Proceedings of the ASME 2016 Dynamic Systems and Control Conference
DSCC 2016
October 12-14, 2016, Minneapolis, USA

DSCC2016-9913

SAFETY-CRITICAL CONTROL OF A 3D QUADROTOR WITH RANGE-LIMITED SENSING

Guofan Wu
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, Pennsylvania, 15213.
Email: gwu@andrew.cmu.edu.

Koushil Sreenath
Depts. of Mechanical Engineering, Robotics
Institute, and Electrical & Computer Engineering,
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213.
Email: koushils@cmu.edu.

ABSTRACT

Enforcing safety is critical for aerial robotics. In this paper we consider the safety control problem for a 3D quadrotor with limited sensing range subject to avoiding collisions with time-varying obstacles. By using the concepts of Control Lyapunov Functions (CLFs) and Control Barrier functions (CBFs), we propose a control algorithm that explicitly considers the nonlinear and underactuated dynamics of a quadrotor to strictly guarantee time-varying safety-critical constraints. We demonstrate the feasibility of our proposed control design through numerical validation of (a) aerial flight through a region of dense cluttered obstacles, and (b) aerial flight through a dense time-varying obstacle field.

1 Introduction

Traditionally, dynamic motion of mobile robots is realized through a strict decoupling of planning from control, wherein dynamically feasible trajectories are generated offline to avoid unsafe regions, while the controller will track these planned trajectories online subject to various constraints. Although these approaches work well when the characteristic speed of robot is small, several challenges arise as the UAV is moving in a much more aggressive manner. On one side, fast agile movement of the UAV requires precise planning of dynamically feasible motions and reliable feedback controller for tracking. On the other hand, for high-speed UAVs operating in cluttered environments with local information, planned trajectories become stale almost immediately as new dynamic obstacles are constantly detected.

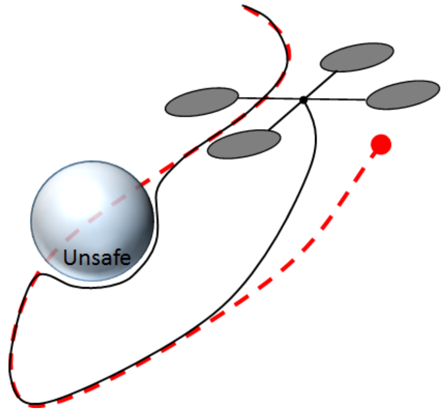


FIGURE 1: Constrained Control of a 3D quadrotor. A reference trajectory (dashed line) and a obstacle are shown. The control goal is to track the reference trajectory while simultaneously strictly enforcing that the state remains outside the safe region. The proposed controller automatically relaxes tracking of the trajectory to enforce strict safety, thereby requiring only an approximate, potentially unsafe, trajectory plan.

Faced with these challenges, researchers have come up with faster and precise algorithms from the perspectives of both plan-

This work is supported in part by the Google Faculty Research Award and in part by NSF grants CMMI-1538869, IIS-1464337, IIS-1526515.

ning and control. To boost the performance of planning algorithms of UAVs subject to external obstacles, some researchers propose computationally efficient methods which can convert the planning problem into a convex programming or a mixed-integer convex programming [10, 13]. By segmenting the free space into convex polytopes and allocating an indicator variable to each basis polynomial, R. Deits et al. [10] is able to let a quadrotor UAV safely navigate through a cluttered environment using mixed-integer semidefinite programming (MI-SDP). Although this work is computationally efficient in addressing large numbers of obstacles, the obstacles are assumed to be static and global information of the environment is required.

For the same problem, M. Watterson et al. [13], proposed a multi-scale planning method which can be divided into a global planner and a local planner. The global planner provides important waypoints to track the goal through Delaunay triangulation, while the local planner takes care of obstacle avoidance at the lower level using local sensor information. Since both planners are based on quadratic programming (QP), this method is also computationally efficient but a dynamically changing environment might deteriorate the speed and may not be strictly enforced since the dynamics of the moving obstacles are not explicitly considered.

For control algorithms, a similar idea of using optimization is employed for tracking control under constraints [11, 1, 3, 14, 15]. One of the approaches is model predictive control (MPC), where all the constraints including system dynamics are directly imposed in the optimization scheme. For instance, [11] proposes a MPC controller which is based on mixed-integer linear programming (MILP). However, other optimization-based control methods focus on the system dynamics for simplification. By employing the system's structure, these control problems could be directly converted to a convex optimization problem.

Control Barrier Functions (CBFs), as proposed in [1], is a key concept for imposing dynamically-feasible constraints. The underlying idea is to enforce forward invariance of a set in the state space through a linear constraint on the control input. By combining control Lyapunov function(CLF) and a CBF candidate, A. Ames et al. [1] is able to propose an adaptive cruise controller based on an online QP. This CBF-CLF-QP controller is elegant but a general CBF construction method is absent since most constraints are imposed on the configuration variables only. Furthermore, time-varying safety constraint are not considered. G. Wu et al. [14] proposed a general CBF candidate, which is called geometric CBF, wherein position constraints on Riemmaninan manifolds are extended to the state space. Finally, work in [15] proposes an augmented CBF for the planar quadrotor, not only to account for the underactuated dynamics, but also to handle time-varying safety-critical constraints.

Our approach addresses the safety-critical tracking control of a 3D quadrotor, wherein, the goal is to track a desired trajectory while enforcing safety-critical constraints to prevent collisions while flying through a dense cluttered time-varying obstacle field with only local environmental information. Aerial flight through dense cluttered flight has been considered in the past [9, 5, 2], however, the obstacles considered are all static.

Our approach employs a geometric CBF and CLF to enforce strict non-collision safety constraints and ensure stability of tracking of nominal planned trajectories in the absence of conflicting safety constraints. In particular, if an obstacle is sensed directly on the desired path, the proposed controller automatically relaxes tracking of the trajectory to enforce strict safety. This enables the use of planners that provide fast and approximate plans with the controller directly accounting for obstacles that were unknown when the plan was made. See Figure 1. In comparison to previous work in [15, 14, 1], we make the following contributions:

- We consider the full 3D underactuated dynamics of the quadrotor evolving in $TSE(3)$.
- We explicitly consider the presence of time-varying obstacles and enforce that they can only be detected when within a limited sensing range of the quadrotor.
- We propose an augmented coordinate-free geometric CBF for the 3D quadrotor and use this with a geometric CLF expressed as a QP to realize safety-critical control.
- We show the safety guarantee for this controller in theory and test the performance with two numerical simulations comprising of aerial flight through dense cluttered (a) static obstacles, and (b) dynamic time-varying obstacles.

The rest of the paper is structured as follows. Section 2 presents mathematical preliminaries on geometric CLFs and CBFs. Section 3 presents the problem of constrained safety-critical control for a quadrotor UAV with range-limited sensing. Section 4 presents the construction of an augmented control Barrier function to convert position constraints to orientation constraints. Section 5 presents the main algorithmic result of this paper in the form of a sequential quadratic program. Section 6 presents simulation results on a quadrotor UAV. Section 7 presents concluding remarks and thoughts on future directions.

2 Mathematical Preliminary

This section introduces control Lyapunov functions (CLFs), control barrier functions (CBFs) and their corresponding geometric versions. These concepts are well studied in recent work [1, 14, 8]. Here, we will extend these concepts for the quadrotor control problem in particular.

$m \in \mathbb{R}$	Mass of the quadrotor
$J \in \mathbb{R}^{3\times 3}$	Inertia matrix of the quadrotor with respect to the body-fixed frame
$R \in SO(3)$	The rotation matrix of the quadrotor from the body-fixed frame to the inertial frame
$\Omega \in \mathbb{R}^3$	Angular velocity of the the quadrotor in the body-fixed frame
$x, v \in \mathbb{R}^3$	Position and velocity vectors of the center of mass of the quadrotor in the inertial frame
$f \in \mathbb{R}$	Thrust produced by the quadrotor
$M \in \mathbb{R}^3$	Moment produced by the quadrotor
$\hat{V}_x \in C^\infty(\mathbb{R}^6)$	Virtual control Lyapunov functions for position
$\hat{V}_R \in C^\infty(TSO(3))$	Geometric control Lyapunov function for orientation
$\mathscr{B}_{i,t} \subset \mathbb{R}^3$	A safe region for the quadrotor's position denoted based on safety concerns
$\mathscr{C}_{i,t} \subset \mathbb{T}SE(3)$	A safe region for the quadrotor's position and orientation which is constructed using $\mathscr{B}_i$
$g_i \in C^\infty(\mathbb{R}^3)$	The region function for $\mathscr{C}_i$, i.e, $\mathscr{C}_i = \{x \in \mathbb{R}^3 : g_i(x) \geq 0\}$ is the safety region in position space
$\hat{g}_i \in C^\infty(SE(3))$	Augmented region function including both position and orientation which preserves safety
$\hat{h}_i \in C^\infty(TSE(3))$	Expansion of the augmented region function $\hat{g}_i$ to the whole state space
$B_i \in C^\infty(TSE(3))$	Augmented control Barrier function based on $\hat{h}_i$ for safety
$\sigma \in C^\infty(\mathbb{R})$	Auxiliary function for the construction of $\hat{g}_i$

TABLE 1: Symbols used and their physical interpretation. Note that $C^\infty(X)$ denotes the set of all smooth scalar functions on X.

2.1 Exponentially-Stabilizing Control Lyapunov Function (ES-CLF)

Consider a control affine system shown below,

$$\dot{x} = \ f(x) + g(x)u, \quad x(t_0) = x_0, \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the states and control inputs, while $f(x), g(x)$ are the drift and control vector fields respectively. For system (1), a smooth function $V : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ is called exponentially stabilizing control Lyapunov Function (ES-CLF) if there exist constants $c_1, c_2, c_3 > 0$ such that $\forall t \in [0,\infty)$, $\forall x \in \mathbb{R}^n$

$$c_1||x||^2 \leq V(x) \leq c_2||x||^2,$$
$$\inf_{u \in \mathbb{R}^m} \{L_f V + L_g V u + c_3 V\} \leq 0.$$

where $L_f V = \partial V / \partial x \cdot f$, $L_g V = \partial V / \partial x \cdot g$ are the Lie derivatives of V with respect to f, g respectively.

To apply a CLF-based controller to a quadrotor, which evolves on a nonlinear manifold, we will extend the above Cartesian definition to the geometric setting next.

2.2 Geometric ES-CLF on SO(3)

The above concept of a ES-CLF is restricted to systems whose configuration space is Cartesian space. The concept of a geometric CLF is introduced to extend to a general nonlinear manifold in [14]. Here, we focus on ES-CLFs for $SO(3)$, where $SO(3)$ is a Lie group that represents the space of rotations. The attitude dynamics of the quadrotor is given by:

$$\dot{R} = R\hat{\Omega}, \qquad J\dot{\Omega} + \Omega \times J\Omega = M, \tag{2}$$

where $R \in SO(3), \Omega \in \mathbb{R}^3$ represent the orientation and angular velocity of the rigid body, and the hat map $\hat{\cdot} : R^3 \to \mathfrak{so}(3)$ is defined such that $\hat{x}y = x \times y$, $\forall x, y \in \mathbb{R}^3$. We also denote the vee map $\overset{\vee}{\cdot} : \mathfrak{so}(3) \to \mathbb{R}^3$ as its inverse, such that $(\hat{x})^\vee = x, \forall x \in \mathbb{R}^3$. Next, given a desired reference trajectory $R_d(t), \Omega_d(t)$, a geometric ES-CLF exists to stabilize the system in (2) to the above desired trajectory. From [15], it can be written as:

$$V = \alpha \Psi(R, R_d) + e_\Omega^T J e_\Omega + \varepsilon e_R \cdot e_\Omega \tag{3}$$

where $\alpha, \varepsilon > 0$, $\Psi(R, R_d) = \frac{1}{2}\text{trace}(I - R_d^T R)$ is the configuration error, $e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee$ is the position error and $e_\Omega = \Omega - R^T R_d \Omega$ is the velocity error.

This geometric ES-CLF (3) is based on geometric control theory [4] and is shown to be an almost global CLF in [15]. Since it's an almost global CLF, the set when it fails to be a CLF has measure 0, and thus we can design tracking controllers using this as a CLF for almost all states.

2.3 Time-varying Control Barrier Function (CBF)

A CBF is defined with respect to a region in the state space, which might be potentially time-varying. For system (1), suppose we have a smooth time-varying function $h : [0,\infty) \times \mathbb{R}^n \rightarrow \mathbb{R}$, with its super level set $\mathscr{C}_t = \{x \in \mathbb{R}^n : h(t,x) \geq 0\}$ representing a time-varying safe-set, i.e, we require the controller to keep the state within $\mathscr{C}_t$. If this set always has a non-empty interior, i.e., $\mathscr{C}_t^\circ \neq \emptyset$, for any time $t \in [0,\infty)$, then a smooth scalar function $B : \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty, -\infty\}$ is called a CBF of the set $\mathscr{C}_t$ if there exist two class $\mathscr{K}$ function α_1, α_2 and $\eta > 0$ such that

$$\begin{aligned}
&\frac{1}{\alpha_1(h(t,x))} \leq B(h(t,x)) \leq \frac{1}{\alpha_2(h(t,x))}, \\
&\inf_{u \in \mathbb{R}^m} \{B'(h)(\frac{\partial h}{\partial t} + L_f h + L_g h \cdot u) - \frac{\eta}{B}\} \leq 0, \qquad (4)
\end{aligned}$$

for any $t \in [0,\infty)$ and $x \in \mathscr{C}_t^\circ$ where $L_f h, L_g h$ are the Lie derivatives of h with respect to f and g.

Note that this time-varying CBF condition in (4) is proposed in [14] and can enforce the forward invariance of the safe set $\mathscr{C}_t$. Thus, once the system state starts within $\mathscr{C}_t$, the state will always remain within $\mathscr{C}_t$ as long as(4) is enforced for all time.

Having set up the necessary mathematical background, we will next formally define the safety-critical control problem for a 3D quadrotor with range-limited sensing and construct an augmented CBF in Sec. 4.

3 Safety-Critical Control Problem for 3D Quadrotor with Range-Limited Sensing

Consider a 3D quadrotor system shown in Fig. 2. The dynamics of this quadrotor is given by:

$$\begin{aligned}
\dot{x} &= v, \qquad & m\dot{v} &= FRe_3 - mge_3, \\
\dot{R} &= R\hat{\Omega}, \qquad & J\dot{\Omega} + \Omega \times J\Omega &= M,
\end{aligned} \qquad (5)$$

where $x \in \mathbb{R}^3$ is the position of the quadrotor's CoM, $R \in SO(3)$ is the rotation matrix of the body-fixed frame, g is the gravity acceleration, $m > 0, J \in \mathbb{R}^{3\times3}$ are the mass and inertia of the quadrotor, and $F \in \mathbb{R}, M \in \mathbb{R}^3$ are the scalar thrust force along the body axis and moment in the body-fixed frame respectively.

The safety-critical control problem for the 3D quadrotor is stated next. First, we assume the following are given:

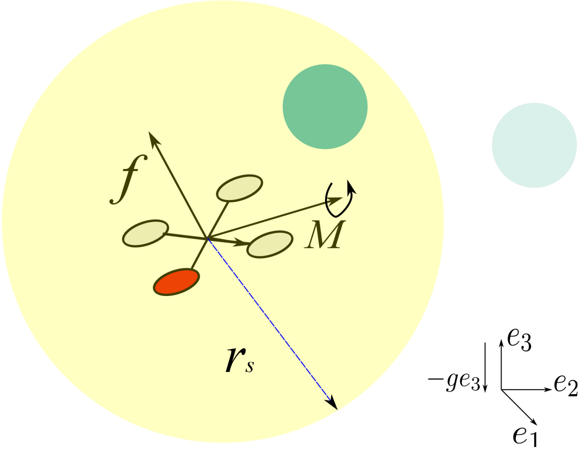


FIGURE 2: Thrust f and moment M produced by a 3D quadrotor, along with a range-limited omnidirectional sensor model with range r_s are shown. Two obstacles are shown (green circles), with the detected obstacle that is within the sensing range highlighted.

A.1 A smooth reference trajectory $(x_d(t), R_d(t))$ for (5), with $x_d \in \mathbb{R}^3$ and $R_d \in SO(3)$ representing the desired position and orientation of the quadrotor.

A.2 A list of time-varying safe regions in the position configuration space, $\mathscr{B}_{i,t} = \{x : g_i(t,x) \geq 0\}$, where x is the position of the quadrotor and $g_i(t,x) = ||x - x_i(t)||^2 - b_i(t),\ b_i(t) > 0, i = 1,2,\cdots,k$, such that $\mathscr{B}_i$ defines the space outside a closed ball centered about $x_i(t) \in \mathbb{R}^3$ of radius $\sqrt{b_i(t)}$, with the overall safe region in position configuration-space denoted as $\mathscr{B}_t = \cap_{i=1}^m \mathscr{B}_{i,t}$ with non-empty interior.

A.3 A finite limited sensing range r_s exists, such that the obstacle $\mathscr{B}_{i,t}$ is detected whenever $||x - x_i(t)|| \leq r_s$.

***Remark* 1.** *Note that in A.1, since the system* (5) *is differentially flat [7], we are able to obtain a smooth reference trajectory easily by choosing a set of flat outputs. Also, note that in A.3, a sensor such as lidar, an omnidirectional camera, 3D IR proximity sensor, or even multiple pairs of stereo vision can provide a full* 360° *sensing.*

Constrained Control Problem: The control goal is then to design a feedback law for the inputs f, M for the system (5) such that the following constraints are satisfied:

$$\begin{aligned}
&(x,R) \rightarrow (x_d, R_d), \text{ when } x_d \in \mathscr{B}_t, \text{as } t \rightarrow \infty && \text{(Stability constraint)} \\
&x(t) \in \mathscr{B}_t, \forall t \in [0,\infty) && \text{(Safety constraint)} \\
&0 \leq F \leq F_{\max},\ \ |M_j| \leq M_{j,\max},\ \ j = 1,2,3 && \text{(Input constraint)}
\end{aligned}$$

***Remark* 2.** *Note that this is a challenging control problem for several reasons:*

1. *We are considering the 3D quadrotor dynamics without typical local Euler angle parametrizations, requiring the control to be directly computed on manifolds.*
2. *We can not assume the desired trajectory is safe, requiring the controller to enforce strict safety constraints even when the desired trajectory violates them.*
3. *Since the safety constraints are in terms of position constraints and the quadrotor can not arbitrarily change it's position (due to underactuation), respecting these position safety constraints is non-trivial.*
4. *The time-varying safety constraints and limited sensing range introduce additional challenges.*
5. *Requiring strict enforcement of actuator limits in addition to all the above challenges makes this a hard control problem.*

Having formulated the safety-critical control problem for 3D quadrotor with range-limited sensing, we will next see how the position-based safety constraints defined by the set $\mathscr{B}_{i,t}$ are extended to all of the configuration through the construction of an augmented CBF.

4 Augmented Geometric CBF Construction

The quadrotor UAV is an underactuated system with 6 degrees of freedom and only 4 control inputs. Typically the attitude is controlled by the moment inputs while the position is controlled through adjusting the attitude. Thus, the controller has to consider both orientation and position at the same time to avoid an obstacle (see Fig. 3.) Since the thrust can only be positive, larger thrust can help Quadrotor Q_2 escape the obstacle while it would deteriorate the situation for Quadrotor Q_1. Hence we need a larger safety margin for Quadrotor Q_1 so that it can adjust its orientation before colliding with the obstacle. To capture this, we augment the definition of $g_i(t,x)$ to depend on the orientation of the quadrotor as well to construct the augmented safety function $\hat{g}_i(t,x,R)$. The intuition is to actively adjust the size of the safe set based on the quadrotor's orientation and it's ability to prevent exiting the safe set. In the following, we present a CBF construction method for a 3D quadrotor system (see [15] for a detailed discussion for the case of a planar quadrotor.)

We will illustrate the construction of the augmented CBF for a 3D quadrotor through a simple example. Given a safety region for position, $\mathscr{B} = \{x \in \mathbb{R}^3 : g(x) \geq 0\}$, determined by an implicitly time-varying smooth configuration safety function $g(x)$ below(we've dropped the explicit dependence of time on x, x_0, b for simplicity):

$$g(x) = ||x - x_o||^2 - b \geq 0, \quad b > 0,$$

we expand it to the whole configuration space by defining the augmented configuration safety function,

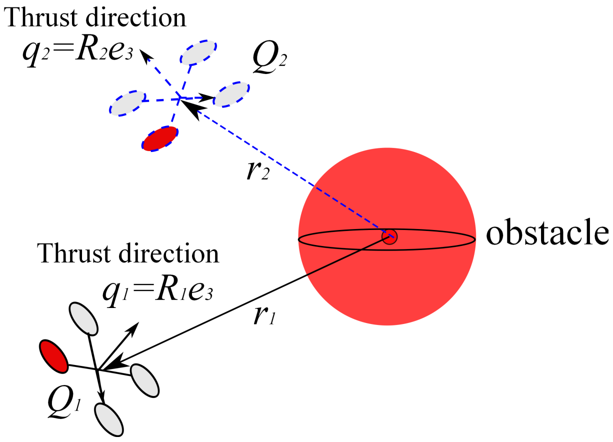


FIGURE 3: The ability of a quadrotor to avoid an obstacle depends both on its position and orientation. In particular, the Quadrotor Q_1 is less capable of avoiding the obstacle than Quadrotor Q_2. This can be captured through $s_1 := r_1 \cdot R_1 e_3 < 0 < s_2 := r_2 \cdot R_2 e_3$.

$$\hat{g}(x,R) := ||x - x_o||^2 - \beta b - \sigma(s), \tag{6}$$

where $s = r \cdot q$, $r = x - x_o$, $q = Re_3$, $\beta > 1$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a smooth function satisfying the following conditions:

$$\begin{aligned} &\sigma'(s) < 0, && \text{(Strictly decreasing)} \quad (7)\\ &|\sigma(s)| \leq \sigma_0 < (\beta - 1)b, \ \forall s \in \mathbb{R}, && \text{(Boundedness)} \quad (8)\\ &2s - \sigma'(s) > \ 0, \forall s \in (-\sqrt{\beta} b, 0), && \text{(Local quadratic)} \quad (9) \end{aligned}$$

***Remark* 3.** *Note that s is the inner product of the direction of the quadrotor thrust and the distance vector between the quadrotor and the center of the unsafe set $\mathbb{R}^3 \backslash \mathscr{B}$. Intuitively, it reflects the quadrotor's ability to avoid a certain obstacle as shown in Fig. 3.*

***Lemma* 1.** *If the system state always stays within $\hat{\mathscr{B}} = \{(x,R) \in SE(3) : \hat{g} \geq 0\}$, then the position state x would always remain within $\mathscr{B}$.*

Proof. It is obvious that whenever $(x,R) \in \hat{\mathscr{B}}$, $||x - x_o|| \geq \beta b - \sigma(s) \geq \beta b - \sigma_0 > \beta b - (\beta - 1)b = b$, based on (8). This implies that $x \in \mathscr{B}$, which guarantees safety in the position space.

The augmented safety function $\hat{g}$ above is a function of the configuration space and does not have relative degree 1. To address this we construct a new safety constraint function

$\hat{h} : TSE(3) \to \mathbb{R}$ as:

$$\begin{aligned}\hat{h} &:= \gamma\alpha(\hat{g}) + \dot{\hat{g}} \\ &= \gamma\alpha(\hat{g}) + 2(\dot{x} - \dot{x}_o)\cdot(x - x_o) - \beta\dot{b} - \sigma'(s)\dot{s},\end{aligned} \tag{10}$$

where $\dot{s} = (\dot{x} - \dot{x}_o)\cdot Re_3 + (x - x_o)\cdot R(\Omega \times e_3)$. We can also define a new safety region in the state space as

$$\mathscr{C} = \{(x, R, \dot{x}, \Omega) \in TSE(3) : \hat{h}(x, R, \dot{x}, \Omega) \geq 0\}.$$

***Lemma* 2.** *If the safety region $\mathscr{C}$ is forward invariant, then the safety region $\mathscr{B}$ is also forward invariant.*

Proof. The proof follows from [14, Prop. 1] and Lemma 1. Suppose $(x(0), R(0), \dot{x}(0), \Omega(0)) \in \mathscr{C}$, then by forward invariance of $\mathscr{C}$, $(x(t), R(t), \dot{x}(t), \Omega(t)) \in \mathscr{C}, \forall t \geq 0$, i.e, $\hat{h}(x(t), R(t), \dot{x}(t), \Omega(t)) = \gamma\alpha(\hat{g}(x(t), R(t))) + \dot{\hat{g}}(x(t), R(t), \dot{x}(t), \Omega(t)) \geq 0$.

Then we could proceed through contradiction. Assume the configuration variable at time $t_2 > 0$ lies outside the region $\hat{\mathscr{B}}$. Then since the function $\hat{g}(x, R)$ is smooth, by intermediate value theorem, there exists $0 < t_1 < t_2$ such that $\hat{g}(x(t_1), R(t_1)) = 0$, $\dot{\hat{g}}(t_1) < 0$, or equivalently, the state would escape $\mathscr{B}$ at t_1. However, due to the above inequality, a contradiction arises since $\dot{\hat{g}}(t_1) \geq -\alpha(\hat{g}(x(t_1), R(t_1))) = 0$. Thus the assumption is not true, which implies that the region $\hat{\mathscr{B}}$ is also forward invariant. Applying Lemma 1 yields that the region $\mathscr{B}$ is forward invariant.

Then select a candidate CBF $\hat{B} := 1/\hat{h}^a$ with $a > 0$. We have the following lemma regarding the safety:

***Lemma* 3.** *If the scalar function σ satisfies the conditions* (7), (8) *and* (9). *Then the candidate function $\hat{B}$ is an almost global CBF for* (5). *Moreover, we can guarantee safety for the trajectory of* (5), *provided that $\hat{h}(0) \geq 0$ and the thresholds $F_{\max}, M_{i,\max}$ are sufficiently large.*

Proof. Assume $F_{\max}, M_{i,\max}$ are unbounded. Then it holds that

$$\begin{aligned}\dot{B} - \frac{\gamma}{B} \leq 0 &\Leftrightarrow -\frac{1}{\hat{h}^{a+1}}(\dot{\hat{h}} + \gamma\hat{h}^{2a+1}) \leq 0, \\ &\Leftrightarrow \dot{\hat{h}} + \gamma\hat{h}^{2a+1} \geq 0.\end{aligned} \tag{11}$$

Note that the time derivative of $\hat{h}$ defined in (10) can be written as

$$\begin{aligned}\dot{\hat{h}} &= \gamma\alpha'(\hat{g})\dot{\hat{g}} + 2(x - x_o)\cdot(\ddot{x} - \ddot{x}_o) \\ &+ 2(\dot{x} - \dot{x}_o)\cdot(\dot{x} - \dot{x}_o) - \beta\ddot{b} - \sigma''(s)\dot{s}^2 \\ &- \sigma'(s)[(Re_3)\cdot(\ddot{x} - \ddot{x}_o) + 2(\dot{x} - \dot{x}_o)\cdot R(\Omega \times e_3) \\ &+ (x - x_o)^T R\hat{\Omega}^2 e_3 - (x - x_o)^T R\hat{e}_3\dot{\Omega}] \\ &= \frac{1}{m}(2s - \sigma'(s))F + (\sigma'(s)(R^T r)^T \hat{e}_3 J^{-1})M + \Gamma_3 \\ &= \Gamma_1 F + \Gamma_2^T M + \Gamma_3\end{aligned} \tag{12}$$

where the term Γ_3 is independent of $\ddot{x}$ and $\dot{\Omega}$ and Γ_1, Γ_2 are the coefficients of f and M respectively.

We can check several cases depending on the term Γ_3. If $\Gamma_3 \neq 0$, then we could assign the following control inputs:

$$F = 0, \quad M = \frac{-\Gamma_3 - \gamma\hat{h}^{2a+1}}{||\Gamma_2||^2} J\Gamma_2$$

which means that we could always avoid the obstacle by adjusting the altitude for this case.

When $\Gamma_2 = 0$, it holds that $r\hat{R}e_3 = 0$ since $\sigma'(s) < 0$ according to (7) and the matrix J is nonsingular. Further, it holds that $e_3 \times R^T r = 0$ which means that the vectors q and r are parallel. For the case when the directions of q and r coincide, we have $\Gamma_1 = 2s - \sigma'(s) > 2s = 2r \cdot q > 0$ using condition (7). This means that we could apply a large enough thrust to satisfy CBF condition as:

$$F = m\frac{-\Gamma_3 - \gamma\hat{h}^{2a+1}}{2s - \sigma'(s)}, \quad M = 0$$

When the directions q and r are contrary to each other, whether the CBF condition holds depends on the range of s. The only condition when CBF would fail is that $s < -\sqrt{\beta}b$. Fortunately, the set $\{(x, R, \dot{x}, \Omega) \in TSE(3) : s \leq -\sqrt{\beta}b\}$ is a submanifold in $TSE(3)$ and has Lebesgue measure zero. So B is an almost global CBF. Consider the case when CBF condition fails, we have that $s \leq -\sqrt{\beta}b$ which means that $-s = -q \cdot r = ||r|| \geq \sqrt{\beta}b$ by condition (9). This means that $g = ||r||^2 \geq \beta b > b$, and thus the system trajectory would remain within $\mathscr{B}$. When the condition of CBF is satisfied, the region $\mathscr{C}$ is forward invariant. By *Lemma* 2, we have the original safety region $\mathscr{B}$ is forward invariant. Combining the previous argument, the system trajectory will always remain within $\mathscr{B}$ for either case and thus stay safe.

***Remark* 4.** *Note that in the proof, we have not considered input saturation. In particular, if the input constraints are too stringent, its possible for the QP to become infeasible. If we assume* *the relative velocity of the obstacle entering the sensing range is bounded, then we can choose the constant γ and the function σ to still guarantee safety. In general, incorporating input saturation into the construction of a Barrier is an open research problem.*

5 Sequential CBF-CLF-QP Control with Limited Sensing Range

Based on the augmented CBF constructed in Sec. 4, we propose a cascade optimization scheme for the CBF-CLF-QP control design. The underlying idea is inspired by the backstepping method in geometric control [12], which makes a singular perturbation argument to separate the fast orientation dynamics from the slow translational dynamics. Similar to this, the scheme here comprises of two levels: the first level is called *position level QP* and the second level is called *orientation level QP*.

First, we construct an augmented CBF $\hat{B}_i$ for each safety region $\mathscr{B}_{i,t}$ as indicated in Eq.(6). The corresponding expanded safe set is $\mathscr{C}_{i,t} = \{(x,\dot{x},R,\Omega) \in TSE(3) : \hat{h}_i \geq 0\}$. Then assume that the underactuated part is "fully-actuated" with the virtual dynamics:

$$v = \dot{x}, \quad m\dot{v} = f \tag{13}$$

Select a quadratic CLF for this virtual system as:

$$\hat{V}_x = \frac{1}{2} m e_v \cdot e_v + \frac{1}{2} k_1 e_x \cdot e_x + \varepsilon_1 e_x \cdot e_v \tag{14}$$

where $e_x = x - x_d, e_v = v - v_d$, and the value of $k_1, \varepsilon_1 > 0$ are chosen specifically to make $\hat{V}$ quadratic.

Then, we are able to compute a virtual force based on $\hat{V}$ through the following QP based on (14):

Position Level QP (virtual force computation)

$$\begin{aligned} f^* = \underset{f \in \mathbb{R}}{\operatorname{argmin}} \ & \frac{1}{2} f^T Q f \\ \text{s. t. } & \dot{\hat{V}}_x(f) + \eta_1 \hat{V}_x \leq 0, \end{aligned}$$

---

where $\eta_1 > 0$. The solution f^* is computed as a *virtual force* and passed onto the lower orientation level of optimization as an input parameter,

Then we decompose this input f^* into its norm $F_c = |f^*|$ and direction $b_{3c} = f^*/F_c$. Then compute a desired rotation matrix as below: $R_c = [b_{1c}, b_{3c} \times b_{1c}, b_{3c}]$ where the unit vector $b_{1c} = -\hat{b}_{3c}^2 \times R_d e_1 / ||\hat{b}_{3c}^2 \times R_d e_1||$. In this way, we can construct a geometric CLF for the orientation part as:

$$\hat{V}_R = \frac{1}{2} e_R^T J e_R + \frac{1}{2} e_\Omega^T K_2 e_\Omega + \varepsilon_2 e_R \cdot e_\Omega \tag{15}$$

Then, the orientation level QP is constructed to obtain our actual control inputs F and M:

Orientation Level QP (virtual force tracking and safety guarantee):

$$\begin{aligned} [F^*, M^*] = \underset{F \in \mathbb{R}, M \in \mathbb{R}^3}{\operatorname{argmin}} \ & \frac{1}{2} \lambda_1 (F - F_c)^2 + \frac{1}{2} M^2 + \frac{1}{2} \lambda_2 \delta^2 \\ \text{subject to } \ & \dot{\hat{V}}_R(M) + \eta_2 \hat{V}_R \leq \delta \\ & \dot{\hat{B}}_j(F,M) \leq \frac{\gamma_j}{\hat{B}_j}, \ j \in I_s(t) \end{aligned}$$

---

where $I_s(t)$ is the collection of indices corresponding to obstacles that are detected by the quadrotor's onboard range-limited sensor, $\dot{\hat{B}}_j(F,M)$ is as computed in (11) and $\lambda_1, \lambda_2, \eta_2, \gamma_j$ are all positive gain parameters.

***Remark* 5.** *Based on analysis in [6], the stability of a geometric controller can be guaranteed roughly under the condition of a fairly large proportional gain in the orientation control. Since the convergence rate of CLF can be related to the proportional gain, we here entail that $\eta_2 \gg \eta_1$. This can be shown in the simulation parameters.*

Regarding the safety property of this controller, we have the following proposition:

***Proposition* 1.** *(Safety Guarantee of Sequential CBF-CLF-QP Controller)*

If the following assumptions are satisfied:

1. *Suppose at any time $t \in [0,\infty)$, the sequential QP controller always admits a solution F, M;*
2. *The conditions of Lemma 3 are satisfied;*
3. *There exists sequence $\{t_n, \ n \in \mathbb{N}\}$ such that I_s is constant between $[t_n, t_{n+1}])$ for each $n \in \mathbb{N}$, and $\hat{B}_i(t_n) \geq 0$ for all $i \in I_s$;*

then the system trajectory would always remain within $\mathscr{C}_t$.

Proof. First observe that only those obstacles within the sensing range will affect the trajectory's safety at each time $t > 0$. Fix $n \in \mathbb{N}$, consider the time interval $[t_n, t_{n+1}]$. By applying Lemma 3 to each function $\hat{B}_i$, the condition of every CBF can be enforced, and thus the set $\mathscr{C}_{i,t}$ is forward invariant for each $i \in I_s$. Thus, the system trajectory would remain within every region $\mathscr{C}_{i,t}$, and the position trajectory would remain within $\mathscr{B}_{i,t}$ by Lemma 2 for each $i \in I_s$. This indicates the system position would remain within $\mathscr{B}_t$ for $[t_n, t_{n+1})$ by the previous argument. Hence, over the entire interval $[0, \infty)$, the system trajectory would always remain within $\mathscr{C}_t$ and thus stay safe.

6 Simulation Results

To numerically validate the performance of the proposed sequential QP controller, we have created a simulation framework in Matlab 2015b. The simulator utilizes ode15s as the solver since the problem is intrinsically stiff due to new obstacles being sensed. Each level of QP is solved by "interior point method" with convergence tolerances 10^{-4} and 10^{-6} respectively. The quadrotor model is an Asctec Hummingbird, with a two meter sensing range. The mass parameters of this quadrotor are provided by the company as $m = 0.52kg, J = \text{diag}[2.32, 2.32, 7.60] \times 10^{-3} kg \cdot m^2$. The function σ is given as $\sigma(s) = -a_1 \arctan(a_2 s + a_3)$. As mentioned in Section. 3, we will choose the positions x_d and yaw angle ϕ_d as the flat outputs to generate the reference $(x_d(t), R_d(t))$. Given the above reference trajectory, we then solve the control problem as described in Section 3 of tracking the reference asymptotically when ever its safe while ensuring the state is in the safe set by avoiding obstacles and enforcing input constraints. We use the position and orientation level QP controllers described in Section 5, however we incorporate the input bounds, $0 \le f \le 50N$, and $|M| \le 0.038Nm$. With this controller, we next present the results of two simulation experiments, where the quadrotor has to detect and avoid randomly generated static and dynamic obstacles respectively.

6.1 Trajectory Tracking with Randomly Generated Static Obstacles

In the first experiment, the reference of the flat outputs is given as a straight line $x_d(t) = [0, 0.75t, 1.5]^T$, $\phi_d(t) = 0$. This straight line will pass through a box $[-2.5, 2.5] \times [0, 10] \times [0, 5]$ where we put in randomly generated sphere-shaped obstacles. These obstacles are generated offline from uniform distribution. The positions of these obstacles are uniformly sampled within the box while the corresponding radii are taken from an uniform distribution over the interval $[0.25, 1]$. We show the simulation results graphically in Fig. 4. As shown, initially we plot out 40 obstacles to show the general setup in Fig. 4a. The rest of figures plot out the reference and actual trajectory of the quadrotor's CoM in red dashed and black solid line separately. As can be seen, the CBF-CLF-QP controller will help avoid the obstacles while tracking the reference trajectory when it's safe.

6.2 Trajectory Tracking with Randomly Generated Dynamic Obstacles

In the second experiment, the corresponding flat output is $x_d(t) = [-\sin(0.5t), 5 + \cos(0.5t), 1.5]^T$, $\phi_d(t) = \frac{\pi}{6}$ which is a planar circle parallel to the XY plane. During the simulation process, we randomly generate 12 obstacles near the wall of a box and shoot them at the quadrotor with speed $1.5m/s$ every 2 seconds. The simulation results are shown in Fig. 5 and Fig.6. Fig. 5 shows the error plot in position and orientation with respect to reference. Since the obstacle will be constantly shot at the quadrotor, asymptotic stability for the trajectory tracking is only possible when there is no obstacle along the trajectory. The controller attempts to track the reference in a stable manner when the reference trajectory is safe, while the controller relaxes tracking the reference when it is no longer safe. This leads to the error to fluctuate from converging to zero to going to non-zero values. Fig. 6 also shows snapshots of the reference and actual trajectory, using the same line pattern as in Fig. 4. In the static case, only a few obstacles are sensed, however for the dynamic case, a lot more obstacles are sensed since the obstacles are shot towards the quadrotor. Due to the presence of numerous dynamic obstacles, the quadrotor has to constantly oscillate around the reference to avoid collision. Check the video link in the caption for a better illustration.

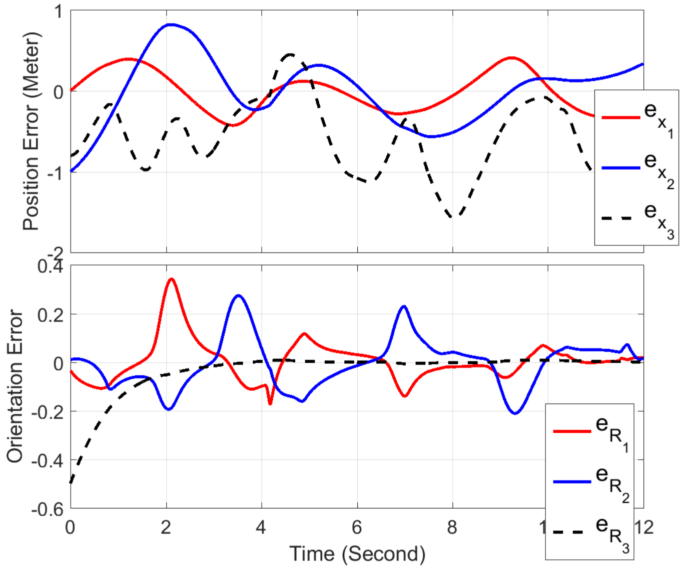
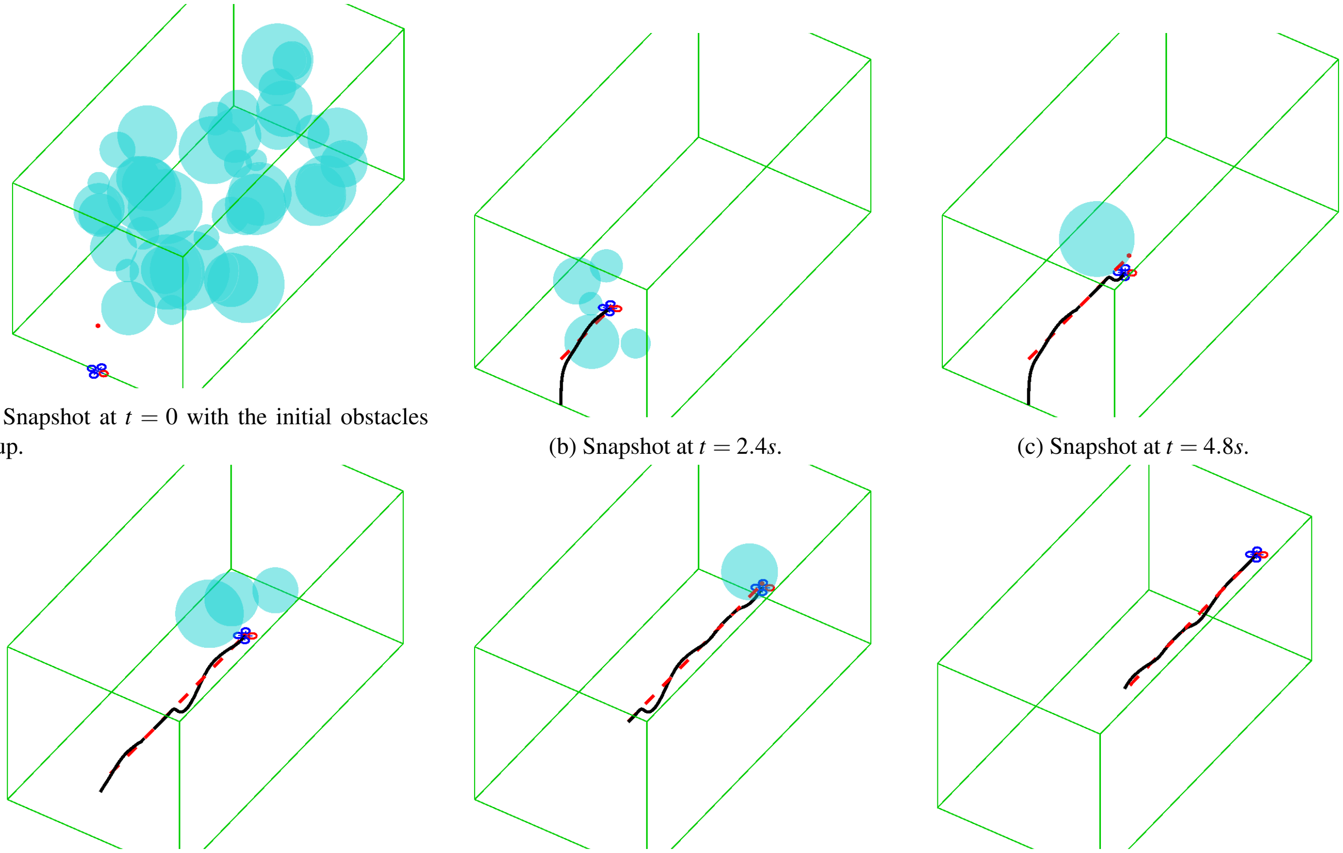


FIGURE 5: Position error e_x and orientation error e_R plot. The controller drives the error to zero when the reference trajectory is safe. When the reference trajectory violates the safety constraint (by passing through an obstacle), the controller automatically relaxes trajectory tracking to strictly enforce safety constraints. All this is done without the need for re-planning.



(a) Snapshot at $t = 0$ with the initial obstacles setup.

(b) Snapshot at $t = 2.4s$.

(c) Snapshot at $t = 4.8s$.

(d) Snapshot at $t = 7.2$.

(e) Snapshot at $t = 9.6s$.

(f) Snapshot at $t = 12s$.

FIGURE 4: Numerical validation of 3D quadrotor flight through a dense cluttered static obstacle field. Snapshots of the simulation for 15 seconds, where we show all the obstacles in the first snapshot and only plot out the obstacles that are detected by the quadrotor in subsequent snapshots. The black solid line is the actual trajectory and the red dashed line is the reference trajectory. The quadrotor strictly guarantees non-collision with the obstacles and tracks the desired reference trajectory when feasible, all without the need for any re-planning. Simulation video: `https://youtu.be/LHNesE603us`.

6.3 Discussion

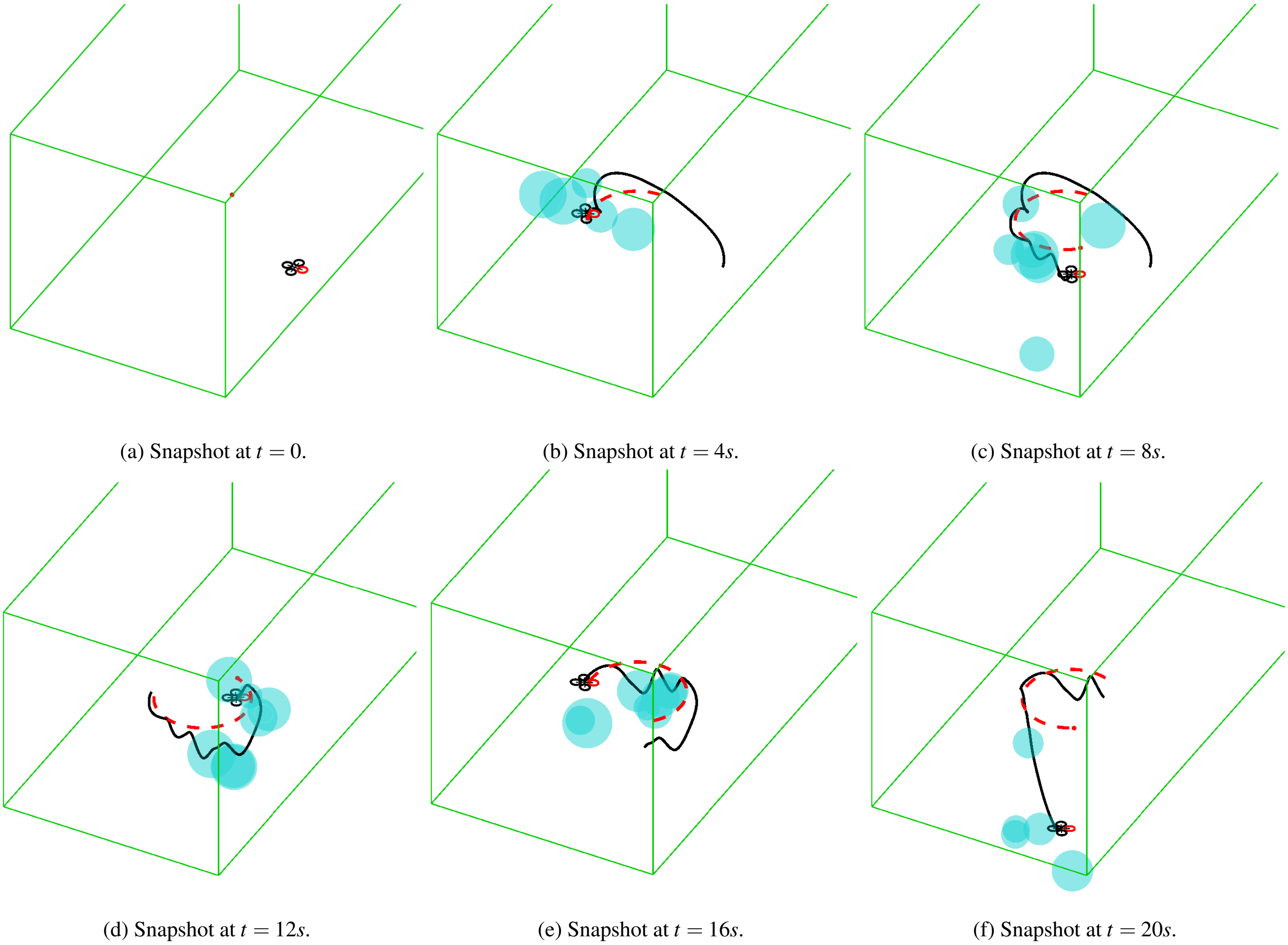
In addition to the promising simulation results presented in Figs. 4,6, we also want to add in a few comments regarding advantages and drawbacks of our controller. In particular, although our controller can adapt to the changing environment rapidly to avoid obstacles while tracking a reference trajectory, there is a possibility that it can get trapped in nonconvex regions as it's only using local information, as described in [13]. However, this can be easily addressed by combining this controller with a long range planner, thereby enabling safe and efficient flight. As mentioned in Remark 4, there is a possibility that the QP becomes infeasible for extreme situations with stringent torque saturation constraints. However, bounding the relative velocity of the obstacle on entering the sensing field and carefully selecting the constant γ and the function σ, in practice we can still retain the guarantees of safety. Finally, the presented σ function was chosen arbitrary to satisfy certain properties. We believe that finding a systematic way to search for a σ function would be an interesting future direction.

Acknowledgment

This work is supported in part by the Google Faculty Research Award and in part by NSF grants CMMI-1538869, IIS-1464337, IIS-1526515.

7 Conclusion

In this paper, we address the constrained control problem of a 3D quadrotor with limited sensing range navigating a dense obstacle field. The control design is proposed as a sequential QP controller that uses geometric CLFs and an augmented CBF.



(a) Snapshot at $t = 0$.

(b) Snapshot at $t = 4s$.

(c) Snapshot at $t = 8s$.

(d) Snapshot at $t = 12s$.

(e) Snapshot at $t = 16s$.

(f) Snapshot at $t = 20s$.

FIGURE 6: Numerical validation of 3D quadrotor flight through a dense cluttered dynamic time-varying obstacle field. Snapshots of the simulation process for 12 seconds are shown. The red dashed line is the reference trajectory while the black solid line is the actual trajectory of the quadrotor's CoM. The obstacles are dynamic and move with randomly generated velocities. Simulation video: `https://youtu.be/LHNesE603us`.

Alongside with rigorous proof, we show the safety guarantee through two numerical experiments of aerial flight through a dense cluttered static and dynamic obstacle fields.

REFERENCES

[1] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 6271–6278.

[2] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[3] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," in *IFAC Conference on Analysis and Design of Hybrid Systems*, 2015.

[4] F. Bullo, *Geometric control of mechanical systems*. Springer Science & Business Media, 2005, vol. 49.

[5] S. Karaman and E. Frazzoli, "High-speed flight in an ergodic forest," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2899–2906.

[6] T. Lee, M. Leok, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE (3)," *IEEE Conference*

on Decision and Control, pp. 5420–5425, 2010.

[7] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[8] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.

[9] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, "Motion primitives and 3d path planning for fast flight through a forest," *The International Journal of Robotics Research*, p. 0278364914558017, 2015.

[10] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Proceedings of the International Symposium of Robotics Research*, 2013.

[11] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 6. IEEE, 2004, pp. 5576–5581.

[12] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *IEEE Conference on Decision and Control (CDC)*, Florence, Italy, Dec. 2013, pp. 2269–2274.

[13] M. Watterson and V. Kumar, "Safe receding horizon control for aggressive mav flight with limited range sensing," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3235–3240.

[14] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 2038–2044.

[15] ——, "Safety-critical control of a planar quadrotor," in *American Control Conference (ACC), 2016*. IEEE, submitted, 2016, pp. 2252–2258.