

# Control Strategies for Pursuit-Evasion Under Occlusion Using Visibility and Safety Barrier Functions

Minnan Zhou<sup>1,\*</sup>  
Shumon Koga<sup>3</sup>

Mustafa Shaikh<sup>1,\*</sup>  
Dimitra Panagou<sup>4</sup>

Vatsalya Chaubey<sup>1,\*</sup>  
Nikolay Atanasov<sup>1</sup>

Patrick Haggerty<sup>2</sup>  
Nikolay Atanasov<sup>1</sup>

**Abstract**— This paper develops a control strategy for pursuit-evasion problems in environments with occlusions. We address the challenge of a mobile pursuer keeping a mobile evader within its field of view (FoV) despite line-of-sight obstructions. The signed distance function (SDF) of the FoV is used to formulate visibility as a control barrier function (CBF) constraint on the pursuer’s control inputs. Similarly, obstacle avoidance is formulated as a CBF constraint based on the SDF of the obstacle set. While the visibility and safety CBFs are Lipschitz continuous, they are not differentiable everywhere, necessitating the use of generalized gradients. To achieve non-myopic pursuit, we generate reference control trajectories leading to evader visibility using a sampling-based kinodynamic planner. The pursuer then tracks this reference via convex optimization under the CBF constraints. We validate our approach in CARLA simulations and real-world robot experiments, demonstrating successful visibility maintenance using only onboard sensing, even under severe occlusions and dynamic evader movements.

## I. INTRODUCTION

Pursuit-evasion problems [1] are studied in computational geometry, control theory, and robotics, motivated by applications in search and rescue [2], security and surveillance [3], and environmental monitoring [4]. Introducing visibility constraints leads to the art gallery problem [5], which has elegant solutions with static pursuers (guards) in 2D polygonal environments but becomes challenging with mobile pursuers in 3D environments. Prior works tackled pursuit-evasion with visibility constraints by graph-theoretic [6] or game-theoretic approaches [7]–[9]. This work proposes a novel control design for pursuit-evasion with field-of-view (FoV) constraints, using control barrier function (CBF) techniques.

Originating from the pioneering works of Wieland and Allgöwer [10] and Ames et al. [11], [12], CBFs have become a key tool for safety-critical applications such as adaptive cruise control [13], robot manipulation [14], robot locomotion [15], and robot flight [16]. CBFs are defined so that the positivity of a barrier function implies forward invariance of a corresponding superlevel set for the trajectories of a

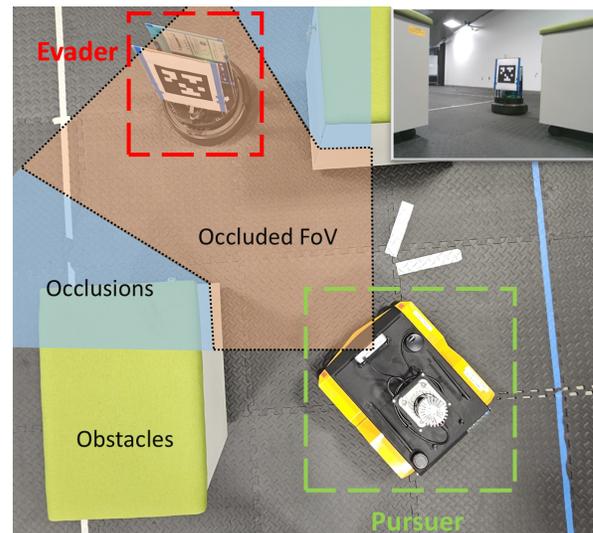


Fig. 1: A mobile *pursuer* aims to keep a mobile *evader* within its field of view despite occlusions and without colliding with obstacles. The inset shows the evader in the pursuer’s camera view.

dynamical system. A key observation is that, for control-affine dynamical systems, CBF constraints are *linear* in the control input. This allows control synthesis via quadratic programming (QP) subject to linear CBF constraints [12].

A challenge for CBF methods is to determine a function whose positivity is equivalent to the desired safe behavior. Previous works have demonstrated ways to validate candidate CBFs for polynomial control systems [17] subject to multiple state and input constraints [18], or even to learn CBFs from expert demonstrations [19]. A key observation in this paper is that visibility under occlusion can also be formulated as a CBF condition using the signed distance function (SDF) of the sensor FoV. An SDF provides the (signed) distance from any point in space to a given set’s boundary. SDFs have been used to model shapes and surfaces in vision and robotics tasks such as object shape reconstruction [20] and robot mapping [21]. In this paper, we introduce a novel *visibility CBF constraint* that enforces negativity of the SDF from an evader to the occluded FoV of the pursuer.

While early works on CBFs primarily focused on continuous systems subject to differentiable CBF constraints, recent advances [18], [22], [23] have studied the regularity of CBF-based controllers and have expanded their applicability to non-smooth dynamics and non-smooth constraints. These methods have been used in [24] to enable safe multi-robot navigation with polytope-shaped robots. In our case, the

This work was supported by ARL DCIST grant W911NF-17-2-0181.

\*These authors contributed equally.

<sup>1</sup>The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA (e-mails: {m9zhou, mushaikh, vchaubey, natanasov}@ucsd.edu).

<sup>2</sup>Patrick Haggerty is with General Dynamics Mission Systems, Bloomington, MN 55431, USA (e-mail: patrick.haggerty@gd-ms.com).

<sup>3</sup>Shumon Koga is with Honda R&D Co. Ltd., Tokyo, Japan (e-mail: shumon.koga@jp.honda).

<sup>4</sup>Dimitra Panagou is with the Department of Robotics and the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA (e-mail: dpanagou@umich.edu).

<https://existentialrobotics.org/VisibilityControl>

visibility CBF is Lipschitz continuous but not differentiable everywhere. It is also time-varying due to the pursuer's and evader's motion. Hence, it is necessary to use a non-smooth time-varying formulation of the visibility CBF constraint involving its generalized gradient.

Maintaining visibility is a requirement in various target tracking problems. A visibility-aware planner proposed in [25] generates sensor trajectories with maximal visibility to a moving target. A vision-based controller for a quadrotor landing on a ground vehicle, while preventing camera occlusions, was developed in [26]. Leader-follower formations for maintaining visibility and safety were developed using dipolar vector fields in a known environment in [27]. Similarly, in the presence of FoV constraints, dipolar vector fields have been used in [28] to formulate a model predictive control scheme for differential-drive robot navigation, maintaining visibility of a static landmark. Closely related to our work, Gao et al. [29] formulate a probabilistic notion of visibility under occlusion, which is maintained using an extended Kalman filter [30]. The authors develop a real-time non-myopic trajectory planner for visibility-aware and safe target tracking under uncertainty. This work is complementary to ours in that it focuses on high-level non-myopic visibility planning using nonlinear optimization and linear SDF approximations, while we focus on low-level myopic visibility control to track a planned reference using convex programming. The contributions of our work are summarized,

- We prove Lipschitz continuity of the signed distance to an occluded FoV to justify its use as a non-smooth time-varying CBF for visibility maintenance.
- We achieve non-myopic visibility maintenance by coupling a kinodynamic planner that generates reference controls leading to evader visibility, with a controller that tracks the planned reference using convex optimization subject to visibility and safety CBF constraints.
- We demonstrate successful visibility maintenance in real robot experiments using only onboard sensing, even under severe occlusions and dynamic evader motion.

## II. PROBLEM STATEMENT

Consider a mobile robot modeled as a system with state  $\mathbf{x} \in \mathbb{R}^n$ , control input  $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$ , and dynamics model:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + G(\mathbf{x}(t))\mathbf{u}(t), \quad (1)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $G : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$  are continuous functions. We refer to the robot as a *pursuer* because we are interested in controlling its motion to maintain visibility of an *evader*. The evader is modeled as a system with configuration  $\mathbf{y}(t) \in \mathbb{R}^p$  and velocity  $\dot{\mathbf{y}}(t) \in \mathbb{R}^p$ . The evader's velocity is assumed to be bounded, i.e.,  $\|\dot{\mathbf{y}}(t)\| \leq k$  for some  $k < \infty$ .

The pursuer is equipped with a sensor with *unoccluded* field of view (FoV), i.e., in the absence of obstacles, modeled by a closed set  $\mathcal{F}_0 \subset \mathbb{R}^p$ , determined by the onboard sensor ranges and parameters. The unoccluded FoV at a particular pursuer state  $\mathbf{x}$ , i.e., transformed to the global coordinate frame, is represented as a linear transformation,  $A(\mathbf{x})\mathcal{F}_0 + b(\mathbf{x})$ , where  $A : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times p}$  and  $b : \mathbb{R}^n \rightarrow \mathbb{R}^p$  are Lipschitz

continuous functions and the multiplication is defined for every element of  $\mathcal{F}_0$ . For example, if  $\mathbf{x}$  contains the position  $\mathbf{p}$  and orientation  $R$  of the pursuer, the transformed FoV would be  $R\mathcal{F}_0 + \mathbf{p}$ . We denote the *occluded* FoV, i.e., determined by obstacles in the environment depending on the pursuer's state  $\mathbf{x}$ , by  $\mathcal{F}(\mathbf{x}) \subseteq A(\mathbf{x})\mathcal{F}_0 + b(\mathbf{x})$ . The evader is *visible* to the pursuer at time  $t$  if  $\mathbf{y}(t) \in \mathcal{F}(\mathbf{x}(t))$ .

Our objective is to design a control policy for the pursuer to maintain visibility of the evader. In addition, the pursuer's state  $\mathbf{x}(t)$  should remain outside of an obstacle set  $\mathcal{O} \subset \mathbb{R}^n$  at all time. In summary, we consider the following problem.

**Problem.** Let  $\mathbf{x}(0) \notin \mathcal{O}$  be an initial condition for the pursuer system in (1). Given the evader state  $\mathbf{y}(t)$ ,  $\dot{\mathbf{y}}(t)$  for all  $t \geq 0$ , design a control policy for the pursuer that ensures  $\mathbf{x}(t) \notin \mathcal{O}$  for all  $t \geq 0$  and maximizes the duration of evader visibility  $\mathbf{y}(t) \in \mathcal{F}(\mathbf{x}(t))$ .

We assume that the evader state  $\mathbf{y}(t)$ ,  $\dot{\mathbf{y}}(t)$  is known at time  $t$  and focus on pursuer motion planning and control instead of on evader motion estimation. In practice, the evader state can be estimated using onboard sensing, e.g., via visual tracking [31] for a camera-equipped pursuer.

## III. TECHNICAL APPROACH

Our key idea is to encode the requirements for the pursuer to maintain visibility and safety as CBF constraints, and synthesize pursuer control inputs subject to these constraints. We begin with a review of the concept of CBF.

### A. Control Barrier Functions

In safety-critical applications, it is desirable to control a dynamical system to ensure its state remains in a safe set:

$$\mathcal{S}(t) = \{\mathbf{x} \in \mathbb{R}^n \mid h(t, \mathbf{x}) \geq 0\}, \quad (2)$$

defined as the superlevel set of a continuously differentiable function  $h : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$ .

**Definition 1** ([32]). A function  $h : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a *time-varying control barrier function* (CBF) on set  $\mathcal{D}$ , if  $\mathcal{S}(t)$  in (2) satisfies  $\mathcal{S}(t) \subset \mathcal{D} \subseteq \mathbb{R}^n$  and there exists an extended class  $\mathcal{K}$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  such that

$$\sup_{\mathbf{u} \in \mathcal{U}} \left\{ \mathcal{L}_f h(t, \mathbf{x}) + \mathcal{L}_G h(t, \mathbf{x})\mathbf{u} + \frac{\partial h}{\partial t}(t, \mathbf{x}) \right\} \geq -\alpha(h(t, \mathbf{x})),$$

for all  $\mathbf{x} \in \mathcal{D}$  and for all  $t \geq 0$ . The terms  $\mathcal{L}_f h(t, \mathbf{x}) = \nabla_{\mathbf{x}} h(t, \mathbf{x})^\top f(\mathbf{x})$  and  $\mathcal{L}_G h(t, \mathbf{x}) = \nabla_{\mathbf{x}} h(t, \mathbf{x})^\top G(\mathbf{x})$  are the Lie derivatives of  $h(t, \mathbf{x})$  along the vector fields  $f, G$  in (1).

When  $\mathcal{S}(t)$  is defined by a CBF, it can be guaranteed that the system state  $\mathbf{x}(t)$  remains within  $\mathcal{S}(t)$  (invariance) using a CBF constraint on the system control inputs.

**Lemma 1** ([32]). *Let  $h : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$  be a time-varying CBF defining a safe set (2). Then, any Lipschitz continuous control policy  $\mathbf{u} = \pi(t, \mathbf{x})$  for the system in (1) satisfying*

$$\mathcal{L}_f h(t, \mathbf{x}) + \mathcal{L}_G h(t, \mathbf{x})\pi(t, \mathbf{x}) + \frac{\partial h}{\partial t}(t, \mathbf{x}) \geq -\alpha(h(t, \mathbf{x})), \quad (3)$$

*for all  $\mathbf{x} \in \mathcal{S}(t)$  and for all  $t \geq 0$ , renders  $\mathcal{S}(t)$  invariant.*

For a locally Lipschitz reference controller  $r(t, \mathbf{x})$ , the quadratic program below with CBF safety constraint provides the minimum perturbation to  $r(t, \mathbf{x})$  to guarantee safety [12]:

$$\begin{aligned} \pi(t, \mathbf{x}) &= \arg \min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - r(t, \mathbf{x})\|^2 \\ \text{s.t. } \mathcal{L}_f h + \mathcal{L}_G h \mathbf{u} + \frac{\partial h}{\partial t} &\geq -\alpha(h). \end{aligned} \quad (4)$$

Barrier functions in real-world applications are not necessarily smooth. As described in [22], generalized gradients and regularity conditions can be used to deal with non-differentiable points of a locally Lipschitz CBF.

**Definition 2** ([33, Thm. 2.5.1]). Let  $h$  be Lipschitz near  $\mathbf{z} = (t, \mathbf{x}) \in \mathbb{R}^{n+1}$ , and suppose  $\mathcal{Z}$  is any set of Lebesgue measure zero in  $\mathbb{R}^{n+1}$ . The generalized gradient of  $h$  is

$$\partial h(\mathbf{z}) = \text{co} \left\{ \lim_{i \rightarrow \infty} \nabla h(\mathbf{z}_i) \mid \mathbf{z}_i \rightarrow \mathbf{z}, \mathbf{z}_i \notin \mathcal{Z} \cup \Omega_h \right\}, \quad (5)$$

where  $\Omega_h$  is the zero-measure set where  $h$  is non-differentiable and  $\text{co}$  is the convex hull.

**Definition 3** ([33, Def. 2.3.4]). A function  $h$  is regular at  $\mathbf{z} \in \mathbb{R}^{n+1}$  provided that for all  $\mathbf{d} \in \mathbb{R}^{n+1}$ , the one-sided directional derivative  $h'(\mathbf{z}; \mathbf{d}) = \lim_{e \downarrow 0} e^{-1} (h(\mathbf{z} + e\mathbf{d}) - h(\mathbf{z}))$  exists and satisfies  $h'(\mathbf{z}; \mathbf{d}) = \limsup_{e \downarrow 0} \frac{h(\mathbf{z} + e\mathbf{d}) - h(\mathbf{z})}{e}$ .

Following the result in [22, Thm. 2], we can extend the notion of time-varying CBF in Def. 1 to functions  $h(t, \mathbf{x})$  that are not differentiable everywhere.

**Definition 4.** A locally Lipschitz, regular function  $h(t, \mathbf{x})$  is a *non-smooth time-varying CBF* on a set  $\mathcal{D}$  if  $\mathcal{S}(t)$  in (2) satisfies  $\mathcal{S}(t) \subset \mathcal{D} \subseteq \mathbb{R}^n$ , and there exists a locally Lipschitz extended class- $\mathcal{K}$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  such that:

$$\sup_{\mathbf{u} \in \mathcal{U}} \inf_{(w, \mathbf{v}) \in \partial h(t, \mathbf{x})} \{\mathbf{v}^\top (f(\mathbf{x}) + G(\mathbf{x})\mathbf{u}) + w\} \geq -\alpha(h(t, \mathbf{x}))$$

for all  $\mathbf{x} \in \mathcal{S}(t)$  and for all  $t \geq 0$ .

### B. Visibility and Safety as CBF Constraints

The condition that the evader is visible to the pursuer can be formulated as a non-smooth time-varying CBF. We model the safe set as the pursuer's FoV  $\mathcal{F}(\mathbf{x})$ , and the CBF as the signed distance from the evader  $\mathbf{y}(t)$  to the FoV boundary.

**Definition 5.** The *signed distance function*  $d : \mathbb{R}^p \times 2^{\mathbb{R}^p} \rightarrow \mathbb{R}$  computes the signed distance from a point  $\mathbf{q} \in \mathbb{R}^p$  to the boundary  $\partial \mathcal{F}$  of a set  $\mathcal{F} \subset \mathbb{R}^p$  as:

$$d(\mathbf{q}, \mathcal{F}) = \begin{cases} -\min_{\mathbf{q}^* \in \partial \mathcal{F}} \|\mathbf{q} - \mathbf{q}^*\|, & \text{if } \mathbf{q} \in \mathcal{F}, \\ \min_{\mathbf{q}^* \in \partial \mathcal{F}} \|\mathbf{q} - \mathbf{q}^*\|, & \text{if } \mathbf{q} \notin \mathcal{F}. \end{cases} \quad (6)$$

We define a *visibility CBF* using the signed distance to the pursuer's FoV:

$$h(t, \mathbf{x}) = -d(\mathbf{y}(t), \mathcal{F}(\mathbf{x})). \quad (7)$$

Note that  $h(t, \mathbf{x})$  in (7) is not differentiable at all points because the FoV can be an arbitrary set under occlusion such that  $\mathcal{F}(\mathbf{x}) \subseteq A(\mathbf{x})\mathcal{F}_0 + b(\mathbf{x})$ . However, we show that  $h(t, \mathbf{x})$  is Lipschitz continuous and thus a valid non-smooth time-varying CBF.

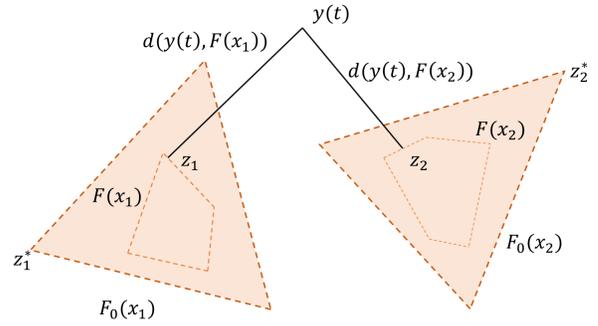


Fig. 2: Illustration of the proof that the visibility CBF is Lipschitz continuous in a 2D environment where the pursuer and evader states are positions and orientations. The triangles show the pursuer's unoccluded FoV, while the dashed polygons inside show the occluded FoV. The distances to the points  $\mathbf{z}_1$  and  $\mathbf{z}_2$  from  $\mathbf{y}(t)$  define the visibility CBF at  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , respectively.

**Proposition 1.** With bounded evader velocity,  $\|\dot{\mathbf{y}}(t)\| \leq k$ , the function  $h(t, \mathbf{x}) = -d(\mathbf{y}(t), \mathcal{F}(\mathbf{x}))$  is Lipschitz continuous.

*Proof.* Consider two points  $(t_1, \mathbf{x}_1)$  and  $(t_2, \mathbf{x}_2)$ . Then, by the triangle inequality:

$$\begin{aligned} |h(t_1, \mathbf{x}_1) - h(t_2, \mathbf{x}_2)| &= |d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_1)) - d(\mathbf{y}(t_2), \mathcal{F}(\mathbf{x}_2))| \\ &\leq |d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_1)) - d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_2))| \\ &\quad + |d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_2)) - d(\mathbf{y}(t_2), \mathcal{F}(\mathbf{x}_2))|. \end{aligned}$$

The distance from a point to a fixed set is 1-Lipschitz [34], and by the mean value theorem, it follows that

$$\begin{aligned} |d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_2)) - d(\mathbf{y}(t_2), \mathcal{F}(\mathbf{x}_2))| &\leq \|\mathbf{y}(t_1) - \mathbf{y}(t_2)\| \\ &\leq k|t_1 - t_2|. \end{aligned}$$

Let  $\mathbf{z}_1 \in \partial \mathcal{F}(\mathbf{x}_1)$  and  $\mathbf{z}_2 \in \partial \mathcal{F}(\mathbf{x}_2)$  be such that  $d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_1)) = \|\mathbf{y}(t_1) - \mathbf{z}_1\|$  and  $d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_2)) = \|\mathbf{y}(t_1) - \mathbf{z}_2\|$ . Then,

$$|d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_1)) - d(\mathbf{y}(t_1), \mathcal{F}(\mathbf{x}_2))| \leq \|\mathbf{z}_1 - \mathbf{z}_2\|.$$

With slight abuse of notation, let  $\mathcal{F}_0(\mathbf{x}) = A(\mathbf{x})\mathcal{F}_0 + b(\mathbf{x})$  be the unoccluded FoV at pursuer state  $\mathbf{x}$ . Since  $\mathcal{F}(\mathbf{x}) \subseteq \mathcal{F}_0(\mathbf{x})$ , we have  $\mathbf{z}_1 \in \mathcal{F}_0(\mathbf{x}_1)$  and  $\mathbf{z}_2 \in \mathcal{F}_0(\mathbf{x}_2)$ , and thus:

$$\|\mathbf{z}_1 - \mathbf{z}_2\| \leq m_H(\mathcal{F}_0(\mathbf{x}_1), \mathcal{F}_0(\mathbf{x}_2)) := \sup_{\mathbf{a} \in \mathcal{F}_0(\mathbf{x}_1), \mathbf{b} \in \mathcal{F}_0(\mathbf{x}_2)} \|\mathbf{a} - \mathbf{b}\|.$$

Now, consider  $\mathbf{z}_1^* \in \mathcal{F}_0(\mathbf{x}_1)$  and  $\mathbf{z}_2^* \in \mathcal{F}_0(\mathbf{x}_2)$  such that  $\|\mathbf{z}_1^* - \mathbf{z}_2^*\| = m_H(\mathcal{F}_0(\mathbf{x}_1), \mathcal{F}_0(\mathbf{x}_2))$ . Since  $\mathbf{z}_1^*, \mathbf{z}_2^*$  are points obtained from a Lipschitz linear transformation from  $\mathbf{x}_1, \mathbf{x}_2$ , there exists a constant  $C$  such that

$$\|\mathbf{z}_1^* - \mathbf{z}_2^*\| \leq C\|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Putting everything together, we conclude that:

$$|h(t_1, \mathbf{x}_1) - h(t_2, \mathbf{x}_2)| \leq k|t_1 - t_2| + C\|\mathbf{x}_1 - \mathbf{x}_2\|. \quad \square$$

Fig. 2 illustrates the above analysis for a 2D environment with a triangular unoccluded FoV. Thus, based on Def. 4, we can define a visibility CBF constraint as:

$$\inf_{(w, \mathbf{v}) \in \partial h(t, \mathbf{x})} \{\mathbf{v}^\top (f(\mathbf{x}) + G(\mathbf{x})\mathbf{u}) + w\} \geq -\alpha_v(h(t, \mathbf{x})) + \delta, \quad (8)$$

where  $\alpha_v$  is a Lipschitz extended class- $\mathcal{K}$  function and  $\delta \in \mathbb{R}$  is a slack variable that allows relaxing or tightening the visibility constraint. It can also be noted that for points where  $h(t, \mathbf{x})$  is differentiable, the above constraint reduces to a usual CBF constraint:

$$\begin{aligned} \nabla_{\mathbf{x}} d(\mathbf{y}, \mathcal{F}(\mathbf{x}))^\top (f(\mathbf{x}) + G(\mathbf{x})\mathbf{u}) + \nabla_{\mathbf{y}} d(\mathbf{y}, \mathcal{F}(\mathbf{x}))^\top \dot{\mathbf{y}} \\ \leq \alpha_v(-d(\mathbf{y}, \mathcal{F}(\mathbf{x}))) + \delta. \end{aligned} \quad (9)$$

To capture the safety requirement as well, we use the signed distance function  $d(\mathbf{x}, \mathcal{O})$  from the pursuer state  $\mathbf{x}$  to the obstacle set  $\mathcal{O}$  to define a CBF. When  $d(\mathbf{x}, \mathcal{O})$  is positive, the pursuer is outside of the obstacle set  $\mathcal{O}$ . Similar to the visibility CBF,  $d(\mathbf{x}, \mathcal{O})$  is non-smooth but Lipschitz [34]. Thus, we can define a safety CBF constraint as:

$$\inf_{\mathbf{v} \in \partial d(\mathbf{x}, \mathcal{O})} \{ \mathbf{v}^\top (f(\mathbf{x}) + G(\mathbf{x})\mathbf{u}) \} \geq -\alpha_s(d(\mathbf{x}, \mathcal{O})), \quad (10)$$

which simplifies for differentiable points of  $d(\mathbf{x}, \mathcal{O})$  to:

$$\nabla_{\mathbf{x}} d(\mathbf{x}, \mathcal{O})^\top (f(\mathbf{x}) + G(\mathbf{x})\mathbf{u}) \geq -\alpha_s(d(\mathbf{x}, \mathcal{O})). \quad (11)$$

In summary, we can impose the visibility and safety CBF constraints in (8) and (10) on a reference control signal  $r(t, \mathbf{x}, \mathbf{y})$  for the pursuer by synthesizing control inputs as:

$$\min_{\mathbf{u} \in \mathcal{U}, \delta \in \mathbb{R}} \|\mathbf{u} - r(t, \mathbf{x}, \mathbf{y})\|^2 + \lambda \delta^2 \quad \text{s.t. (8), (10),} \quad (12)$$

where  $\delta$  is a slack variable penalized by  $\lambda > 0$  that ensures feasibility of the program by relaxing the visibility constraint in (8) if necessary. Note that since  $\partial h(t, \mathbf{x})$  and  $\partial d(\mathbf{x}, \mathcal{O})$  are convex sets (Def. 2), the infima over linear functions in (8), (10) are concave functions of  $\mathbf{u}$  and, hence, (8), (10) are convex constraints in  $\mathbf{u}$ . At points where  $h(t, \mathbf{x})$  and  $d(\mathbf{x}, \mathcal{O})$  are differentiable, the constraints are linear in  $\mathbf{u}$ .

### C. Planning and Control with Visibility & Safety Constraints

The formulation in (12) is myopic in the sense that the future motion of the evader is predicted only through its instantaneous velocity  $\dot{\mathbf{y}}(t)$ . To maintain visibility of the evader, it is desirable to synthesize a non-myopic control policy for the pursuer that takes into account a longer-horizon prediction of the evader motion.

One option is to formulate an optimal control problem with horizon  $T$  to obtain a non-myopic control policy  $\pi(t, \mathbf{x}, \mathbf{y})$ :

$$\min_{\pi, \delta} \int_t^{t+T} \|\pi(\tau, \mathbf{x}(\tau), \mathbf{y}(\tau)) - r(\tau, \mathbf{x}(\tau), \mathbf{y}(\tau))\|^2 + \lambda \delta^2(\tau) d\tau$$

subject to the visibility and safety constraints in (8), (10) and the pursuer dynamics in (1). Unfortunately, there are a number of challenges with solving this problem. First, it requires predicting the evader configuration  $\mathbf{y}(\tau)$  over a long horizon  $\tau \in [t, t+T]$ , which is much more challenging than estimating the instantaneous evader velocity  $\dot{\mathbf{y}}(t)$  as required in (12). Second, the constraints in (8) and (10) would no longer be convex in  $\mathbf{u}$  because, at times  $\tau > t$ , the policy  $\pi(\tau, \mathbf{x}(\tau), \mathbf{y}(\tau))$  depends on the pursuer state  $\mathbf{x}(\tau)$ , which evolves nonlinearly according to the dynamics in (1). Third, while it is possible to formulate the problem as a nonlinear program, e.g., using CasADi [35], the non-convex

and potentially non-smooth constraints (8), (10) would make solving it numerically challenging and inefficient.

To avoid the optimal control formulation above, we introduce a trajectory planner that provides a reference control signal  $r(\tau)$  for  $\tau \in [t, t+T]$  that can be tracked by the evader using the myopic but convex optimization problem in (12). The planner's objective is to rapidly find a collision-free and feasible control trajectory for the pursuer that achieves visibility of the evader as a goal condition. We use Stable Sparse RRT (SST) [36], which is a kinodynamic planner that samples within the control space  $\mathcal{U}$  to generate a reference  $r(\tau)$  for  $\tau \in [t, t+T]$ . To ensure that the visibility constraint is satisfied, the goal condition of SST is set such that  $\mathbf{y}(t) \in \mathcal{F}(\mathbf{x}(T))$ . Note that this goal condition requires the evader configuration  $\mathbf{y}(t)$  only at time  $t$  and, thus, avoids predicting the future evader motion but can still lead to non-myopic behavior via re-planning.

To increase the efficiency of SST, we use the pre-trained Motion Planning Transformer (MPT) [37] to suggest sampling regions. MPT uses a transformer architecture that maps patches of an occupancy grid map to a sequence of regions that may contain feasible trajectories to the goal. SST's sampling is biased to the sequence of regions predicted by MPT to focus the tree expansion towards areas that may observe the evader.

A challenge with the formulation in (12) is the computation of gradients for the visibility and safety CBFs. In the case of camera or LiDAR sensors, the visibility CBF value can be computed by approximating the FoV  $\mathcal{F}(\mathbf{x})$  through ray tracing and finding the distance from the evader configuration  $\mathbf{y}$  to the FoV. However, there is no analytical method to compute the FoV gradient. We approximate the gradient of the visibility CBF  $h(t, \mathbf{x}) = -d(\mathbf{y}(t), \mathcal{F}(\mathbf{x}))$  using a least-squares finite-difference method. Let  $d^*(\mathbf{x}) := d(\mathbf{y}, \mathcal{F}(\mathbf{x}))$  be the distance to the FoV as a function of the pursuer's state, and let  $\mathcal{X}_p = \{\delta \mathbf{x}_i\}_{i=1}^N$  be a set of pursuer state perturbations. By Taylor expansion, the distance perturbation is  $d^*(\mathbf{x} + \delta \mathbf{x}_i) \approx d^*(\mathbf{x}) + \nabla_{\mathbf{x}} d^*(\mathbf{x})^\top \delta \mathbf{x}_i$ . Thus, by defining  $\Delta d_i^*(\mathbf{x}) = d^*(\mathbf{x} + \delta \mathbf{x}_i) - d^*(\mathbf{x})$ , we can approximate the gradient as  $\nabla_{\mathbf{x}} d^*(\mathbf{x}) \approx \arg \min_{\mathbf{p}} \sum_{i=1}^N \|\delta \mathbf{x}_i^\top \mathbf{p} - \Delta d_i^*(\mathbf{x})\|$ , which is computed using least squares. To achieve higher gradient accuracy, a larger number of perturbations  $N$  should be considered. For points where  $d^*(\mathbf{x})$  is not differentiable the least-squares problem does not have a unique solution. Generalized gradient vectors can be obtained as  $\mathbf{v} = \mathbf{C}^\dagger \mathbf{c} + (\mathbf{I} - \mathbf{C}^\dagger \mathbf{C})\mathbf{n}$ , where  $\mathbf{C} \in \mathbb{R}^{N \times n}$  has rows  $\delta \mathbf{x}_i^\top$ ,  $\mathbf{c} \in \mathbb{R}^N$  has elements  $\Delta d_i^*(\mathbf{x})$ , and  $\mathbf{n} \in \mathbb{R}^n$  is arbitrary.

The safety CBF value and gradient can be estimated from a point cloud observation,  $\mathcal{P} = \{\boldsymbol{\rho}_i\}_{i=1}^P$  with  $\boldsymbol{\rho}_i \in \mathbb{R}^P$ , e.g., obtained from a LiDAR or RGBD camera sensor. Consider the distance and direction of  $N$ -nearest points in  $\mathcal{P}$  to the pursuer, determined as  $\psi_j = \|\boldsymbol{\rho}_j - \phi(\mathbf{x})\|$  and  $\mathbf{g}_j = (\boldsymbol{\rho}_j - \phi(\mathbf{x})) / \|\boldsymbol{\rho}_j - \phi(\mathbf{x})\|$ , respectively, where the function  $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^P$  maps the pursuer state  $\mathbf{x}$  to its position  $\phi(\mathbf{x}) \in \mathbb{R}^P$ . The generalized gradient can be approximated as the set of these directions vectors, i.e.  $\partial g(\mathbf{x}) \approx \{\mathbf{g}_j\}_{j=1}^N$ , and the safety CBF value can be estimated as the minimum

TABLE I: Performance metrics for simulated ablations in CARLA

| Ablation              | Planner only | Controller only | Full system |
|-----------------------|--------------|-----------------|-------------|
| Initialization time ↓ | 24 s         | 8 s             | 7 s         |
| % of time in FoV ↑    | 59%          | 97%             | 98%         |
| Mean SDF ↓            | 8.3 m        | -3.9 m          | -5.0 m      |
| Max. relocate time ↓  | 37 s         | 6.4 s           | 4.6 s       |
| No. of Collisions ↓   | 4            | 0               | 0           |
| Control Frequency ↑   | 5 Hz         | 37 Hz           | 32 Hz       |

distance,  $d(\mathbf{x}, \mathcal{O}) \approx \min_{j \in \{1 \dots N\}} \psi_j$ .

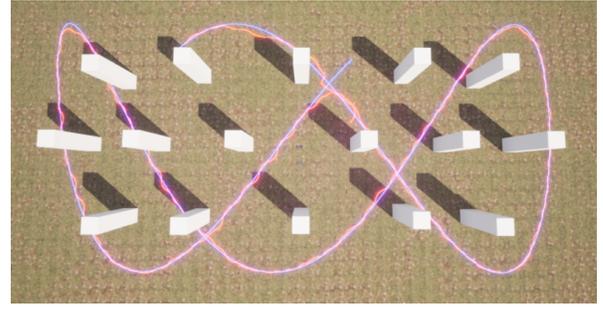
#### IV. EXPERIMENTS AND RESULTS

We use a mobile robot, equipped with a camera and a LiDAR, as a pursuer in our evaluation. The robot has state  $\mathbf{x} = [x, y, \theta]^\top \in \mathbb{R}^2 \times [-\pi, \pi]$  consisting of its position  $(x, y)$  and orientation  $\theta$ , input  $\mathbf{u} = [v, \omega]^\top \in \mathbb{R}^2$  defining its linear and angular velocities, and differential-drive dynamics,  $\dot{x} = v \cos(\theta)$ ,  $\dot{y} = v \sin(\theta)$ ,  $\dot{\theta} = \omega$ . In our experiments, the environment consists of obstacles  $\mathcal{O} \subset \mathbb{R}^2 \times [-\pi, \pi]$  placed on a planar surface. We use Hector SLAM [38] with 2D scans extracted from the LiDAR point clouds for occupancy mapping and pursuer state localization. The evader is modeled as a system with position  $\mathbf{y} \in \mathbb{R}^2$  and velocity  $\dot{\mathbf{y}} \in \mathbb{R}^2$ . The pursuer uses camera detections of the evader and an extended Kalman filter [30] with constant velocity motion model to estimate the evader state. The FoV  $\mathcal{F}(\mathbf{x})$  is computed using ray tracing in the occupancy map generated by Hector SLAM. The visibility and safety CBF values and gradients are computed as discussed at the end of Sec. III-C. We also include box constraints  $\mathcal{U}$  on  $\mathbf{u}$  in (12). We evaluated the performance of our method in 3D CARLA simulations [39] and on a real Jackal robot.

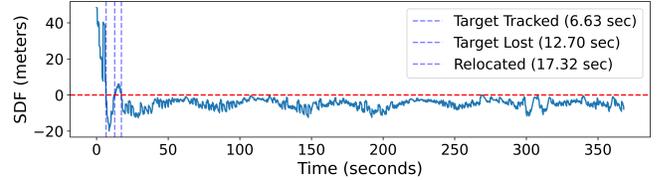
##### A. CARLA Simulations

A 3D environment and two cars that act as pursuer and evader were simulated in the CARLA autonomous driving simulator [39], shown in Fig. 3a. The pursuer’s FoV  $\mathcal{F}_0$  was a circular sector with range 80 m spanning  $60^\circ$ . The control space  $\mathcal{U}$  was set to  $[0, 12] \text{ m/s} \times [-1, 1] \text{ rad/s}$ . The evader follows a Lissajous curve, executing periodic movements within the simulation environment, described by  $\mathbf{y}(t) = [A \sin(at + \gamma), B \sin(bt)]^\top$ , with  $A = 180$ ,  $a = 0.15$ ,  $B = 90$ ,  $b = 0.40$ , and  $\gamma = 2.05$ . The evader’s path includes straight sections and sharp turns close to obstacles, allowing us to assess the controller’s capabilities in tracking and obstacle avoidance. Measurements of  $\mathbf{y}(t)$  and  $\dot{\mathbf{y}}(t)$  were provided to the pursuer directly by the CARLA simulator.

In the simulation setup, shown in Fig. 3a, the evader starts outside of the pursuer’s FoV. Initially, the pursuer lacks evader and environment information. Thus, the pursuer’s trajectory until the first evader detection is not included in the assessment. As shown in Fig. 4a, pursuer achieves its first evader detection within 11 seconds and loses tracking only once due to occlusion, as shown in Fig. 4b. It took 4.6 seconds to re-establish tracking, as depicted in Fig. 4c. The SDF values shown in Fig. 3b indicate that the pursuer successfully tracked the evader with brief loss in visibility during sharp turns or near obstacles.



(a) Pursuer (red) and evader (blue) trajectories in the CARLA simulation. The environment spans  $400 \times 400 \text{ m}^2$  and has sixteen white pillars of size  $5 \times 5 \text{ m}^2$  as obstacles.



(b) SDF values from the evader to the pursuer’s FoV. The red line denotes the FoV boundary. Segments above the line indicate that the evader is outside the FoV. The initial distance between the evader and pursuer was 110 meters.

Fig. 3: CARLA simulation trajectory

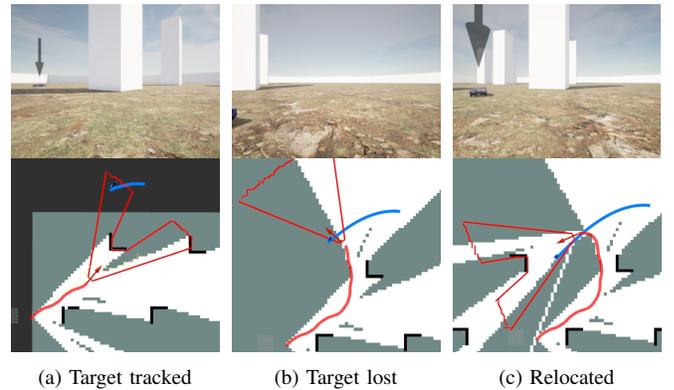


Fig. 4: Top: Snapshots from pursuer’s view along its trajectory. Bottom: The red and blue curves denote the pursuer and evader trajectories, respectively. (a) The instance when the target is first observed by the pursuer; (b) The pursuer loses the evader around a corner; (c) The pursuer relocates the evader after turning the corner.

We also carried out an ablation study of the planner and controller components in our method with different pursuer and evader initialization. We compared: (a) directly using the planned reference trajectory to control the pursuer (*Planner only*); (b) using only the controller with visibility and safety CBF constraints (*Controller only*); and (c) Asynchronous planner+controller (*Full system*). The results are reported in Table I. *Planner only* struggled without real-time safety constraints, leading to collisions. *Controller only* performed better than *Planner only*, maintaining visibility and avoiding collisions. *Full system* shows the best performance in visibility and safety across all metrics, maintaining the evader within the pursuer’s FoV for 98% of the trajectory while avoiding collisions with obstacles. This shows our method’s effectiveness for reliable, safe navigation in realistic simulated environments.

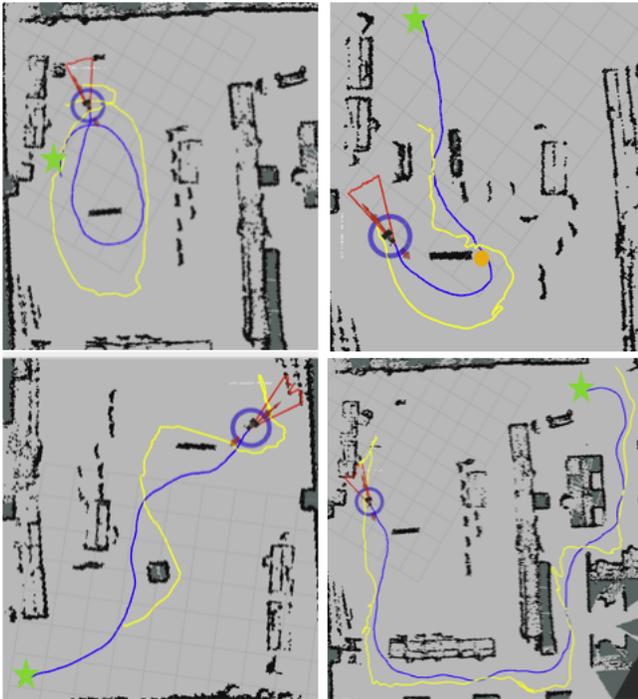


Fig. 5: **Real-world experiments.** The pursuer and evader paths are shown in blue and yellow, respectively. The green star indicates the pursuer’s starting position. The red region shows the pursuer FoV. Different experiments are shown clockwise from top left: (1) Simple loop around a single obstacle; (2) Evader follows an ‘S’ shape. The pursuer grazes an obstacle near a sharp turn denoted by the orange circle; (3) The evader charts a long path through a cluttered environment; (4) The pursuer cuts off the evader.

TABLE II: Performance metrics for Jackal robot experiments

| Experiment                           | 1      | 2       | 3      | 4      |
|--------------------------------------|--------|---------|--------|--------|
| % of time in FoV $\uparrow$          | 92%    | 76%     | 92%    | 91%    |
| Mean SDF $\downarrow$                | -0.8 m | -0.25 m | -1.0 m | -1.4 m |
| Max. relocate time $\downarrow$      | 1.0 s  | 2.9 s   | 2.2 s  | 0.6 s  |
| Min. dist. to obstacles $\downarrow$ | 11 cm  | 0 cm*   | 3 cm   | 20 cm  |

## B. Real World Experiments

In the real experiments, the pursuer was a differential-drive Jackal robot, equipped with an OS-1 Ouster 3D LiDAR and D455 RealSense depth camera. We used a human-controlled Turtlebot, affixed with an AprilTag as an evader. We imposed a limited, triangular FoV  $\mathcal{F}_0$  which spans  $30^\circ$  with range 2 m to demonstrate our method’s ability to deal with limited sensing. We ran the *Controller Only* method at 50 Hz onboard the Jackal, consisting of sensing, mapping, evader detection, and control. Due to the complex environment and high computational demands (about ten times slower for MPT inference), we excluded the planner on Jackal robot, which struggles with incomplete and noisy maps to provide a reliable path. The pursuer control space  $\mathcal{U}$  was set to  $[0, 0.5] \text{ m/s} \times [-0.5, 0.5] \text{ rad/s}$ . We conducted all experiments in a lab environment and used common objects as obstacles.

We present 4 different experiments, shown in Fig. 5, designed to highlight a specific capability or challenge that our method faces. Fig. 6 shows the actual setup for experiments 2 and 3. We initialized all experiments with the



Fig. 6: Actual setup for ‘S’-shape (left) and a section of long cluttered path (right). On the left, the pursuer navigates through rows of desks. The evader can be seen in the background to the right of the pursuer. On the right, the pursuer navigates through an aisle and turns two corners in an ‘S-shape’.

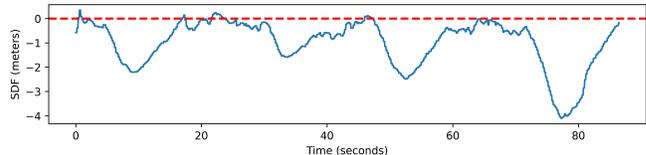


Fig. 7: SDF of evader to pursuer FoV in Experiment 3 (‘Long cluttered path’). The red line corresponds to the FoV boundary. Segments above the line indicate that the evader is outside the FoV.

evader inside the pursuer FoV to provide an initial detection for the estimator. Table II shows performance metrics for each experiment. In experiment 2 (‘S-shape’), the pursuer grazes an obstacle (leg of a standing whiteboard) as it turns the corner marked by the orange circle in Fig. 5. The confined space and sharp turns explain lower visibility for this configuration. In experiment 4 (‘Cut-off’), we see that the pursuer is able to cut off the evader as it moves around an obstacle, rather than following its path exactly. Fig. 7 shows that in experiment 3 (‘Long cluttered path’), the SDF value increases during turns as the pursuer momentarily loses the evader, and decreases during straight sections. The max relocate time is the longest time elapsed between losing the evader and bringing it back into the pursuer’s FoV. Experiment 4 (‘Cut-off’) has the lowest relocate time as the pursuer keeps its distance and does not follow the evader around obstacles, a behavior that would cause it to lose sight of the evader for longer. As expected, this metric is the highest for experiment 2 (‘S-shape’) in which sharp turns mean that the pursuer loses the evader for longer periods. Overall, we see that the pursuer achieves close tracking across various configurations.

## V. CONCLUSION

This paper introduces a convex programming formulation for control synthesis with visibility and safety constraints. We show that both constraints can be formulated using non-smooth but Lipschitz continuous CBFs. Coupling our controller with a kinodynamic planner that generates non-myopic trajectories to observe a mobile evader, we demonstrate autonomous robot pursuit in 3D simulations and real-world environments using onboard sensing under occlusion. Future work will focus on supplementing our method with an exploration strategy if the pursuer does not begin with a detection of the evader, and with a robust estimator that maintains multiple hypotheses for evader motion.

## REFERENCES

- [1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, pp. 299–316, 2011.
- [2] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 24–33, 2004.
- [3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.
- [4] K. D. Julian and M. J. Kochenderfer, "Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1768–1778, 2019.
- [5] J. O'Rourke, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.
- [6] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.
- [7] S. Bhattacharya, S. Hutchinson, and T. Basar, "Game-theoretic analysis of a visibility based pursuit-evasion game in the presence of obstacles," in *American Control Conference (ACC)*, 2009, pp. 373–378.
- [8] S. Bhattacharya and S. Hutchinson, "On the existence of nash equilibrium for a two-player pursuit—evasion game with visibility constraints," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 831–839, 2010.
- [9] R. Zou and S. Bhattacharya, "On optimal pursuit trajectories for visibility-based target-tracking game," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 449–465, 2019.
- [10] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007, iFAC Symposium on Nonlinear Control Systems.
- [11] 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, 2016.
- [12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European control conference (ECC)*, 2019, pp. 3420–3431.
- [13] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Conference on Decision and Control (CDC)*, 2014, pp. 6271–6278.
- [14] W. Shaw Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 530–545, 2021.
- [15] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, p. 8352–8358.
- [16] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3293–3298.
- [17] A. Clark, "Verification and synthesis of control barrier functions," in *IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6105–6112.
- [18] A. Isaly, M. Ghanbarpour, R. G. Sanfelice, and W. E. Dixon, "On the feasibility and continuity of feedback controllers defined by multiple control barrier functions for constrained differential inclusions," in *American Control Conference (ACC)*, 2022, pp. 5160–5165.
- [19] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *IEEE Conference on Decision and Control (CDC)*, 2020, pp. 3717–3724.
- [20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 165–174.
- [21] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.
- [22] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [23] M. Ghanbarpour, A. Isaly, R. G. Sanfelice, and W. E. Dixon, "Optimal safety for constrained differential inclusions using nonsmooth control barrier functions," *IEEE Control Systems Letters*, vol. 7, pp. 1303–1308, 2023.
- [24] A. Thirugnanam, J. Zeng, and K. Sreenath, "Nonsmooth control barrier functions for obstacle avoidance between convex regions," *arXiv preprint arXiv:2306.13259*, 2023.
- [25] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5249–5256.
- [26] T. Hoang, E. Bayasgalan, Z. Wang, G. Tschepnakis, and D. Panagou, "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," in *American Control Conference (ACC)*, 2017, pp. 5580–5585.
- [27] D. Panagou and V. Kumar, "Maintaining visibility for leader-follower formations in obstacle environments," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1811–1816.
- [28] S. Maniopoulos, D. Panagou, and K. J. Kyriakopoulos, "Model predictive control for the navigation of a nonholonomic vehicle with field-of-view constraints," in *American Control Conference (ACC)*, 2013, pp. 3967–3972.
- [29] H. Gao, W. Pengying, S. Yao, K. Zhou, M. Ji, H. Liu, and C. Liu, "Probabilistic visibility-aware trajectory planning for target tracking in cluttered environments," in *American Control Conference (ACC)*, 2024, pp. 594–600.
- [30] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction 1961.
- [31] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upercroft, "Simple online and realtime tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2016.
- [32] K. Garg and D. Panagou, "Robust control barrier and control lyapunov functions with fixed-time convergence guarantees," in *American Control Conference (ACC)*, 2021, pp. 2292–2297.
- [33] F. H. Clarke, *Optimization and nonsmooth analysis*. SIAM, 1990.
- [34] S. Fitzpatrick, "Metric projections and the differentiability of distance functions," *Bulletin of the Australian Mathematical Society*, vol. 22, no. 2, p. 291–312, 1980.
- [35] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [36] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.
- [37] J. J. Johnson, U. S. Kalra, A. Bhatia, L. Li, A. H. Qureshi, and M. C. Yip, "Motion planning transformers: A motion planning framework for mobile robots," *arXiv preprint arXiv:2106.02791*, 2022.
- [38] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *IEEE International Symposium on Safety, Security and Rescue Robotics*, 2011.
- [39] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning (CoRL)*, 2017.