. Introduction

Quadrotors are used in a wide range of applications, including search and rescue, environmental monitoring, agriculture, transportation, and entertainment [1]. In many of these applications, quadrotors operate in complex and dynamic environments, where they must navigate around obstacles such as trees, buildings, and other drones. The literature presents a variety of methods such as artificial potential field [2], reachability analysis [3] [4], and nonlinear model predictive control [5] to address the problem of obstacle avoidance in UAVs.

In recent years, the Control Barrier Functions (CBFs) based approach [6] [7] has emerged as a promising strategy for ensuring safe operation of autonomous systems. This is a model-based control design method, which provides a computationally efficient solution that can handle complex situations while guaranteeing safety. CBFs can be formulated as a Quadratic Problem (QP) and can be solved online, making them well-suited for real-time safety-critical applications. CBFs are specifically designed to enforce safety constraints and provide hard constraints on the system's trajectory, making them superior to Nonlinear MPC (NMPC) in terms of safety guarantees. NMPC, on the other hand, provides soft constraints on the system's trajectory, with the degree of constraint satisfaction dependent on the optimization algorithm's performance.

This research is supported by ARTPARK.

[1]Cyber-Physical Systems, Indian Institute of Science (IISc), Bengaluru. {manantayal, shishirk}@iisc.ac.in .

[2]Department of Mechanical Engineering, Indian Institute of Science (IISc), Bengaluru. {rajpalsingh, kjishnu}@iisc.ac.in .

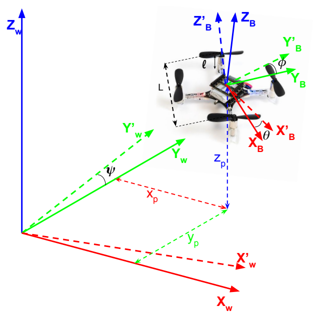


Fig. 1: World coordinates and body fixed coordinates of Crazyflie and Euler's angles defined in these coordinates

In situations that involve complex interactions between subsystems or where safety requirements are highly dynamic and subject to frequent changes, reachability analysis may be limited. In such cases, CBFs are more suitable due to their ability to handle highly dynamic safety requirements [8]. While the artificial potential field approach is easy to implement, it suffers from limitations such as the possibility of getting stuck in local minima and difficulties in handling complex environments with multiple obstacles and [9] shows that CBFs offer a viable, and arguably improved alternative to APFs for real-time obstacle avoidance. Thus, the Control Barrier Functions approach provides a better solution for obstacle avoidance in UAVs, particularly in safety-critical scenarios. [10] has shown collision avoidance using CBFs in planar quadrotor case. In a recent work, [11], [12] showed that Higher order CBFs are a generalized form of the exponential CBFs, thus it also addresses this problem of obstacle avoidance. However, a major challenge with this approach is the need to identify suitable penalty parameters (p's) and class $\mathcal{K}$ functions (α's) that can yield optimal results.

With regards to obstacle avoidance in dynamic environments, another class of approaches that is widely used is the method of collision cones [13], [14], [15] It involves defining a cone-shaped region between two objects to represent the potential area of collision, which can be avoided by adjusting the object's trajectory to prevent the relative velocity vector from falling within the cone. This approach has several advantages, including its simplicity, efficiency, and adaptability to different environments. The method can be easily integrated into existing motion planning algorithms, takes into account the speed and direction of moving objects and the shape and size of potential obstacles, and can work in dynamic and unpredictable environments. As a result, the Collision Cone approach provides a reliable and flexible means of avoiding collisions in various robotic and autonomous systems.

Despite its simplicity and effectiveness, the collision cone method has largely been restricted to offline motion planning/navigation problems, and its extensions for real-time implementations have been limited. However, by exploiting the CBF-QP formulations, we can synthesize a new class of CBFs through the notion of collision cones, which can then be implemented in real-time. This will be the main objective of this paper. This idea was originally proposed in [16] for the planar case (2D) and for wheeled robots, while we aim to extend this for 3D and for quadrotors, which are underactuated and have higher degrees of freedom (DoF).

A. Contribution and Paper Structure

The main idea is to realize a CBF-QP formulation for the quadrotor dynamics and for obstacles with non-zero velocity values. The main contributions of our work are:

- We formulate a direct method for safe trajectory tracking of quadrotors based on collision cone control barrier functions expressed through a quadratic program.
- We consider static and constant velocity obstacles of various dimensions and provide mathematical guarantees for collision avoidance.
- We compare the collision cone CBF with the state-of-the-art higher-order CBF (HO-CBF), and show how the former is better in terms of feasibility and safety guarantees.

B. Organisation

The rest of this paper is organized as follows. Preliminaries explaining the quadrotor model, the concept of control barrier functions (CBFs), collision cone CBFs, and controller design are introduced in section II. The application of the above CBFs on the quadrotor to avoid obstacles of various shapes is discussed in section III. The simulation setup and results will be discussed in section IV. Finally, we present our conclusion in section V.

II. Preliminaries

In this section, first, we will describe the dynamics of quadrotor. Next, we will formally introduce Control Barrier Functions (CBFs) and their importance for real-time safety-critical control. Finally, we will introduce Collision Cone Control Barrier Function (C3BF) approach.

A. Quadrotor model

The quadrotor model has four propellers, which provides upward thrusts of (f_1, f_2, f_3, f_4) (see Fig. 1) and the states needed to describe the quadrotor system is given by $x = [x_p, y_p, z_p, \dot{x}_p, \dot{y}_p, \dot{z}_p, \phi, \theta, \psi, \omega_1, \omega_2, \omega_3]$. The quadrotor dynamics is as follows [17]:

$$\underbrace{\begin{bmatrix}\dot x_p\\ \dot y_p\\ \dot z_p\\ \ddot x_p\\ \ddot y_p\\ \ddot z_p\\ \dot\phi\\ \dot\theta\\ \dot\psi\\ \dot\omega_1\\ \dot\omega_2\\ \dot\omega_3\end{bmatrix}}_{\dot x} = \underbrace{\begin{bmatrix} \dot x_p\\ \dot y_p\\ \dot z_p\\ 0\\ 0\\ -g\\ \mathbf{W}^{-1}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}\\ -I^{-1}\vec\omega\times I\vec\omega\\ .\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\\ \frac{1}{m_Q}\mathbf{R}\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&1&1&1\end{bmatrix}\\ \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\\ I^{-1}L\begin{bmatrix}1&0&-1&0\\0&1&0&-1\\c_\tau&-c_\tau&c_\tau&-c_\tau\end{bmatrix}\end{bmatrix}}_{g(x)}\underbrace{\begin{bmatrix}f_1\\ f_2\\ f_3\\ f_4\end{bmatrix}}_{u} \tag{1}$$

x_p, y_p, and z_p denote the coordinates of the vehicle's center of the base of the quadrotor in an inertial frame. ϕ, θ and ψ represents the (roll, pitch & yaw) orientation of the quadrotor. (see Fig. 1). $\mathbf{R}$ is the rotation matrix (from the body frame to the inertial frame), m_Q is the mass of the quadrotor, $\mathbf{W}$ is the transformation matrix for angular velocities from the inertial frame to the body frame, I is the inertia matrix and L is the diagonal length of the quadrotor. c_τ is the constant that determines the torque produced by each propeller. Note that even though we restrict our study to quadrotors in this paper, the extension of the proposed CBF-QPs for different multi-rotor UAVs is straightforward.

B. Control barrier functions (CBFs)

Having described the vehicle models, we now formally introduce Control Barrier Functions (CBFs) and their applications in the context of safety. Given the quadrotor model, we have the nonlinear control system in affine form:

$$\dot{x} = f(x) + g(x)u \tag{2}$$

where $x \in \mathcal{D} \subseteq \mathbb{R}^n$ is the state of system, and $u \in \mathbb{U} \subseteq \mathbb{R}^m$ the input for the system. Assume that the functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are continuously differentiable. Specific formulation of f, g for the quadrotor were described in (1). Given a Lipschitz continuous control law $u = k(x)$, the resulting closed loop system $\dot{x} = f_{cl}(x) = f(x) + g(x)k(x)$ yields a solution $x(t)$, with initial condition $x(0) = x_0$. Consider a set $\mathcal{C}$ defined as the *super-level set* of a continuously differentiable function $h : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ yielding,

$$\mathcal{C} = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) \geq 0\} \tag{3}$$

$$\partial\mathcal{C} = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) = 0\} \tag{4}$$

$$\mathrm{Int}\,(\mathcal{C}) = \{x \in \mathcal{D} \subset \mathbb{R}^n : h(x) > 0\} \tag{5}$$

It is assumed that $\mathrm{Int}\,(\mathcal{C})$ is non-empty and $\mathcal{C}$ has no isolated points, i.e. $\mathrm{Int}\,(\mathcal{C}) \neq \phi$ and $\overline{\mathrm{Int}\,(\mathcal{C})} = \mathcal{C}$. The system is safe w.r.t. the control law $u = k(x)$ if $\forall\, x(0) \in \mathcal{C} \implies x(t) \in \mathcal{C} \quad \forall t \geq 0$. We can mathematically verify if the controller $k(x)$ is safeguarding or not by using Control Barrier Functions (CBFs), which is defined next.

Definition 1 (Control barrier function (CBF)): Given the set $\mathcal{C}$ defined by (3)-(5)*, with $\frac{\partial h}{\partial x}(x) \neq 0\ \forall x \in \partial\mathcal{C}$, the function h is called the control barrier function (CBF) defined on the set $\mathcal{D}$, if there exists an extended class $\mathcal{K}$ function κ such that for all $x \in \mathcal{D}$:*

$$\underbrace{sup}_{u\in\mathbb{U}} \left[\underbrace{\mathcal{L}_f h(x) + \mathcal{L}_g h(x) u}_{\dot{h}(x,u)} + \kappa\left(h(x)\right) \right] \geq 0 \tag{6}$$

where $\mathcal{L}_f h(x) = \frac{\partial h}{\partial x} f(x)$ and $\mathcal{L}_g h(x) = \frac{\partial h}{\partial x} g(x)$ are the Lie derivatives.

Given this definition of a CBF, we know from [7] and [18] that any Lipschitz continuous control law $k(x)$ satisfying the inequality: $\dot{h} + \kappa(h) \geq 0$ ensures safety of $\mathcal{C}$ if $x(0) \in \mathcal{C}$, and asymptotic convergence to $\mathcal{C}$ if $x(0)$ is outside of $\mathcal{C}$.

C. Safety Filter Design

Having described the CBF, we can now describe the Quadratic Programming (QP) formulation of CBFs. CBFs act as *safety filters* which take the desired input $u_{des}(x,t)$ and modify this input in a minimal way:

$$\begin{aligned} u^*(x,t) &= \underset{u\in\mathbb{U}\subseteq\mathbb{R}^m}{\arg\min} \|u - u_{des}(x,t)\|^2 \\ &\text{s.t. } \mathcal{L}_f h(x) + \mathcal{L}_g h(x) u + \kappa\left(h(x)\right) \geq 0 \end{aligned} \tag{7}$$

This is called the Control Barrier Function based Quadratic Program (CBF-QP). The CBF-QP control u^* can be obtained by solving the above optimization problem using KKT conditions.

D. Collision Cone CBF (C3BF) candidate for quadrotors

We now formally introduce the proposed CBF candidate for quadrotors. Let us assume that the obstacle is centered at $(c_x(t), c_y(t), c_z(t))$ and with dimensions c_1, c_2, c_3. We assume that $c_x(t), c_y(t), c_z(t)$ are differentiable and their derivatives are piece-wise constants. The proposed approach combines the idea of potential unsafe directions given by collision cone (Fig. 2, 3) as an unsafe set to formulate a CBF as in [16]. Consider the following CBF candidate:

$$h(x,t) = < p_{\text{rel}}, v_{\text{rel}} > + \|p_{\text{rel}}\| \|v_{\text{rel}}\| \cos\phi, \tag{8}$$

where p_{rel} is the relative position vector between the body center of the quadrotor and the center of the obstacle, v_{rel} is the relative velocity, $< \cdot, \cdot >$ is the dot product of 2 vectors and ϕ is the half angle of the cone, the expression of $\cos\phi$ is given by $\frac{\sqrt{\|p_{\text{rel}}\|^2 - r^2}}{\|p_{\text{rel}}\|}$ (see Fig. 2, 3). Precise mathematical definitions for $p_{\text{rel}}, v_{\text{rel}}$ will be given in the next section. The proposed constraint simply ensures that the angle between $p_{\text{rel}}, v_{\text{rel}}$ is less than $180° - \phi$.

In [16], it was shown that the proposed candidate (8) is valid CBF for wheeled mobile robots, i.e., the unicycle and bicycle. With this result, CBF-QPs were constructed that yielded collision-avoiding behaviors in these models. We aim to extend this to the class of quadrotors.

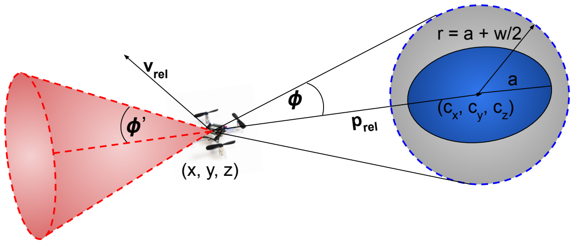


Fig. 2: **3D CBF** candidate: The dimensions of the obstacle are comparable to each other, it can be assumed as a sphere

III. Collision Cone CBFs on Quadrotor

Having described the Collision Cone CBF candidate, we will see their application on quadrotors in this section. Based on the shape of the obstacle we can divide the proposed candidates into two cases:

A. 3D CBF candidate

In scenarios where the dimensions of the obstacle are roughly equal, we can model the obstacle as a sphere, as illustrated in Fig. 2. The resulting CBF in this context is referred to as the **3D CBF**. The relative position vector between the body center of the quadrotor and the center of the obstacle is as follows:

$$p_{\text{rel}} := \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} - \left(\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \right) \tag{9}$$

Here l is the distance of the body center from the base (see Fig. 1). c_x, c_y, c_z represents the obstacle location as a function of time. Also, since the obstacles are of constant velocity, we have $\ddot{c}_x = \ddot{c}_y = \ddot{c}_z = 0$. We obtain its relative velocity as

$$v_{\text{rel}} := \dot{p}_{rel} \tag{10}$$

Having introduced Collision Cone CBF candidates in II-D, the next step is to formally verify that they are, indeed, valid CBFs. We have the following result.

Theorem 1: Given the quadrotor model (1)*, the proposed CBF candidate* (8) *with $p_{\text{rel}}, v_{\text{rel}}$ defined by* (9), (10) *is a valid CBF defined for the set $\mathcal{D}$.*
Please refer to [19, Thm 3] for the proof of Theorem.

B. Projection CBF candidate

When an obstacle has significantly disparate dimensions, it can be approximated as a cylinder, giving rise to the **Projection CBF** (refer to Fig. 3). To derive this, we calculate the relative position vector between the quadrotor's body center and the intersection point of the obstacle's axis with the projection plane, which is perpendicular to the axis. Thus, we obtain:

$$(p_{\text{rel}})_{proj} := \mathcal{P}\left(\begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} - \left(\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \right) \right). \tag{11}$$

Here l is the distance of the body center from the base (see Fig. 1). $\mathcal{P} : \mathbb{R}^3 \to \mathbb{R}^3$ is the projection operator, which can

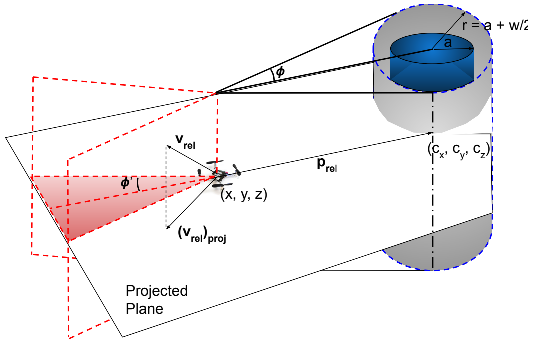


Fig. 3: **Projection CBF** candidate: One of the dimensions, of the obstacle, is bigger than the other dimensions, it can be assumed as a cylinder.

be assumed to be a constant[1]. Now, since the relative position lies on the projection plane, we have one more condition to satisfy:

$$<(p_{\rm rel})_{proj},\hat{n}>=0, \tag{12}$$

where, $\hat{n}$ is the normal to the plane. Also, the relative velocity is given by:

$$(v_{\rm rel})_{proj}:=\frac{d(p_{rel})_{proj}}{dt}=(\dot{p}_{\rm rel})_{proj} \tag{13}$$

Theorem 2: Given the quadrotor model (1)*, the proposed CBF candidate* (8) *with* $p_{\rm rel}, v_{\rm rel}$ *defined by* (11), (13) *is a valid CBF defined for the set* $\mathcal{D}$.
Please refer to [19, Thm 4] for the proof of Theorem.

C. Comparison with Higher Order CBFs

We introduce the state-of-the-art Higher Order Control Barrier Functions (HO-CBFs) and compare them with the proposed C3BF in this section. Given that the collision constraints are with respect to position, the associated CBF has a relative degree of two. Therefore, it is necessary to establish a higher-order CBF with $m=2$ as outlined in [12, Eq. 16], which is expressed as:

$$\begin{aligned}\psi_1(x,t)&=\dot{b}(x,t)+p\alpha_1(b(x,t))\\ \psi_2(x,t)&=\dot{\psi}_1(x,t)+p\alpha_2(\psi_1(x,t)),\end{aligned} \tag{14}$$

where $b(x,t)=(c_x(t)-x_p)^2+(c_y(t)-y_p)^2+(c_z(t)-z_p)^2-r^2$, and r is the encompassing radius given by $r=max(c_1,c_2,c_3)$. α_1,α_2 are both class $\mathcal{K}$ functions, and p is a tunable constant. As explained previously, c_x,c_y,c_z is the center location of the obstacle as a function of time. Let us examine the form of HO-CBF where α_1 is a square root function (which is also strictly increasing), and α_2 is a linear function, due to its similarity to C3BF. Consequently, the resulting Higher Order CBF candidate takes the following form:

$$h_{HO}(x,t)=<p_{\rm rel},v_{\rm rel}>+\gamma\sqrt{(\|p_{\rm rel}\|^2-r^2)}. \tag{15}$$

[1]Note that the obstacles are always translating and not rotating. In addition, it is not restrictive to assume that the translation direction is always perpendicular to the cylinder axis. This makes the projection operator a constant.

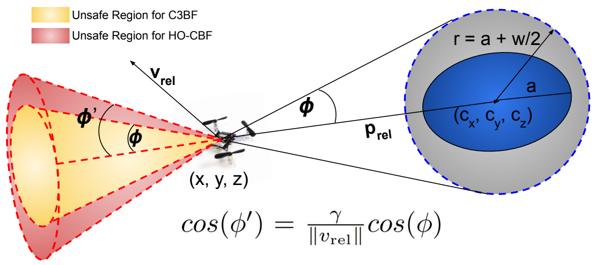


Fig. 4: Comparison of HO-CBF with C3BF. Here we are trying to compare the ϕ' and ϕ obtained from the two CBF formulations. It can be observed that ϕ' (pink cone) is dependent on v_{rel}, while ϕ (yellow cone) is a constant. The HO-CBF guarantees safety for a set that is not only smaller but also dependent on v_{rel} as shown by the pink cone. Hence, HO-CBF is more conservative compared to C3BF.

We can show that the above-mentioned HO-CBF is also a valid CBF for quadrotors. We will now compare it with the proposed C3BF.

The C3BF concept aims to prevent the $v_{\rm rel}$ vector, which represents the relative velocity between the quadrotor and the obstacle, from entering the collision cone region defined by the half-angle ϕ. Figures 2, 3 and 4 illustrate this idea. We can rewrite the HO-CBF formula presented in (15) in the following form:

$$h_{HO}(x,t)=<p_{\rm rel},v_{\rm rel}>+\|p_{\rm rel}\|\|v_{\rm rel}\|cos(\phi') \tag{16}$$

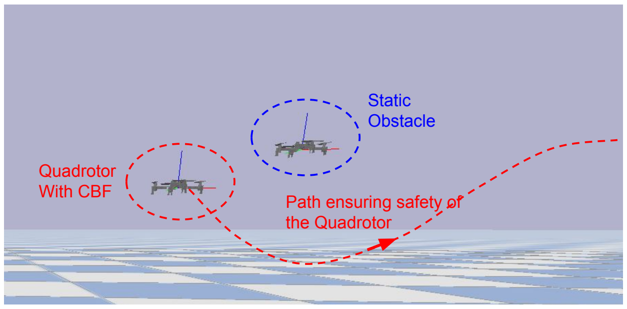
where, $cos(\phi')=\frac{\gamma}{\|v_{\rm rel}\|}cos(\phi)$. If we are able to identify a suitable γ (penalty term) for the given HO-CBF, it would result in a valid CBF as per [12]. Nonetheless, in such a scenario where γ remains constant and $\|v_{\rm rel}\|$ goes on increasing, it leads to an increase in ϕ', thus, overestimating the cone as can be seen in Fig.4. Conversely, with the C3BF approach, we permit the penalty term to vary over time, i.e., $\gamma=\|v_{\rm rel}\|$, resulting in a more precise estimation of the collision cone compared to the HO-CBF case. This also shows that C3BF is not a special case of Higher Order CBF. This is also evident from the simulation outcomes of both CBFs, as demonstrated in Section IV.

IV. Results and Discussions

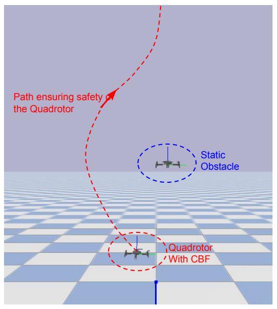
We have validated the C3BF-QP based controller on quadrotors for both 3D and Projection CBF cases. We have used a PD controller as a reference controller to track the desired path with constant target velocities. Note that the reference controller can be replaced by any existing trajectory tracking/path-planning like the MPC [20]. For the class $\mathcal{K}$ function in the CBF inequality, we chose $\kappa(h)=\gamma h$, where $\gamma=1$.

A. Simulation setup

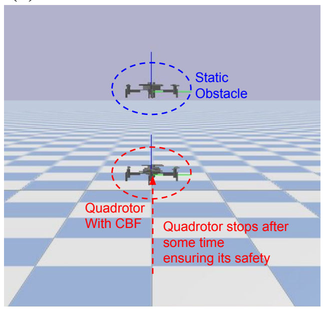
The simulations were conducted using the multi-drone environment [21] on Pybullet [22], a Python-based physics simulation engine. The parameters of Crazyflie are tabulated in I. Having presented our proposed control method design, we now test our framework under three different scenarios to



(a)



(b)



(c)

Fig. 5: Interaction with static obstacles: overtaking (a), (b), and braking (c) behavior of the quadrotor, Section III-A.In all these cases the reference velocity of the quadrotor is 1m/s.

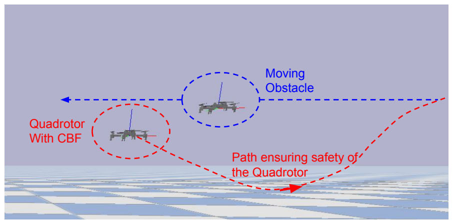
illustrate the controller performance. These scenarios include the interaction of a quadrotor with (1) a static obstacle (3D case) Fig. 5, (2) a moving obstacle (3D case) Fig. 6 and (3) an elongated obstacle (Projection case) Fig. 7.

B. *Experimental Results*

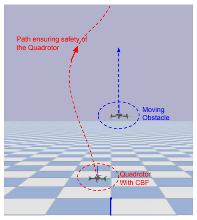
The experimental results with Bitcraze™ Crazyflie 2.1 aerial drone are presented to demonstrate the efficacy of the C3BF controller framework. The global position of the drone as well as the obstacle is measured using Qualisys™ Miqus M3 motion capture system with a tracking frequency of 100 Hz. Further, for the drone, the global position from the motion capture system is fused with the onboard IMU data via the Extended Kalman Filter to get the filtered state. The control commands are generated by an off-board computer and transmitted to the drone via a radio link. The communication with the drone is facilitated through the Crazyflie Python library [23]. Experiments are performed for the cases with a single static obstacle, multiple static obstacles, and moving obstacles. The graphs and videos of hardware experiments are available here[2]. Hence, the experimental results verify the efficacy of the proposed scheme for obstacle avoidance.

Variables	Definition	Value
g	Gravitational acceleration	$9.81 kg \cdot m/s^2$
m	Mass of quadrotor	$0.027 kg$
L	Distance between two opp. rotors	$0.130\ m$
l	Distance of center from base	$0.014 m$
Ix, Iy	Inertia about x, y-axis	$2.39 \cdot 10^{-5} kg \cdot m^2$
Iz	Inertia about z-axis	$3.23 \cdot 10^{-5} kg \cdot m^2$
kf	Motor's thrust constant	$3.16 \cdot 10^{-10}$
km	Motor's torque constant	$7.94 \cdot 10^{-12}$

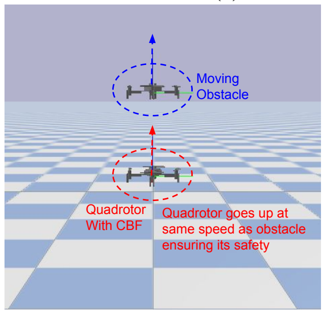
TABLE I: Modelling parameters of Crazyflie



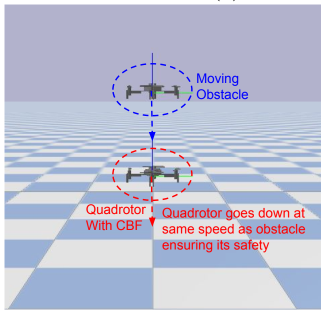
(a)



(b)

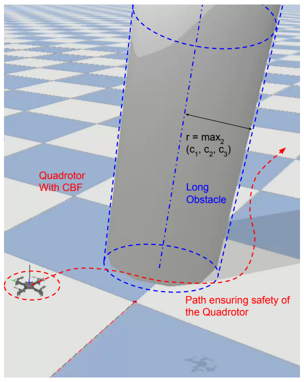

Moving Obstacle
Quadrotor With CBF
Quadrotor goes up at same speed as obstacle ensuring its safety

(c)

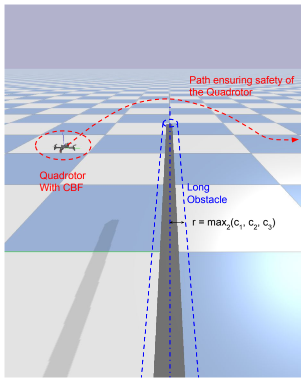

Moving Obstacle
Quadrotor With CBF
Quadrotor goes down at same speed as obstacle ensuring its safety

(d)

Fig. 6: Interaction with moving obstacles: overtaking (a), (b), slowing (c), and reversing (d) behavior of the quadrotor, section III-A. In all these cases the reference velocity of the quadrotor is 1m/s and the obstacle quadrotor speed is 1m/s in case (a) and 0.1 m/s in (b),(c),(d).



(a)



(b)

Fig. 7: Interaction with longer obstacles: moving from side (a) and top (b), Section III-B. In all these cases the reference velocity of the quadrotor is 1m/s.

C. *Comparison between C3BF and HO-CBF*

All the aforementioned cases were tested with the HO-CBF to compare its performance against C3BF. We observe that the HO-CBF could not avoid a high-speed approaching obstacle. Moreover, it is not able to properly avoid the longer obstacles in the projection CBF case. These shortcomings of the Higher Order CBF are also demonstrated in the simulation video available here[2].

[2] https://tayalmanan28.github.io/C3BF-UAV/

D. Robustness of C3BF

Without changing the above control framework we can observe that the C3BF is robust in the following two cases:

1) Multiple Obstacles: The quadrotor successfully navigates through a series of obstacles (both Spherical and Long obstacles) avoiding collisions and showcasing robustness as in Fig. 8 (a) & (b).

2) Multiple quadrotors with C3BF-QPs: In multi-agent scenarios, where multiple quadrotors employ the collision cone CBF-QP (Fig. 8 (c)), both the ego-quadrotor and the approaching quadrotor are able to avoid collision in different configurations (static or moving), thus demonstrating robustness with respect to obstacles following the same Collision Cone CBF controller.

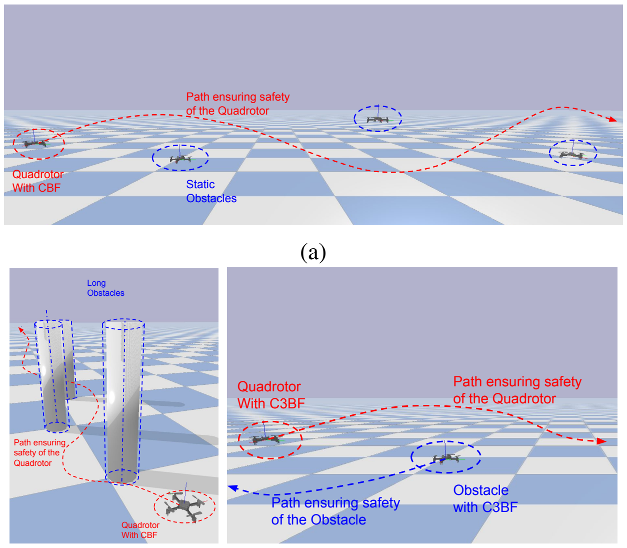

Path ensuring safety of the Quadrotor
Quadrotor With CBF
Static Obstacles
(a)
Long Obstacles
Path ensuring safety of the Quadrotor
Quadrotor With CBF
(b)
Quadrotor With C3BF
Path ensuring safety of the Quadrotor
Path ensuring safety of the Obstacle
Obstacle with C3BF
(c)

Fig. 8: Robustness in scenarios with multiple obstacles (a), (b) and with obstacle also following Collision Cone CBF(c).

V. Conclusions

We presented the extension of a novel collision cone CBF formulation for quadrotors to avoid collision with static and moving obstacles of various shapes and sizes. We successfully constructed CBF-QPs with the proposed CBF for the quadrotor model and guaranteed safety by avoiding moving obstacles. This includes collision avoidance with spherical and cylindrical obstacles. We also showed that the current state-of-the-art Higher Order CBFs is more conservative and fails in certain scenarios (shown in the video). Finally, we demonstrated the robustness of the proposed CBF-QP controller for safe navigation in a cluttered environment consisting of multiple obstacles and agents with the same safety filters. In our future work, we plan to implement the controller on quadrotors in real-world situations, interacting with a variety of obstacles. We also intend to explore applications such as the safe teleoperation of quadrotors. Additionally, we aim to investigate the potential of applying the C3BF formulation to legged robots walking in confined spaces.

References

[1] V. Kumar, “The future of flying robots,” *TED Talk*, 2015.

[2] J. Sun, J. Tang, and S. Lao, “Collision avoidance for cooperative uavs with optimized artificial potential field algorithm,” *IEEE Access*, vol. 5, pp. 18 382–18 390, 2017.

[3] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.

[4] Y. Lin and S. Saripalli, “Collision avoidance for uavs using reachable sets,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 226–235.

[5] M. Castillo-Lopez, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, “Model predictive control for aerial collision avoidance in dynamic environments,” in *2018 26th Mediterranean Conference on Control and Automation (MED)*, 2018, pp. 1–6.

[6] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE Conference on Decision and Control*, 2014.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, aug 2017. [Online]. Available: https://doi.org/10.1109%2Ftac.2016.2638961

[8] Z. Li, “Comparison between safety methods control barrier function vs. reachability analysis,” 2021. [Online]. Available: https://arxiv.org/abs/2106.13176

[9] A. W. Singletary, K. Klingebiel, J. R. Bourne, N. A. Browning, P. T. Tokumaru, and A. D. Ames, “Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance,” *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8129–8136, 2021.

[10] G. Wu and K. Sreenath, “Safety-critical control of a planar quadrotor,” in *2016 American Control Conference (ACC)*, 2016, pp. 2252–2258.

[11] W. Xiao and C. Belta, “Control barrier functions for systems with high relative degree,” *CoRR*, vol. abs/1903.04706, 2019. [Online]. Available: http://arxiv.org/abs/1903.04706

[12] ——, “High-order control barrier functions,” *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[13] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using the relative velocity paradigm,” in *IEEE International Conference on Robotics and Automation*, 1993, pp. 560–565 vol.1.

[14] ——, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. [Online]. Available: https://doi.org/10.1177/027836499801700706

[15] A. Chakravarthy and D. Ghose, “Obstacle avoidance in a dynamic environment: a collision cone approach,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, 1998.

[16] P. Thontepu, B. G. Goswami, M. Tayal, N. Singh, S. S. P I, S. S. M G, S. Sundaram, V. Katewa, and S. Kolathaya, “Collision cone control barrier functions for kinematic obstacle avoidance in ugvs,” in *2023 Ninth Indian Control Conference (ICC)*, 2023, pp. 293–298.

[17] C. Folkestad, S. X. Wei, and J. W. Burdick, “Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1344–1350.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *2019 18th European Control Conference (ECC)*, 2019.

[19] M. Tayal, B. Giri Goswami, K. Rajgopal, R. Singh, T. Rao, J. Keshavan, P. Jagtap, and S. Kolathaya, “A Collision Cone Approach for Control Barrier Functions,” *arXiv e-prints*, p. arXiv:2403.07043, 2024.

[20] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[21] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, “Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[22] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for robotics and machine learning,” http://pybullet.org.

[23] Bitcraze. (2018) Crazyflie python library. hhttps://github.com/bitcraze/crazyflie-lib-python.