Hide and Seek with Visibility Constraints using Control Barrier Functions

Shumon Koga[1] Minnan Zhou[2] Dimitra Panagou[3] Nikolay Atanasov[2]

***Abstract*— This paper formulates hiding and seeking objectives for mobile sensors with limited field of view (FoV) as control barrier function (CBF) constraints. We consider a mobile pursuer aiming to keep an evader within its FoV despite line-of-sight occlusions. The signed distance function (SDF) of the FoV set is used to formulate visibility seeking as a CBF constraint on the control input space of the pursuer. An equivalent CBF constraint can achieve hiding, i.e., avoiding a pursuer's FoV from the evader's perspective. Maintaining visibility of a moving target is demonstrated in CARLA simulations, employing OctoMap ray-tracing to estimate the SDF and the SDF derivative of an occluded FoV.**

I. Introduction

Pursuit-evasion problems [1] are widely 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 the pursuit-evasion with visibility constraints by graph [6] or game-theoretic approaches [7], [8]. This work proposes a novel control design for hide-and-seek with field-of-view (FoV) constraints, employing control barrier function (CBF) techniques.

Originated from the pioneering work by Ames and coauthors in [9], CBFs for safety-critical systems have been intensively developed, such as robust CBF [10] and adaptive CBF [11]. CBFs are defined so that the positivity of CBF implies the forward invariance of the corresponding safe set. Therefore, the safe control is designed so that CBF maintains positive value, which is typically employed by Quadratic Programming (QP) to minimize the square norm of the control input subject to the CBF constraint which is linear in control, thereby the feasibility of the solution is ensured [12]. As a method related also to stabilization, [13] proposed fixed-time CBF and utilizing CBF for stability at a given goal region, subject to the safety constraints.

One challenge of CBF-based methods is how to determine the CBF so that the positivity of the function is equivalent to the desired safety [14]. This paper introduces the Signed Distance Function (SDF) as CBF, which returns a positive distance from a point to the closest point on the boundary of the given set if the point is outside the set, and returns a negative distance if the point is inside the set. SDF represents the continuous field of a given object, which has been applied to many computer vision tasks such as shape representations [15] and mapping an environment [16]. Utilizing SDF for modeling FoV of a sensing robot has been done in [17] for active mapping and in [18] for target tracking. However, those work do not consider occlusion that is caused when FoV is overlapped with objects. Traditionally, occlusion in sensing has been treated by ray-tracing through laser scanning [19].

Vision-based control renders a challenge in the treatment of raw image inputs and visibility constraints in the control design. In the presence of FoV constraints, [20] proposed model predictive control for differential-drive robot navigation, while maintaining visibility of a static landmark. Leader-follower formations for maintaining visibility were addressed in [21]. A vision-based target tracking for quadrotor landing on a ground vehicle was focused in [22]. For a safety-critical system, [23] developed vision-based feedback control by combining CBF and Neural Radiance Fields (NeRF) [24]. In [25], the authors proposed a safe control under limited perception in a dynamic environment.

This paper proposes hide-and-seek control design for a mobile pursuer tracking a moving evader with limited FoV and occlusions from the environment geometry. The objective is to enforce the negativity of SDF of FoV evaluated at the evader position in the pursuer-view coordinate. Owing to the versatility of SDF, the change of shape in FoV due to occlusion is represented by utilizing ray-tracing at every time. We prove that the required time to seek an evader under the CBF-based control is bounded by a known constant. The performance is demonstrated in a CARLA simulation [26], employing OctoMap[27] ray-tracing to estimate the SDF and the SDF derivative of the occluded FoV.

II. Preliminaries

Consider a system given by ordinary differential equation:

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

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$ are the state and control input and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are continuous vector-valued and matrix-valued functions, respectively.

In safety-critical applications, it is desirable to control the state of a dynamical system to ensure that it remains within a safe set:

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

[1] The author is with Honda R&D Co. Ltd., Tokyo, Japan, [2] The authors are with the Department of Electrical and Computer Engineering, UC San Diego, 9500 Gilman Drive, La Jolla, CA, 92093, USA, {skoga,m9zhou,natanasov}@ucsd.edu. [3] The author is with the Department of Robotics and the Department of Aerospace Engineering, University of Michigan, Ann Arbor, USA. dpanagou@umich.edu

where $h : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.

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

$$\sup_{\mathbf{u}\in\mathcal{U}}\{\mathcal{L}_{\mathbf{f}}h(t,\mathbf{x}) + \mathcal{L}_{\mathbf{G}}h(t,\mathbf{x})\mathbf{u}\} + \frac{\partial h}{\partial t}(t,\mathbf{x}) \geq -\alpha(h(t,\mathbf{x})),$$

for all $\mathbf{x} \in \mathcal{D}$ and for all $t \geq 0$.

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

Lemma 1 ([13]). *Let $h : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}$ be the time-varying CBF defining a time-varying safe set* (2)*. Then, for the system* (1)*, any Lipschitz continuous control input $\mathbf{u}(\mathbf{x})$ satisfying*

$$\mathcal{L}_{\mathbf{f}}h(t,\mathbf{x}) + \mathcal{L}_{\mathbf{G}}h(t,\mathbf{x})\mathbf{u}(\mathbf{x}) + \frac{\partial h}{\partial t}(t,\mathbf{x}) \geq -\alpha(h(\mathbf{x})), \quad (3)$$

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

For a locally Lipschitz reference control signal $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^m$, the following CBF quadratic program (QP) obtains the minimum control perturbation to guarantee safety:

$$\begin{aligned}\mathbf{u}^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^m} & (\mathbf{u}-\mathbf{r}(\mathbf{x}))^\top P(\mathbf{u}-\mathbf{r}(\mathbf{x})) \\ & \text{s.t. } \mathcal{L}_{\mathbf{f}}h + \mathcal{L}_{\mathbf{G}}h\mathbf{u} + \frac{\partial h}{\partial t} \geq -\alpha h.\end{aligned} \quad (4)$$

where $P \in \mathbb{R}_+^{m\times m}$ is a positive-definite weight matrix.

III. Seeking under Occluded FoV

A. Problem Statement

Consider a dynamical system given by (1), where the state $\mathbf{x} \in \mathbb{R}^n$ represents the pursuer's physical state including both position and orientation. We also consider a moving evader where the evader's physical state at time t is denoted as $\mathbf{y}(t) \in \mathbb{R}^n$. To model visibility, we introduce two functions. One is the relative vector $\mathbf{q} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^3$ defined by

$$\mathbf{q}(\mathbf{x},\mathbf{y}) = R^\top(s_p(\mathbf{y}) - s_p(\mathbf{x})). \quad (5)$$

where $R \in SO(3) \subset \mathbb{R}^{3\times3}$ is the orientation matrix of the pursuer, and $s_p : \mathbb{R}^n \to \mathbb{R}^3$ is a function mapping the state to the position. The other is the Signed Distance Function (SDF) defined below.

Definition 2. The *signed distance function* $d : \mathbb{R}^3 \to \mathbb{R}$ associated with a set $\mathcal{F} \subset \mathbb{R}^3$ is:

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

where $\partial\mathcal{F}$ is the boundary of $\mathcal{F}$.

Let $\bar{\mathcal{F}} \subset \mathbb{R}^3$ be the fixed non-occluded FoV set in the coordinate of pursuer-view. Let $\mathcal{F}_x(\mathbf{x})$ be the occluded FoV, which satisfies $\mathcal{F}_x(\mathbf{x}) \subset \bar{\mathcal{F}}$ for all $\mathbf{x} \in \mathbb{R}^n$. Seeking is achieved when it holds $d(\mathbf{q}(\mathbf{x},\mathbf{y}),\mathcal{F}_x(\mathbf{x})) \leq 0$.

Problem: Given the target state $\mathbf{y}(t)$ and its time derivative $\dot{\mathbf{y}}(t)$ for all $t \geq 0$, design the control law $\mathbf{u}$ to make $\mathbf{x}(t) \in \mathcal{S}(t) = \{\mathbf{x} \in \mathbb{R}^n | d(\mathbf{q}(\mathbf{x},\mathbf{y}(t)),\mathcal{F}_x(\mathbf{x})) \leq 0\}$ for all $t \geq T$ for some user-defined $T > 0$.

Equivalently to the seeking problem above, hiding problem can also be solved by introducing the evader FoV and the corresponding SDF as a CBF constraint, which we do not focus in this paper.

B. SDF-Based CBF for Finite-Time Convergence

Let h be CBF for seeking formulated as

$$h(t,\mathbf{x}) = -\gamma\left(d(\mathbf{q}(\mathbf{x},\mathbf{y}(t)),\mathcal{F}_x(\mathbf{x})) + \varepsilon\bar{d}\right), \quad (7)$$

where $\varepsilon \in (0,1)$ and $\bar{d} = -\min_{\mathbf{q}} d(\mathbf{q},\bar{\mathcal{F}})$. Let $\mathbf{q}_x := \mathbf{q}(\mathbf{x},\mathbf{y})$ be the relative position of the robot state within the target view-coordinate. To solve the Problem, we apply the CBF-QP and show the following lemma.

Proposition 1. *Consider the closed-loop system of* (1) *under the CBF-QP* (4) *with CBF* (7)*. For any parameter $\varepsilon \in (0,1)$, it holds $\mathbf{x}(t) \in \mathcal{S}(t) = \{\mathbf{x} \in \mathbb{R}^n | d(\mathbf{q}(\mathbf{x},\mathbf{y}(t)),\mathcal{F}_x(\mathbf{x})) \leq 0\}$ for all $t \geq T$, where*

$$T = \frac{1}{\alpha}\log\left(1 + \frac{d_0}{\varepsilon\bar{d}}\right) \quad (8)$$

and $d_0 = d(\mathbf{q}(\mathbf{x}(0),\mathbf{y}(0)),\mathcal{F}_x(\mathbf{x}(0))) > 0$.

Proof. Since $d_0 = d(\mathbf{q}(\mathbf{x}(0),\mathbf{y}(0)),\mathcal{F}_x(\mathbf{x}(0))) > 0$, by (7), it holds that $h(0) < 0$ for any $\varepsilon > 0$. Since the CBF constraint ensures $h(t) \geq h(0)e^{-\alpha t}$, it holds that

$$d(\mathbf{q}(\mathbf{x}(t),\mathbf{y}(t)),\mathcal{F}_x(\mathbf{x}(t))) \leq (d_0 + \varepsilon\bar{d})e^{-\alpha t} - \varepsilon\bar{d}. \quad (9)$$

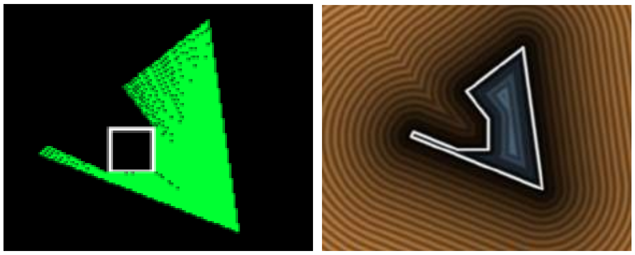
Thus, if the right hand side of the inequality above becomes 0 at $t = T$, it is clear that $d(\mathbf{q}(\mathbf{x}(t),\mathbf{y}(t)),\mathcal{F}_x(\mathbf{x}(t))) < 0$ for all $t \geq T$. The condition yields (8). □

C. Occlusion

A challenging task is to obtain the gradient of SDF of FoV with respect to the pursuer pose state. Here, let $d^*(\mathbf{x}) := d(\mathbf{q}(\mathbf{x},\mathbf{y}),\mathcal{F}_x(\mathbf{x}))$ be SDF w.r.t. the pursuer pose state. Consider a set of pursuer pose perturbation $\mathcal{X} = \{[r\boldsymbol{\eta}_i, 0], [r\boldsymbol{\eta}_i, \pm\delta\theta]\}_{i=1}^N \in \mathbb{R}^{3\times3N}$, where $r \in \mathbb{R}_+$ and $\delta\theta \in \mathbb{R}_+$ are sufficiently small positive scalar values and $\boldsymbol{\eta}_i = [\cos(\frac{2\pi i}{N}), \sin(\frac{2\pi i}{N})]$ and the corresponding SDF $d^*(\mathbf{x}+\delta\mathbf{x}_i)$ for each $\delta\mathbf{x}_i \in \mathcal{X}$. By Taylor expansion, the perturbation of SDF is $d^*(\mathbf{x}+\delta\mathbf{x}_i) = d^*(\mathbf{x}) + \frac{\partial d^*}{\partial\mathbf{x}}(\mathbf{x})\delta\mathbf{x}_i$. Thus, by defining $\Delta d_i^*(\mathbf{x}) = d^*(\mathbf{x}+\delta\mathbf{x}_i) - d^*(\mathbf{x})$, one can approximate the gradient as $\frac{\partial d^*}{\partial\mathbf{x}}(\mathbf{x}) \approx \operatorname{argmin}_{\mathbf{p}} \sum_{i=1}^{3N} ||\mathbf{p}^\top\delta\mathbf{x}_i - \Delta d_i^*(\mathbf{x})||$, which is solved by the least square method.

IV. Experiment

In this section, we evaluate the performance of the proposed hide-and-seek method in 2-D and 3-D simulation environments. All experiments are conducted using Python and the CARLA simulator.



(a) Ray-tracing (b) Signed distance

Fig. 1: pursuer 2D perception with ray-tracing and the corresponding SDF.

A. 2D Simulation

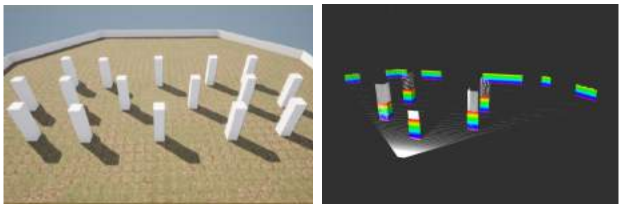
1) Environment Settings: In the planar simulation, we construct a scene with multiple polygonal obstacles and a polygonal map boundary. The octagonal scene contains sixteen rectangular obstacles, providing varying angles of obscured pursuer vision in complex evader motion trajectories. The scale of the map is $400m \times 400m$, and each obstacle measures $5m \times 5m$. Physical limitations of the environment, such as friction and collision interactions, are not considered during the actual motion process.

2) Pursuer Motion and Perception: We consider a unicycle model with control input $\mathbf{u} = [u, \omega]^\top \in \mathbb{R}^2$ where u is the linear velocity and ω is the angular velocity, the state variable is defined as $\mathbf{x} = [x, y, \theta] \in \mathbb{R}^3$ for position in $x-y$ coordinate, and the angle θ. Namely, the dynamics is given by $\dot{x} = u\cos(\theta), \dot{y} = u\sin(\theta), \dot{\theta} = \omega$. In 2-D simulation, the pursuer is equipped with a depth camera boasting a 60° FoV and an effective measurement range of 10 meters. We have simplified the camera's maximum detection area into a triangular region rather than a sectorial one to ensure that the actual visible area conforms to a polygonal shape. We employ ray tracing to determine the pursuer's actual visible region, with the ray-tracing featuring an angular resolution of 0.6°. The resulting visible area through ray-tracing is depicted in Fig. 1a. After determining the pursuer's visible range, we calculate the signed distance from the evader to this visible region using the polygonal Signed Distance Function (SDF). The distribution of distances within the polygon is depicted in Fig. 1b. Upon obtaining the signed distance, we employ the finite difference method to compute the approximate gradient of the signed distance with respect to the pursuer's pose.

3) Evader Settings: The evader model follows a Lissajous curve executing periodic movements within the simulation environment, described by $x = A\sin at + \delta$ and $y = B\sin bt$ in $x-y$ coordinate at time $t \geq 0$. Throughout its movement, the evader persistently navigates through gaps between obstacles, encompassing scenarios like near-linear paths, gradual turns, and sharp maneuvers. The trajectory parameters, along with other simulation details, are listed in Table I.

TABLE I: Chosen values for each parameter

Parameters	Values	Parameters	Values
$\mathbf{x}(0)$	(0,0,0)	τ	0.1
A	180	a	0.15
B	90	b	0.40
δ	2.05	ε	0.2
$\mathbf{r}(\mathbf{x})$	(18,0)	γ	5



(a) Environment (b) OctoMap

Fig. 2: CARLA 3-D simulation setup.

B. CARLA 3D Simulation

CARLA [26] is a renowned open-source simulator designed specifically for autonomous driving research. Established on realistic simulation environments, it offers a rich set of features to facilitate the development, training, and validation of autonomous driving systems. The simulator's high-fidelity graphics and physics engine provide a near-real-world experience, enabling researchers to study autonomous vehicle behavior under many conditions.

1) Environment Settings: In CARLA, leveraging the capabilities of the Unreal Engine, we have constructed a straightforward 3-D simulation map, bounded and reflecting the same quantity and positioning of obstacles as the previously mentioned 2-D map. In this 3D representation, the ground is textured with grass, while obstacles receive a concrete finish. We have disabled the physical interactions between obstacles and vehicles, meaning no tangible physical collisions will occur. Nevertheless, we ensure that the vehicle sensors accurately detect the presence of these obstacles. Throughout the simulation, we refrain from employing CARLA's Traffic-Manager for traffic planning, as well as its built-in automated navigation feature. Analogous to the 2D simulation, the vehicle position is directly affected by the CBF controller.

2) CARLA Communication: In CARLA, sensor data and vehicle position information are communicated through the Robot Operating System (ROS) [28]. Within the 3-D simulation environment, we have activated the RGBD camera, LiDAR, and GPS positioning systems. This data is subsequently published to corresponding ROS nodes for access by the controller or during debugging processes. The CBF controller influences the simulation process by manipulating elements within the simulation environment, such as the vehicle's physical parameters and the simulation running frequency, achieved through interfacing with the CARLA Python API.

3) OctoMap in CARLA Simulation: OctoMap [27] is a pioneering open-source library designed to create volumetric 3D environment maps in real time. Rooted in a probabilistic occupancy estimation framework, it represents an environment with an octree data structure, allowing for efficient storage, manipulation, and queries. In the 3-D simulation, we employ OctoMap to construct both the global map and

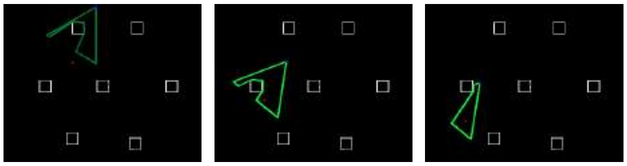


Fig. 3: 2D simulation process illustrating the pursuer's progression from initial discovery to stable tracking, Light green and bright green indicate successful tracking, signifying that, at that moment, the signed distance from the evader to the pursuer's visible region is less than zero.

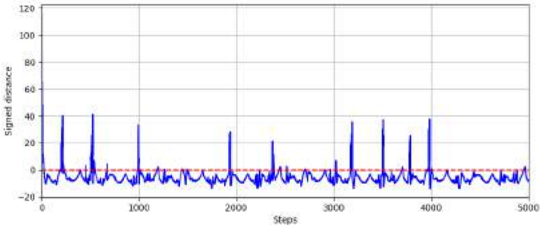


Fig. 4: Variation of the signed distance from the evader to the pursuer's visible region during the simulation.

the pursuer's visible region. The mapping process with OctoMap during the simulation is illustrated in Fig. 2b, where colors indicate the height from ground and white blocks are point clouds. Initially, we convert the RGBD images into point clouds using the ROS `depth_image_proc` package. Subsequently, we leverage the OctoMap library to transform these point clouds into an accessible global OctoMap, along with its projection on a 2-D plane.

C. Results

Fig. 3 shows the snapshots of the 2-D simulation, which illustrates a successful result in a transition from not detecting the evader to tracking and maintaining the visibility. The green box represents the pursuer's visible region, while the blue and red points correspond to the pursuer and evader, respectively. Fig. 4 displays the variation of the signed distance from the evader to the pursuer's visible region throughout the simulation. Fig. 5 depicts linear and angular velocities of both the pursuer and the evader. When the evader resides within the pursuer's field of view, the control input remains consistently at $(18, 0)$. We utilize the parameter ε to delineate a safe region within the pursuer's visual field, which enlarges the control input to enforce the target to stay close to the center of the pursuer's FoV. The pursuer's angular velocity variation closely aligns with that of the evader, maintaining a stable phase difference. From this, we deduce that the CBF controller adeptly modifies the reference input $\mathbf{r}(\mathbf{x})$, equipping the pursuer to effectively and consistently track the evader within a set duration. As evident from the figures, following the initial phase and the pursuer's first tracking success, in most cases, the pursuer manages to maintain stable tracking convinced by the negativity of the signed distance. The 2-D simulation ran on a Windows platform, requiring approximately 160 ms of computation time for each time step on an i7-13700H processor.

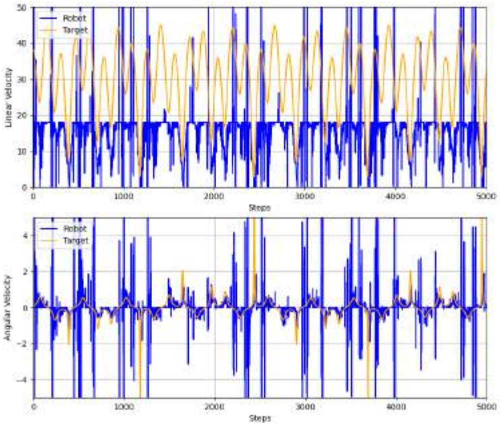


Fig. 5: Control inputs of the proposed method.

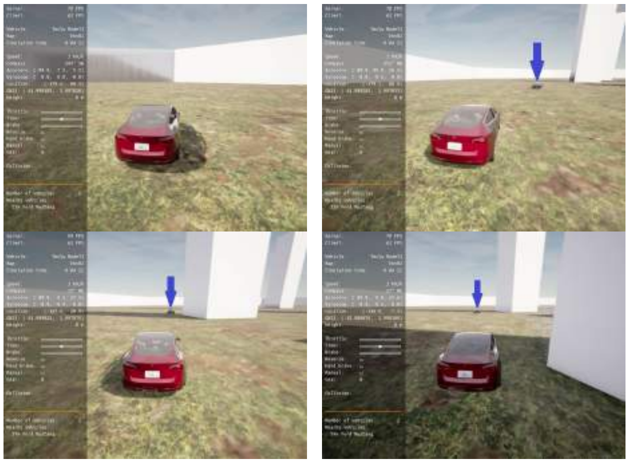


Fig. 6: A sequence from the CARLA simulation illustrating the pursuer's perspective, transitioning from initial non-tracking to achieving consistent evader tracking. The red vehicle represents the Seeker, while the blue vehicle, indicated by the indigo arrow, is the evader vehicle.

The 3-D CARLA simulation ran on a desktop with i7-7700K and RTX3060 graphics card. Within our custom map, the server side operated at 90Hz, while the client side ran approximately at 60Hz. The CBF controller updated the positions of both the evader and the pursuer every 0.16 seconds. Fig. 6 illustrates a sequence of images from the CARLA simulation, capturing the pursuer's perspective as it transitions from initially not tracking the evader to achieving stable vision-tracking.

V. Conclusion

This paper has addressed hide-and-seek control design for a single pursuer and a evader under visibility constraint. We introduced the Signed Distance Function for modeling the visibility under occlusion. Using the barrier function method, we show that the visibility of the moving evader is achieved within a finite time. The occlusion of the field-of-view is handled by the ray-tracing. The performance is demonstrated in both 2-D simulation and 3-D CARLA simulator employing the OctoMap. In future work, we will consider a non-myopic control by the model predictive control to minimize a long-term cost, sensor-based control by constructing an estimator of the evader state, and the hide-and-seek scenario where the evader is also trying to seek the pursuer.

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*, vol. 57. Oxford University Press Oxford, 1987.

[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 *2009 American control conference*, pp. 373–378, IEEE, 2009.

[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] 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.

[10] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[11] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2021.

[12] 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)*, pp. 3420–3431, IEEE, 2019.

[13] K. Garg and D. Panagou, "Robust control barrier and control lyapunov functions with fixed-time convergence guarantees," in *2021 American Control Conference (ACC)*, pp. 2292–2297, IEEE, 2021.

[14] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, IEEE, 2020.

[15] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

[16] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373, IEEE, 2017.

[17] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active exploration and mapping via iterative covariance regulation over continuous $SE(3)$ trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2735–2741, 2021.

[18] P. Yang, S. Koga, A. Asgharivaskasi, and N. Atanasov, "Policy learning for active target tracking over continuous $se(3)$ trajectories," in *Learning for Dynamics and Control Conference*, pp. 64–75, PMLR, 2023.

[19] A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and shannon mutual information computation for robot exploration," *IEEE Transactions on Robotics*, 2023.

[20] S. Maniatopoulos, D. Panagou, and K. J. Kyriakopoulos, "Model predictive control for the navigation of a nonholonomic vehicle with field-of-view constraints," in *2013 American control conference*, pp. 3967–3972, IEEE, 2013.

[21] D. Panagou and V. Kumar, "Maintaining visibility for leader-follower formations in obstacle environments," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1811–1816, IEEE, 2012.

[22] T. Hoang, E. Bayasgalan, Z. Wang, G. Tsechpenakis, and D. Panagou, "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," in *2017 American Control Conference (ACC)*, pp. 5580–5585, IEEE, 2017.

[23] M. Tong, C. Dawson, and C. Fan, "Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10511–10517, IEEE, 2023.

[24] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[25] D. Agrawal, R. Chen, and D. Panagou, "Gatekeeper: Safety critical control of nonlinear systems with limited perception in unknown and dynamic environments," *arXiv preprint arXiv:2211.14361*, 2022.

[26] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," in *Autonomous Robots*, vol. 34, pp. 189–206, Springer, 2013.

[28] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.