2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Beyond Collision Cones: Dynamic Obstacle Avoidance for Nonholonomic Robots via Dynamic Parabolic Control Barrier Functions

Hun Kuk Park*, Taekyung Kim* and Dimitra Panagou

***Abstract*— Control Barrier Functions (CBFs) are a powerful tool for ensuring the safety of autonomous systems, yet applying them to nonholonomic robots in cluttered, dynamic environments remains an open challenge. State-of-the-art methods often rely on collision-cone or velocity-obstacle constraints which, by only considering the angle of the relative velocity, are inherently conservative and can render the CBF-based quadratic program infeasible, particularly in dense scenarios. To address this issue, we propose a Dynamic Parabolic Control Barrier Function (DPCBF) that defines the safe set using a parabolic boundary. The parabola's vertex and curvature dynamically adapt based on both the distance to an obstacle and the magnitude of the relative velocity, creating a less restrictive safety constraint. We prove that the proposed DPCBF is valid for a kinematic bicycle model subject to input constraints. Extensive comparative simulations demonstrate that our DPCBF-based controller significantly enhances navigation success rates and QP feasibility compared to baseline methods. Our approach successfully navigates through dense environments with up to 100 dynamic obstacles, scenarios where collision cone-based methods fail due to infeasibility. [Project Page][1] [Code] [Video]**

I. INTRODUCTION

Ensuring safety is a fundamental challenge for autonomous systems, particularly nonholonomic robots and autonomous vehicles operating in dynamic and cluttered environments. Control Barrier Functions (CBFs) have emerged as a powerful tool for enforcing safety constraints in real-time, formulated within a Quadratic Program (QP) [1] or with Model Predictive Control (MPC) [2]. Their effectiveness has led to widespread adoption in applications from robotic navigation [3] to multi-agent coordination [4].

Collision avoidance can be encoded through a distance-based CBF, which defines the safe set based on the Euclidean distance to an obstacle. To incorporate the relative velocity between the robot and the obstacle, one can employ a High-Order CBF (HOCBF) [5]. However, it requires all control inputs to appear in the CBF condition, which makes it difficult to be applied to systems with inputs of different relative degrees [6].

Recent work addresses dynamic obstacles within the CBF framework by leveraging velocity-obstacle (VO) constraints [7], also referred to as collision cones in other literature [8]. These methods define the unsafe set as a collision cone in the relative-velocity space and constrain the relative velocity to lie outside a fixed cone [9], [10]. This approach has been successfully applied to various systems, including the kinematic bicycle model, by showing that the constraint has relative degree one with respect to all control inputs. Despite their advantages for dynamic obstacle avoidance, cone-based and VO-based methods exhibit fundamental conservatism. Because the safety constraint depends only on the heading angle of the relative velocity, the robot is prohibited from moving toward the obstacle, regardless of their distance or relative speed. This rigidity can induce immediate QP infeasibility when the initial relative velocity lies within a collision cone, or in dense environments where the union of multiple cones removes all feasible control inputs, even when sufficient collision-free space exists (see Fig. 1a).

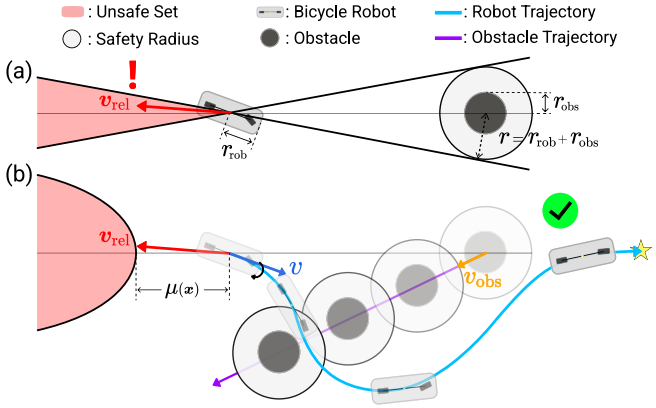

Fig. 1: Illustrative comparison of two CBF mechanisms in dynamic obstacle avoidance scenarios. (a) Since the Collision Cone CBF (C3BF) evaluates only the heading of the relative velocity, it may classify the robot as unsafe regardless of its actual distance from the obstacle. (b) Our Dynamic Parabolic CBF (DPCBF) establishes a more flexible safety condition by evaluating both relative position and the magnitude of relative velocity, which avoids unnecessary restrictions when clearance is large. As shown in (b), the parabola's vertex shifts away from the robot's origin by $\mu(\boldsymbol{x})$. This relaxes the safety constraint, allowing for less restrictive movements that approach the boundary of the unsafe set while remaining provably safe.

This paper introduces a ***Dynamic Parabolic Control Barrier Function (DPCBF)*** that explicitly incorporates both clearance and the magnitude of the relative velocity. Instead of a fixed cone, we define a state-dependent parabolic safety boundary whose curvature and vertex adapt with distance and relative velocity (see Fig. 1b). This design yields less conservative safety constraints, improving the feasibility of the CBF-based controller in cluttered, dynamic environments. The main contributions of this work are:

*These authors contributed equally to this work
The authors are with the Department of Robotics, University of Michigan, Ann Arbor, MI 48109 USA parkcart@umich.edu, taekyung@umich.edu, dpanagou@umich.edu
Dimitra Panagou is also with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, USA
[1]Project page: https://www.taekyung.me/dpcbf

- We propose a DPCBF for nonholonomic robots in dynamic obstacle avoidance tasks, which dynamically shapes the safety boundary to provide less conservative safety margins by adapting to distance and relative velocity.
- We prove that DPCBF is valid for the kinematic bicycle model under input constraints.
- We show extensive simulation results in dense, dynamic environments, demonstrating higher feasibility and success rates, and lower control intervention, compared to state-of-the-art CBF methods.

II. PRELIMINARIES

A. Control Barrier Functions

Consider a continuous-time, control-affine system:

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

where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ is the state and $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, with $\mathcal{U}$ representing the admissible control set for System (1). The functions $f : \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{U} \to \mathbb{R}^{n\times m}$ are both assumed to be locally Lipschitz continuous.

Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. We define

$$\mathcal{C} \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq 0\}, \tag{2a}$$
$$\partial\mathcal{C} \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) = 0\}, \tag{2b}$$
$$\mathrm{Int}(\mathcal{C}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) > 0\}, \tag{2c}$$

where $\mathcal{C}$ is referred to as the *safe set*.

Definition 1 (Forward Invariance)**.** *A closed set $\mathcal{C} \subset \mathbb{R}^n$ is* forward invariant *for System (1) under a state-feedback control law $\boldsymbol{u} = \pi(\boldsymbol{x})$ if the solution $\boldsymbol{x}(t)$ of the closed-loop system $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t)) + g(\boldsymbol{x}(t))\pi(\boldsymbol{x}(t))$ for every initial state $\boldsymbol{x}(0) \in \mathcal{C}$ satisfies $\boldsymbol{x}(t) \in \mathcal{C}, \forall t \geq 0$.*

Definition 2 (Control Barrier Function [1])**.** *Given the set $\mathcal{C}$ defined by* (2a)*, the function h is a* CBF *for System (1) if there exists an extended class $\mathcal{K}_\infty$ function $\alpha(\cdot)$ such that*

$$\sup_{\boldsymbol{u}\in\mathcal{U}} \big[\underbrace{L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u}}_{\dot{h}(\boldsymbol{x},\boldsymbol{u})}\big] \geq -\alpha(h(\boldsymbol{x})) \quad \forall \boldsymbol{x} \in \mathcal{C}. \tag{3}$$

We denote $L_f h$ and $L_g h$ as the Lie derivatives of the function h with respect to f and g.

Lemma 1. ([11, Theorem 1]) *Let h satisfy the CBF condition* (3) *and define*

$$K_{\text{cbf}}(\boldsymbol{x}) \coloneqq \Big\{\boldsymbol{u} \in \mathcal{U} \;\Big|\; L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} \geq -\alpha\big(h(\boldsymbol{x})\big)\Big\}. \tag{4}$$

Then, any Lipschitz continuous feedback controller $\boldsymbol{u} = \pi(\boldsymbol{x}) \in K_{\text{cbf}}(\boldsymbol{x})$ renders $\mathcal{C}$ forward invariant for System (1).

To enforce that trajectories of (1) remain in $\mathcal{C}$ (2a), we solve the following Quadratic Program with CBF constraint (CBF-QP):

$$\begin{aligned}\boldsymbol{u}^\star(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \|\boldsymbol{u} - \boldsymbol{u}_{\text{ref}}(\boldsymbol{x})\|_2^2 \\ \text{s.t. } L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} \geq -\alpha\big(h(\boldsymbol{x})\big).\end{aligned} \tag{5}$$

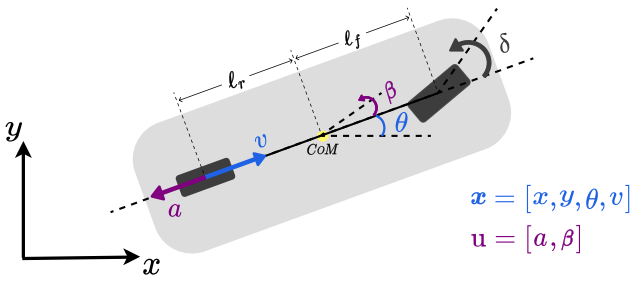

Fig. 2: Schematic of the kinematic bicycle model. The robot's state is defined by its Center of Mass (CoM) position (x, y), heading angle θ, and forward velocity v. The distances from the CoM to the front and rear axles are ℓ_f and ℓ_r, respectively. The front-wheel steering angle is δ, and β is the resulting vehicle slip angle.

Note, inputs are bounded: $\mathcal{U} \neq \mathbb{R}^m$. By Lemma 1, if h is a CBF, applying $\boldsymbol{u} = \boldsymbol{u}^\star(\boldsymbol{x})$ guarantees the state in the safe set $\mathcal{C}$ for all time.

B. Bicycle Model

In this paper, we consider a robot modeled by the kinematic bicycle model [12], [13] (see Fig. 2). The state is $\boldsymbol{x} = [x, y, \theta, v]^\top$, where (x, y) denotes the position of the vehicle's center of mass (CoM), θ is the heading angle, and v is the forward velocity. The control inputs are longitudinal acceleration a and the forward-wheel steering angle δ. Let ℓ_f and ℓ_r denote the distances from the CoM to the front and rear axles, respectively, and define the slip angle $\beta = \tan^{-1}\big(\tan(\delta)\,\ell_r/(\ell_f + \ell_r)\big)$.

To model the kinematic bicycle as a control affine system as in (1), we consider that the slip angle β is small, i.e., $\sin\beta \approx \beta$. Then, the dynamics equation follows [12]

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{v}\end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix}v\cos\theta\\ v\sin\theta\\ 0\\ 0\end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix}0 & -v\sin\theta\\ 0 & v\cos\theta\\ 0 & \frac{v}{\ell_r}\\ 1 & 0\end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix}a\\ \beta\end{bmatrix}}_{\boldsymbol{u}}, \tag{6}$$

where the inputs are now $\boldsymbol{u} = [a, \beta]^\top$.

C. Obstacle Model

We model a scenario with multiple dynamic obstacles. The state of the j-th dynamic obstacle, where $j \in \{1, \ldots, N_{\text{obs}}\}$, is represented by

$$\boldsymbol{x}^j_{\text{obs}} = [x^j_{\text{obs}}, y^j_{\text{obs}}, \theta^j_{\text{obs}}, v^j_{\text{obs}}]^\top, \tag{7}$$

where $x^j_{\text{obs}}, y^j_{\text{obs}}$ denote the obstacle's center position, θ^j_{obs} its heading angle, and v^j_{obs} its forward speed.

The dynamics of the j-th obstacle is described by a unicycle model with constant velocity: $\dot{x}^j_{\text{obs}} = v^j_{\text{obs}}\cos\theta^j_{\text{obs}}$ and $\dot{y}^j_{\text{obs}} = v^j_{\text{obs}}\sin\theta^j_{\text{obs}}$. We assume that the obstacle states are fully observable. For the remainder of the paper, we omit the superscript j and describe the CBF constraint for each obstacle for notational simplicity.

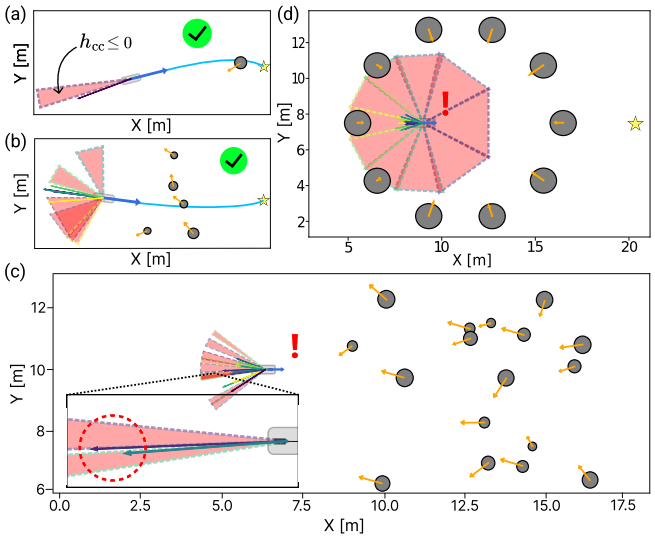

Fig. 3: A closer look at collision cone-based CBF. (a) Single obstacle: if $h_{\text{cc}}(\boldsymbol{x}(t_0)) \geq 0$, the CBF constraint keeps $h_{\text{cc}}(\boldsymbol{x}(t)) \geq 0$ for all $t \geq t_0$. (b) Five obstacles: if $h_{j,\text{cc}}(\boldsymbol{x}(t)) \geq 0$ for all j-th obstacle and the CBF-QP is feasible at t, safety is at least maintained at a given time t. (c) Even with $h_{j,\text{cc}}(\boldsymbol{x}(t_0)) \geq 0$ for all j-th obstacle, cone intersections can leave no admissible relative velocity direction, leading the CBF-QP infeasible. (d) Given the initial configuration where the robot is surrounded by the union of the collision cones, there is no feasible solution to the CBF-QP even though a large collision-free area exists nearby.

D. Distance-Based CBF

A distance-based CBF is a collision avoidance formulation based solely on the Euclidean distance. Let $r_{\text{rob}} > 0$ and $r_{\text{obs}} > 0$ denote conservative safety radii that over-approximate the robot and obstacle geometries, and define $r \coloneqq r_{\text{rob}} + r_{\text{obs}}$. Then, with the robot position $\boldsymbol{p} = [x, y]^\top$ and obstacle position $\boldsymbol{p}_{\text{obs}} = [x_{\text{obs}}, y_{\text{obs}}]^\top$, a distance-based safety constraint function is:

$$h_{\text{dist}}(\boldsymbol{x}) = \|\boldsymbol{p} - \boldsymbol{p}_{\text{obs}}\|^2 - r^2. \tag{8}$$

Because it only considers distance, this barrier is not, in general, a CBF except for simple systems in which the control inputs directly affect the velocity. Its myopic nature makes it particularly unsuitable for systems with nonholonomic constraints [9].

E. Collision Cone CBF

The Collision Cone CBF (C3BF) [9] was recently proposed for dynamic obstacle avoidance and constructs the CBF with the Velocity Obstacle (VO) [7] constraint. Given the relative position $\boldsymbol{p}_{\text{rel}}$ and relative velocity $\boldsymbol{v}_{\text{rel}}$, a conservative circle of radius r is placed around the obstacle center, and the collision cone is formed by the pair of tangents from the robot's center to the circle (see Fig. 1a). To implement the outside-of-cone constraint, C3BF defines

$$h_{\text{cc}}(\boldsymbol{x}) = \langle \boldsymbol{p}_{\text{rel}}, \boldsymbol{v}_{\text{rel}} \rangle + \|\boldsymbol{p}_{\text{rel}}\| \|\boldsymbol{v}_{\text{rel}}\| \cos\phi \tag{9}$$

where ϕ is half the cone angle, and $\cos\phi = \frac{\sqrt{\|\boldsymbol{p}_{\text{rel}}\|^2 - r^2}}{\|\boldsymbol{p}_{\text{rel}}\|}$. As illustrated in Fig. 3a, the unsafe set is the collision cone, i.e., $\{\boldsymbol{x} \mid h_{\text{cc}}(\boldsymbol{x}) < 0\}$. If the relative velocity at time t_0 lies outside this cone, the CBF constraint enforces the control input such that the relative velocity remains outside for all future time, thereby avoiding collision. This collision-cone approach is designed for moving obstacle avoidance tasks and it is also shown to be applicable to the kinematic bicycle model (6) [14]. Recently, it also has been extended to navigation tasks of quadrotors [15], ground mobile robots, and autonomous vehicles [14].

However, the mechanism itself poses conversely fundamental limitations of the C3BF. Since the CBF only monitors the relative velocity's angle with respect to the collision cone, the robot cannot ever drive towards the unsafe set, no matter how far away from those unsafe sets and how small the velocities are, the resulting behavior is ***extremely conservative***. In addition, if the initial relative velocity lies inside of the collision cone whenever the controller just gets initiated, the problem becomes immediately infeasible, even though there is a large free space in between the robot and the obstacle (see Fig. 3c). Furthermore, this problem is more prominent in multi-obstacle cases as shown in Fig. 3d. If the robot is surrounded by obstacles, the union of each cone shrinks the set of admissible relative velocity directions, making it easily infeasible in dense environments.

III. DYNAMIC PARABOLIC CBF

In this paper, we present a novel CBF formulation for dynamic-obstacle collision avoidance tasks. Existing methods either do not provide safety guarantees for moving obstacles or are prone to infeasibility when multiple obstacles are nearby. Accordingly, we focus on improving the key criteria: (i) guaranteeing safety for dynamic obstacles under input constraints, and (ii) improving the feasibility of the resulting CBF-based controller.

At a high level, we construct a safety constraint that explicitly accounts for the ***magnitude of the relative velocity***. Unlike C3BF, which relies on a fixed cone and only evaluates the heading of the relative velocity, our approach allows for a less restrictive safety condition. This distinction is crucial, as it permits the robot to safely move toward an obstacle when the relative velocity is low and clearance is large. We introduce a geometric strategy inspired by finite-time velocity obstacle formulations [16], in particular the truncated cone construction and parabolic approximation of the safe set boundary [17], [18].

A. DPCBF Formulation

Consider a robot modeled as System (6) navigating in an environment with dynamic obstacles.

1) Relative Coordinates: Define the relative position and velocity between the robot and the obstacle:

$$\boldsymbol{p}_{\text{rel}} = \begin{bmatrix} p_{\text{rel},x} \\ p_{\text{rel},y} \end{bmatrix} = \begin{bmatrix} x_{\text{obs}} - x \\ y_{\text{obs}} - y \end{bmatrix} \in \mathbb{R}^2, \tag{10a}$$

$$\boldsymbol{v}_{\text{rel}} = \begin{bmatrix} v_{\text{rel},x} \\ v_{\text{rel},y} \end{bmatrix} = \begin{bmatrix} v_{\text{obs}} \cos\theta_{\text{obs}} - v\cos\theta \\ v_{\text{obs}} \sin\theta_{\text{obs}} - v\sin\theta \end{bmatrix} \in \mathbb{R}^2 \tag{10b}$$

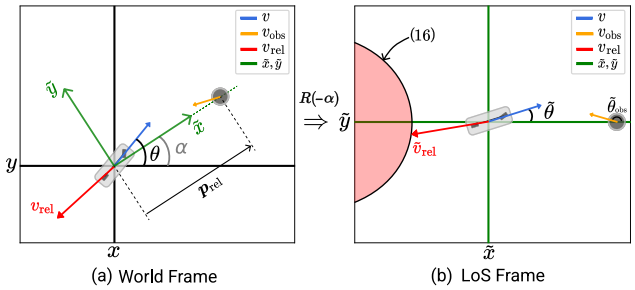

Fig. 4: Visualization of the global world frame (x, y) and the rotated Line-of-Sight (LoS) frame $(\tilde{x}, \tilde{y})$ used in our formulation. By rotating the coordinates by an angle α (11a), the $\tilde{x}$-axis of the LoS frame is aligned with the vector from the robot to the obstacle, $\boldsymbol{p}_{\text{rel}}$. This transformation simplifies the definition of the parabolic safety boundary (16), allowing its position and curvature to adapt online based on the relative velocity components in this new frame.

with norms $\|\boldsymbol{p}_{\text{rel}}\| = \sqrt{p_{\text{rel},x}^2 + p_{\text{rel},y}^2}$, $\|\boldsymbol{v}_{\text{rel}}\| = \sqrt{v_{\text{rel},x}^2 + v_{\text{rel},y}^2}$. Then, we rotate the coordinates to align with the line connecting the robot and the obstacle (see Fig. 4a). Denote the angle between the global x-axis and this new x-axis as the rotation angle:

$$\alpha = \operatorname{atan2}(p_{\text{rel},y}, p_{\text{rel},x}), \tag{11a}$$

$$\mathbf{R}(-\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \in SO(2). \tag{11b}$$

We refer to the rotated frame defined by the rotation matrix $\mathbf{R}$ as Line-of-Sight (LoS) frame throughout this paper (see Fig. 4b). Finally, we define the relative velocity in the LoS frame:

$$\tilde{\boldsymbol{v}}_{\text{rel}} = \begin{bmatrix} \tilde{v}_{\text{rel},x} \\ \tilde{v}_{\text{rel},y} \end{bmatrix} = \mathbf{R}(-\alpha) \begin{bmatrix} v_{\text{rel},x} \\ v_{\text{rel},y} \end{bmatrix}. \tag{12}$$

2) CBF Formulation and Design Maps: Let $r \in \mathbb{R}$ be the combined robot-obstacle radius defined in (8). Then,

$$d(\boldsymbol{x}) = \sqrt{\|\boldsymbol{p}_{\text{rel}}\|^2 - r^2}. \tag{13}$$

We introduce tunable parameters $k_\lambda, k_\mu > 0$, and the following functions:

$$\lambda(\boldsymbol{x}) = k_\lambda \frac{d(\boldsymbol{x})}{\|\boldsymbol{v}_{\text{rel}}\|}, \quad \mu(\boldsymbol{x}) = k_\mu\, d(\boldsymbol{x}). \tag{14}$$

Here $\lambda : \mathcal{X} \to \mathbb{R}$ adjusts the curvature of the parabola, and $\mu : \mathcal{X} \to \mathbb{R}$ shifts the parabola forward by the safe distance margin.

We propose the Dynamic Parabolic CBF (DPCBF) as follows:

$$h(\boldsymbol{x}) = \tilde{v}_{\text{rel},x} + \lambda(\boldsymbol{x})\tilde{v}_{\text{rel},y}^2 + \mu(\boldsymbol{x}), \tag{15}$$

where $\tilde{v}_{\text{rel},x}$ and $\tilde{v}_{\text{rel},y}$ are the components of the relative velocity between the robot and the obstacle in the LoS frame. Since $\lambda(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ depend on the relative distance and speed, the parabola representing the unsafe set dynamically adapts its shape online to the current situation. The CBF is now defined by measuring how close the endpoint of the rotated relative velocity (12) is to a specific parabolic region in this new plane (see Fig. 5):

$$\tilde{v}_{\text{rel},x} = -\lambda(\boldsymbol{x})\tilde{v}_{\text{rel},y}^2 - \mu(\boldsymbol{x}). \tag{16}$$

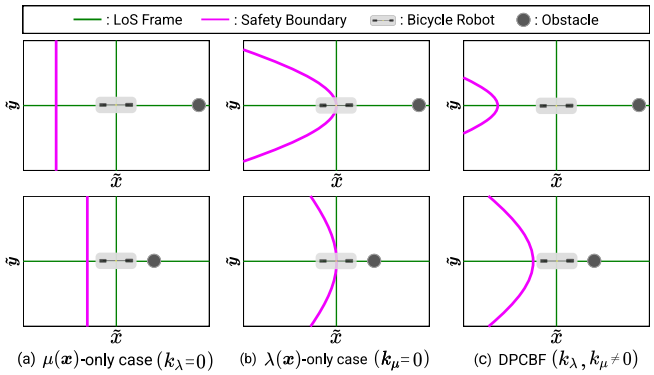

Fig. 5: Three examples illustrating how (16) shapes the safety boundary. (a) When $k_\lambda = 0$ (with k_μ active), as the obstacle gets closer, the parabola's vertex moves toward the robot, shrinking the safe region. (b) When $k_\mu = 0$ (with k_λ active), as the obstacle approaches or the relative velocity magnitude increases, the curvature of the parabola decrease, leading to a larger unsafe set. (c) In DPCBF, where $k_\lambda \neq 0$ and $k_\mu \neq 0$, both the vertex and the curvature of the parabola adapt dynamically.

This provides a significant advantage over cone-based methods. As illustrated in Fig. 1b, the boundary of the unsafe set (16) does not pass through the origin of the LoS-frame relative-velocity plane whenever the clearance to the obstacle, $d(\boldsymbol{x})$, is non-zero. This shift creates a feasible space where motion toward an obstacle is no longer treated immediately as unsafe. Instead, safety is now evaluated jointly using the current clearance $d(\boldsymbol{x})$ and the relative velocity $\boldsymbol{v}_{\text{rel}}$. We empirically show in Sec. IV that this design improves the feasibility of the DPCBF-based controller compared with prior work.

B. Validity of DPCBF

To make DPCBF valid for System (6), we require the following assumptions:

Assumption 1. The forward speed of System (6) is bounded by $v \in [v_{\min}, v_{\max}]$, where $v_{\max} > v_{\min} > 0$.

Assumption 1 is a required but mild assumption as in [5], [19], [20], since it excludes the degeneracy at $v = 0$ in (3) where $L_f h(\boldsymbol{x}) = 0$.

Assumption 2. The admissible distance satisfies $p_{\max} \geq \|\boldsymbol{p}_{\text{rel}}\| \geq p_{\min} := s\,r > 0$, where $s > 1$ is a safety margin. Therefore, $d(\boldsymbol{x}) = \sqrt{\|\boldsymbol{p}_{\text{rel}}\|^2 - r^2} > d_{\min} := \sqrt{p_{\min}^2 - r^2} > 0$. The maximum distance to obstacle $p_{\max}$ is determined by the finite sensing range.

To prove the proposed DPCBF is valid for System (6), we show that for any state on safe set boundary $\boldsymbol{x} \in \partial\mathcal{C}$, there exists an admissible control input $\boldsymbol{u} \in \mathcal{U} := \{[a, \beta]^\top \mid |a| \leq a_{\max}, |\beta| \leq \beta_{\max}\}$ that satisfies the CBF condition (3).

We first derive the corresponding terms for System (6). Let $\Phi(\boldsymbol{x})$ denote the maximum control authority at $\boldsymbol{x}$:

$$\begin{aligned} \Phi(\boldsymbol{x}) &:= \sup_{\boldsymbol{u}\in\mathcal{U}} L_g h(\boldsymbol{x})\boldsymbol{u} = \sup_{\boldsymbol{u}\in\mathcal{U}} \begin{bmatrix} C^a(\boldsymbol{x}) \\ C^\beta(\boldsymbol{x}) \end{bmatrix}^\top \boldsymbol{u} \\ &= \left|C^a(\boldsymbol{x})\right| a_{\max} + \left|C^\beta(\boldsymbol{x})\right| \beta_{\max}, \end{aligned} \tag{17}$$

where $C^a(\boldsymbol{x})$ and $C^\beta(\boldsymbol{x})$ are the derived control terms:

$$\begin{aligned}\big|C^a(\boldsymbol{x})\big| = \Bigg| &\underbrace{\Big[-1 + k_\lambda \frac{d(\boldsymbol{x})}{\|v_{\text{rel}}\|^3}\, v_{\text{obs}}\, \cos\tilde{\theta}_{\text{obs}} \tilde{v}_{\text{rel},y}^2\Big]}_{:=\eta_{a,\cos}(\boldsymbol{x})} \cos\tilde{\theta} \\ &+ \underbrace{\Big[k_\lambda \frac{d(\boldsymbol{x})}{\|v_{\text{rel}}\|^3} v_{\text{obs}} \sin\tilde{\theta}_{\text{obs}} \tilde{v}_{\text{rel},y}^2 - 2k_\lambda \frac{d(\boldsymbol{x})}{\|v_{\text{rel}}\|} \tilde{v}_{\text{rel},y}\Big]}_{:=\eta_{a,\sin}(\boldsymbol{x})} \sin\tilde{\theta} \\ &+ \underbrace{\Big[-k_\lambda \frac{d(\boldsymbol{x})}{\|v_{\text{rel}}\|^3} v\, \tilde{v}_{\text{rel},y}^2\Big]}_{:=\eta_{a,0}(\boldsymbol{x})} \Bigg|, \end{aligned} \tag{18}$$

and

$$\big|C^\beta(\boldsymbol{x})\big| = \big|\eta_{\beta,\cos}(\boldsymbol{x}) \cos\tilde{\theta} + \eta_{\beta,\sin}(\boldsymbol{x}) \sin\tilde{\theta}\big|, \tag{19}$$

where

$$\begin{aligned}\eta_{\beta,\cos}(\boldsymbol{x}) \coloneqq v \Bigg[&-\frac{\tilde{v}_{\text{rel},y}}{\|\boldsymbol{p}_{\text{rel}}\|} + 2k_\lambda \frac{d(\boldsymbol{x})}{\|\boldsymbol{v}_{\text{rel}}\|} \frac{\tilde{v}_{\text{rel},y}\tilde{v}_{\text{rel},x}}{\|\boldsymbol{p}_{\text{rel}}\|} \\ &+ k_\lambda \frac{v}{\ell_r} \frac{d(\boldsymbol{x})}{\|\boldsymbol{v}_{\text{rel}}\|} \tilde{v}_{\text{rel},y} \Big(\frac{v_{\text{obs}}}{\|\boldsymbol{v}_{\text{rel}}\|^3} \sin\tilde{\theta}_{\text{obs}}\, \tilde{v}_{\text{rel},y} - 2 \Big) \Bigg], \end{aligned} \tag{20a}$$

$$\begin{aligned}\eta_{\beta,\sin}(\boldsymbol{x}) \coloneqq v \Bigg[&k_\lambda \frac{\|\boldsymbol{p}_{\text{rel}}\|}{d(\boldsymbol{x})} \frac{\tilde{v}_{\text{rel},y}^2}{\|\boldsymbol{v}_{\text{rel}}\|} + k_\mu \frac{\|\boldsymbol{p}_{\text{rel}}\|}{d(\boldsymbol{x})} \\ &+ \frac{v}{\ell_r} \Big(1 - k_\lambda \frac{d(\boldsymbol{x})}{\|\boldsymbol{v}_{\text{rel}}\|^3}\, v_{\text{obs}} \cos\tilde{\theta}_{\text{obs}}\, \tilde{v}_{\text{rel},y}^2 \Big) \Bigg]. \end{aligned} \tag{20b}$$

Now, we aim to verify the following Nagumo's condition:

$$L_f h(\boldsymbol{x}) + \Phi(\boldsymbol{x}) \geq 0 \qquad \forall \boldsymbol{x} \in \partial\mathcal{C}, \tag{21}$$

where

$$\begin{aligned}L_f h(\boldsymbol{x}) = v \Bigg[&\Bigg(-k_\lambda \frac{\|\boldsymbol{p}_{\text{rel}}\|}{d(\boldsymbol{x})} \frac{\tilde{v}_{\text{rel},y}^2}{\|\boldsymbol{v}_{\text{rel}}\|} - k_\mu \frac{\|\boldsymbol{p}_{\text{rel}}\|}{d(\boldsymbol{x})} \Bigg) \cos\tilde{\theta} \\ &+ \Bigg(2k_\lambda \frac{\tilde{v}_{\text{rel},y}}{\|\boldsymbol{v}_{\text{rel}}\|} \frac{d(\boldsymbol{x})}{\|\boldsymbol{p}_{\text{rel}}\|} \tilde{v}_{\text{rel},x} - \frac{\tilde{v}_{\text{rel},y}}{\|\boldsymbol{p}_{\text{rel}}\|} \Bigg) \sin\tilde{\theta} \Bigg]. \end{aligned} \tag{22}$$

We partition the safe set boundary $\partial\mathcal{C}$ as $\partial\mathcal{C}_1$ and $\partial\mathcal{C}_2$, i.e., $\partial\mathcal{C}_1 \cup \partial\mathcal{C}_2 = \partial\mathcal{C}$:

$$\partial\mathcal{C}_1 \coloneqq \{\boldsymbol{x} \mid |\sin\tilde{\theta}| \geq \bar{s}\}, \quad \partial\mathcal{C}_2 \coloneqq \{\boldsymbol{x} \mid |\sin\tilde{\theta}| < \bar{s}\}, \tag{23}$$

where $\bar{s} \coloneqq \frac{v_{\text{obs}}}{v} \sin\tilde{\theta}_{\text{obs}} \in [0, 1)$. Therefore, we verify (21) in these two sub-groups separately, for $i \in \{1, 2\}$, yielding:

$$\inf_{\boldsymbol{x}\in\partial\mathcal{C}_i} \big[L_f h(\boldsymbol{x}) + |C^a(\boldsymbol{x})| a_{\max} + |C^\beta(\boldsymbol{x})| \beta_{\max}\big] \tag{24a}$$

$$\geq \underbrace{\inf_{\boldsymbol{x}\in\partial\mathcal{C}_i} L_f h(\boldsymbol{x})}_{D_{i,\min}(k_\lambda,k_\mu)} + \underbrace{\inf_{\boldsymbol{x}\in\partial\mathcal{C}_i} \big[|C^a(\boldsymbol{x})| a_{\max}\big]}_{C^a_{i,\min}(k_\lambda,k_\mu)} + \underbrace{\inf_{\boldsymbol{x}\in\partial\mathcal{C}_i} \big[|C^\beta(\boldsymbol{x})| \beta_{\max}\big]}_{C^\beta_{i,\min}(k_\lambda,k_\mu)} \tag{24b}$$

$$= D_{i,\min}(k_\lambda, k_\mu) + \underbrace{C^a_{i,\min}(k_\lambda, k_\mu) + C^\beta_{i,\min}(k_\lambda, k_\mu)}_{\Phi_{i,\min}(k_\lambda,k_\mu)} \geq 0. \tag{24c}$$

We will show that (24c) holds for both $i \in \{1, 2\}$, which together implies that (21) is satisfied.

Case 1 ($i = 1$). We aim to verify (24c) for the subset $\partial\mathcal{C}_1$ of the safety boundary. Dividing (24c) by v yields:

$$\begin{aligned}\inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} \frac{L_f h(\boldsymbol{x})}{v} + \inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} \Big|\frac{C^a(\boldsymbol{x})}{v}\Big| a_{\max} \\ + \inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} \Big|\frac{C^\beta(\boldsymbol{x})}{v}\Big| \beta_{\max} \geq 0 \end{aligned} \tag{25}$$

By $|\sin\tilde{\theta}| \geq \bar{s}$ in (23), we have $\inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} |C^a(\boldsymbol{x})/v| = 0$ (see Appendix[2] Sec. D). Also, we show that the control term for β and the drift term are both lower bounded by $\Phi_{1,\min}(k_\mu)$ and $D_{1,\min}(k_\lambda, k_\mu)$, respectively, and they depend on the hyperparameters k_λ and k_μ:

$$\inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} \Big|\frac{C^\beta(\boldsymbol{x})}{v}\Big| \beta_{\max} \coloneqq \Phi_{1,\min}(k_\mu) > 0 \tag{26}$$

$$\inf_{\boldsymbol{x}\in\partial\mathcal{C}_1} \frac{L_f h(\boldsymbol{x})}{v} \coloneqq D_{1,\min}(k_\lambda, k_\mu). \tag{27}$$

Therefore, the following condition is sufficient to satisfy (25):

$$\Phi_{1,\min}(k_\mu) \geq -D_{1,\min}(k_\lambda, k_\mu). \tag{28}$$

Case 2 ($i = 2$). Similarly, we show (24c) for $\partial\mathcal{C}_2$.

$$\begin{aligned}\inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} L_f h(\boldsymbol{x}) + \inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} |C^a(\boldsymbol{x})|\, a_{\max} \\ + \inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} |C^\beta(\boldsymbol{x})|\, \beta_{\max} \geq 0 \end{aligned} \tag{29}$$

By $|\sin\tilde{\theta}| < \bar{s}$ in (23), we have $\inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} |C^\beta(\boldsymbol{x})| = 0$ (see Appendix Sec. E). Also, each of the remaining terms are lower-bounded:

$$\inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} |C^a(\boldsymbol{x})| a_{\max} \coloneqq \Phi_{2,\min}(k_\lambda) > 0 \tag{30}$$

$$\inf_{\boldsymbol{x}\in\partial\mathcal{C}_2} L_f h(\boldsymbol{x}) \coloneqq D_{2,\min}(k_\lambda, k_\mu). \tag{31}$$

Therefore, the following condition is sufficient to satisfy (29):

$$\Phi_{2,\min}(k_\lambda) \geq -D_{2,\min}(k_\lambda, k_\mu). \tag{32}$$

Theorem 1. *Under Assumptions 1-2, the DPCBF is valid for System* (6) *under the input constraints, if there exist parameters k_λ and k_μ that satisfy* (28) *and* (32)*.*

Proof. A full proof with step-by-step derivation of each term can be found in Appendix Sec. D-Sec. E. A procedure for finding a feasible set of parameters (k_λ, k_μ) satisfying both (28) and (32) is given in Appendix Sec. F. □

Remark 1. *As is common in CBF analysis, including the prior works we evaluate against, the safety guarantee in Theorem 1 holds for a single CBF constraint, corresponding to one obstacle. For methods on composing multiple CBF constraints into a single constraint, we refer the readers to [6], [21]. A formal investigation into the composition of multiple DPCBFs under input constraints is outside the scope of this paper. We evaluate the performance of our DPCBF-based controller against the compared methods using a QP with multiple constraints in Sec. IV.*

[2]The Appendix can be found in: https://arxiv.org/abs/2510.01402

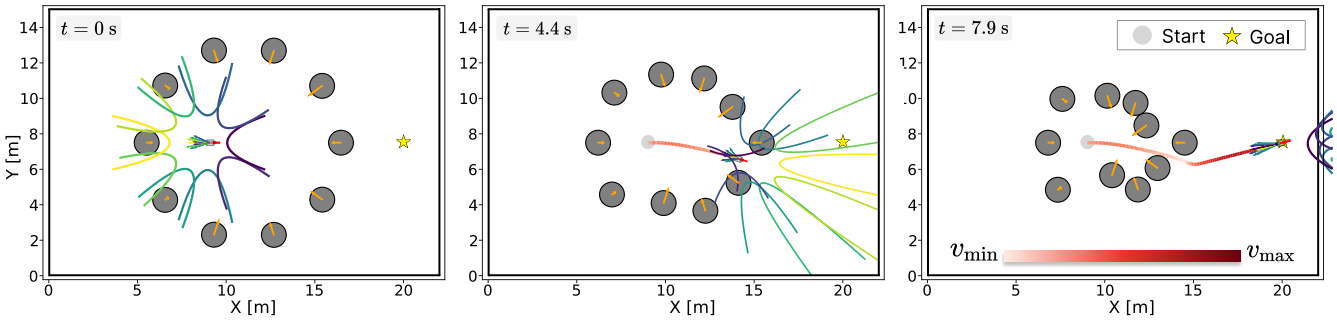

Fig. 6: Demonstration of the proposed DPCBF's navigation behavior in a surrounded environment with ten dynamic obstacles. For the same configuration, collision cone-based methods are infeasible, as shown in Fig 3d. (Left) At $t = 0$ s, with $v(t_0) = 0.5$ m/s, the QP with DPCBF constraints finds a feasible solution even though the robot is surrounded by unsafe sets, by keeping the relative-velocity vectors outside the dynamic parabolic boundaries, allowing it to proceed safely. (Center) By $t = 4.4$ s, the robot successfully maneuvers through a narrow passage. This is possible due to the less conservative formulation of DPCBF, which provides the necessary control flexibility in confined spaces. (Right) The robot safely navigates through the obstacles and reaches the goal at $t = 7.9$ s.

IV. RESULTS

A. Experimental setup

We conduct a series of simulation experiments to evaluate the performance of our proposed DPCBF and compare it against state-of-the-art baseline methods. The primary goal is to assess the ability of DPCBF to maintain safety while reducing conservatism, particularly in challenging scenarios with multiple dynamic obstacles. All experiments are performed in a simulated 2D environment. The robot is modeled as a kinematic bicycle (6) with parameters specified in Table I. Dynamic obstacles are modeled as discs with varying radii and move with constant velocity. The nominal controller $\boldsymbol{u}_{\text{ref}}$ is a simple proportional controller that drives the robot towards a goal location. We compare our DPCBF against three established CBF methods for dynamic obstacle avoidance:

(i) C3BF [15]: The collision-cone based CBF described in Sec. II-E.

(ii) MA-CBF-VO [22]: This method uses a velocity obstacle formulation for guidance and a separate, distance-based CBF to formally guarantee safety. To avoid the conservative behavior of VO approaches, the VO constraint is relaxed into a soft constraint by using a slack variable in the optimization's objective function, while the distance-based CBF remains a hard constraint for collision avoidance.

(iii) Dynamic zone-based CBF [23]: This approach modulates a circular safety zone around each obstacle based on the relative motion between the robot and the obstacle. The radius of this zone dynamically expands only when the robot and an obstacle are moving toward each other.

For all methods, the safety constraints are enforced via the CBF-QP formulation. We evaluate performance based on four key metrics: (i) Success rate: the percentage of trials where the robot reaches the goal without collision or infeasibility. (ii) Infeasible rate: the percentage of trials where the CBF-QP becomes infeasible, leading to mission failure. (iii) Collision rate: the percentage of trials where the robot's body intersects with an obstacle. (iv) QP cost: the total amount of deviation from the reference control input, calculated as the cumulative sum of the instantaneous QP cost, $\|\boldsymbol{u} - \boldsymbol{u}_{\text{ref}}\|_2^2$, over the trajectory, where a lower QP cost implies a more efficient and less conservative method.

Parameter	Bicycle Robot	Obstacles
Maximum velocity	3.5 [m/s]	1.2 [m/s]
Minimum velocity	0.2 [m/s]	0 [m/s]
Maximum sensing range	15 [m]	-
$a_{\max}$	5.0 [m/s^2]	-
β_{max}	0.28 [rad]	-
Max/Min radius	0.3 / - [m]	0.7 / 0.1 [m]
Rear axes distance ℓ_r	0.2 [m]	-
Safety buffer s	1.05	-

TABLE I: Main parameters for the simulation studies.

B. Comparison with C3BF

We first demonstrate a crucial qualitative comparison in Fig. 6, directly addressing the failure case for C3BF shown in Fig. 3d. In this challenging scenario, the robot is initially surrounded by obstacles. While C3BF becomes infeasible due to the complete overlap of collision cones, DPCBF successfully finds a path to the goal. Although the robot is similarly enveloped by parabolic safety boundaries, the dynamic nature of DPCBF provides a key advantage. Specifically, the state-dependent term $\mu(\boldsymbol{x})$ in (14) creates sufficient feasible space for the relative velocity in the CBF-QP. This directly illustrates how DPCBF overcomes the conservatism of cone-based methods.

C. Experimental Results

Performance Analysis in Dense Dynamic Environments. To test the core hypothesis that DPCBF alleviates infeasibility issues while ensuring safety, we simulate navigation in environments with an increasing number of dynamic obstacles, from 1 to 100. The results are summarized in Fig. 8. We first evaluate the methods in single-obstacle scenarios, where the formal safety guarantee holds for all methods except for Dynamic zone-based CBF. As expected from the theoretical guarantee, all such methods achieve a 100% success rate. The Dynamic zone-based CBF exhibits a 1.7% infeasibility rate because it is not a valid CBF for the kinematic bicycle model, regardless of the number of

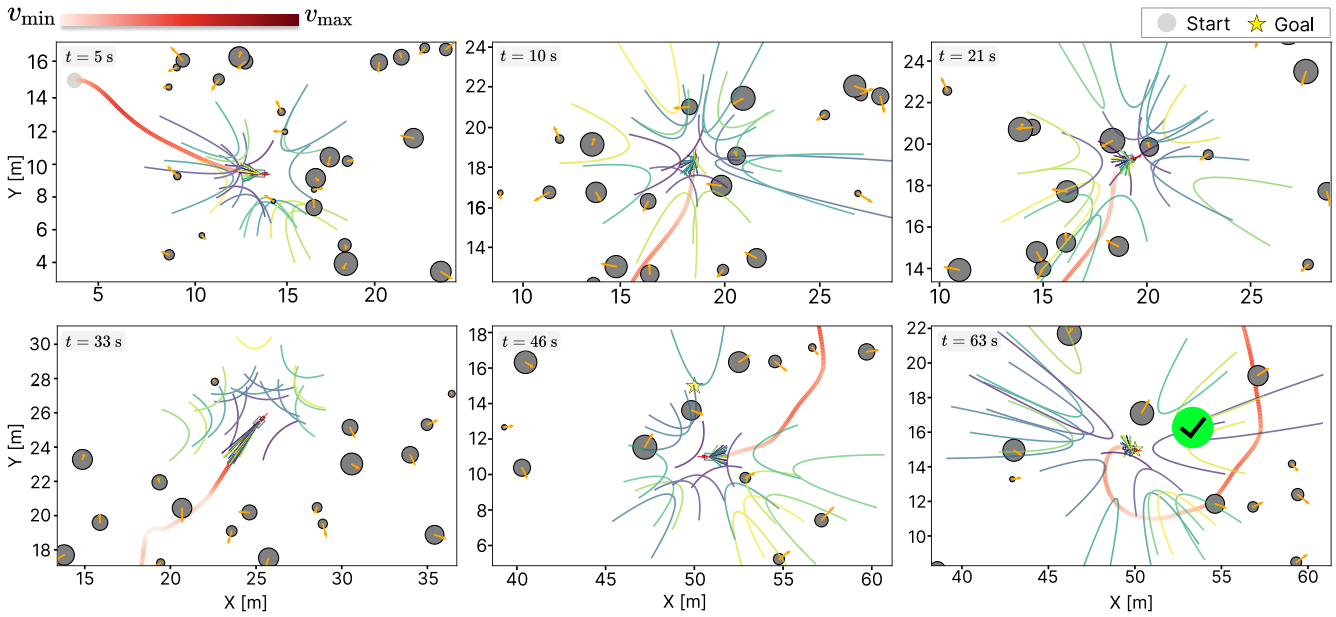

Fig. 7: A visualization of a successful navigation scenario using DPCBF with 100 dynamic obstacles, drawn from the statistical results in Fig. 8. All other compared baseline methods failed in this same configuration.

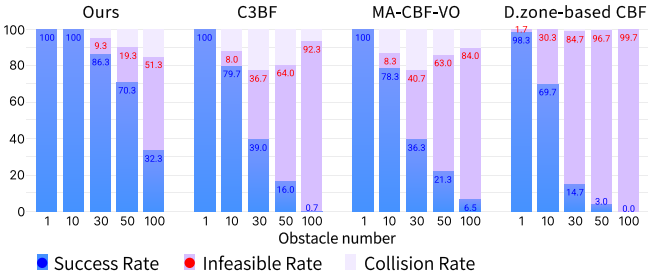

Fig. 8: Performance comparison of success, infeasible, and collision rates for our method and three baselines as the number of obstacles increases from 1 to 100. Each bar represents the average of 300 trials, conducted across three scenarios with varying maximum obstacle radii (0.3, 0.5 and 0.7 m). The results highlight that our approach outperforms other state-of-the-art CBF methods by maintaining a high success rate in dense environments where baselines frequently become infeasible.

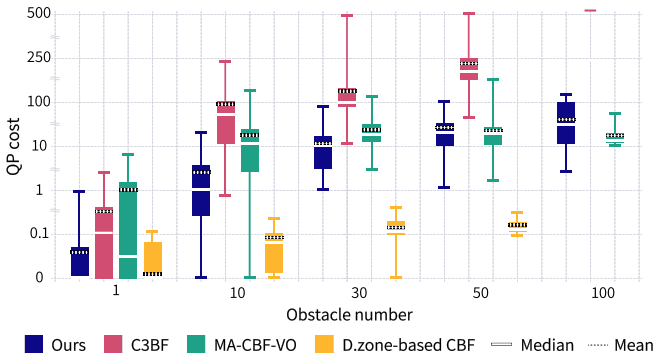

Fig. 9: Control intervention, measured by QP cost $\|\boldsymbol{u} - \boldsymbol{u}_{\text{ref}}\|_2^2$, is plotted against obstacle density for each method. Lower costs indicate greater efficiency and less conservative behavior.

constraints. The performance of the compared methods drops dramatically as the number of dynamic obstacles increases, resulting in frequent QP infeasibility or even collisions. Notably, DPCBF achieves a 100% success rate even in the 10-obstacle cases. This shows that the collision-cone based methods [15], [22] suffer in obstacle-dense environments, where overlapping collision cones severely constrain the feasible control space, leading to frequent QP failures.

Analysis on Conservatism. Fig. 9 details the QP cost for each method. Our DPCBF consistently exhibits the lower median and mean QP cost, navigating complex scenarios with minimal deviation from the reference controller. In contrast, C3BF requires the largest control interventions. This reveals a fundamental design limitation that becomes prominent in scenarios with multiple dynamic obstacles: overlapping collision cones overly shrink the safe set. Consequently, the QP with C3BF constraints is forced to either decelerate constantly to maintain the minimum velocity or take a large detour from the optimal trajectory, leading to a longer time to reach the goal. Although the Dynamic zone-based CBF appears to have the lowest QP cost, it is highly prone to infeasibility, as shown in Fig. 8. In scenarios with over 50 obstacles, its success rate drops to nearly 0%. Furthermore, while MA-CBF-VO shows a QP cost comparable to the proposed DPCBF, it has a higher infeasibility and collision rate. This is because its VO constraints are soft constraints that are often relaxed in obstacle-dense environments.

Qualitative Trajectory Analysis. Fig. 7 visualizes a challenging navigation scenario in which the DPCBF-based QP guides the robot through a dense group of 100 dynamic obstacles with a maximum obstacle radius of $r_{\text{obs,max}} = 0.7$ m. We also visualize the velocity of the kinematic bicycle along its trajectory, with the corresponding QP cost and control inputs shown in Fig. 10. This demonstrates that DPCBF constraints guide the CBF-QP to effectively adjust both longitudinal and lateral motion around multiple obstacles, successfully performing safe navigation. Importantly, at snapshots taken at $t = 10$ s and $t = 21$ s, the union of the

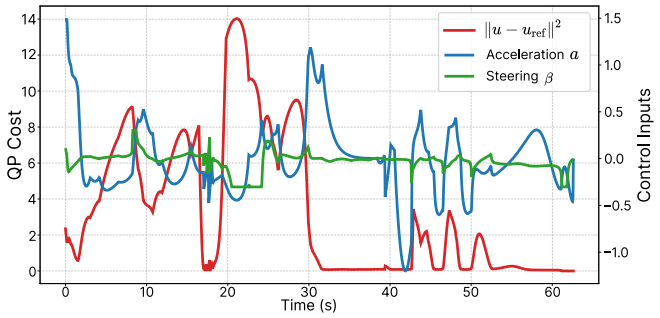

Fig. 10: QP cost and resulting control inputs over time for the dense scenario from Fig. 7. The peaks in QP cost near $t = 21$ s correspond to the most obstacle-dense moments of the trajectory shown in Fig. 7.

unsafe sets does not render the feasible space empty, whereas methods based on VO or collision cone would be infeasible in the same configurations. The robot is also able to regain high velocity at $t = 33$ s when the obstacles are no longer driving towards it. These examples highlight how DPCBF actively modifies the nominal control inputs to guarantee safety without being overly conservative.

V. CONCLUSION

In this paper, we introduced the Dynamic Parabolic Control Barrier Function (DPCBF), a novel CBF formulation for nonholonomic robots navigating in dynamic environments. By defining a safety boundary with a parabola that can adapt based on both relative distance and velocity, DPCBF generates a less conservative constraint that significantly improves the feasibility of the corresponding QP. Extensive simulations validated our approach, demonstrating higher navigation success rates in dense environments compared to state-of-the-art methods, particularly in challenging scenarios with up to 100 obstacles where cone-based approaches fail. Future work will focus on implementing DPCBF on physical hardware and investigating its extension to other complex dynamical systems.

REFERENCES

[1] 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, 2017.

[2] J. Zeng, B. Zhang, and K. Sreenath, "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," in *American Control Conference (ACC)*, 2021, pp. 3882–3889.

[3] T. Kim and D. Panagou, "Visibility-Aware RRT* for Safety-Critical Navigation of Perception-Limited Robots in Unknown Environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4508–4515, 2025.

[4] S. Zhang, O. So, K. Garg, and C. Fan, "GCBF+: A Neural Graph Control Barrier Function Framework for Distributed Safe Multi-Agent Control," *IEEE Transactions on Robotics*, vol. 41, pp. 1533–1552, 2025.

[5] W. Xiao and C. Belta, "Control Barrier Functions for Systems with High Relative Degree," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 474–479.

[6] K. Garg, J. Usevitch, J. Breeden, M. Black, D. Agrawal, H. Parwana, and D. Panagou, "Advances in the Theory of Control Barrier Functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems," *Annual Reviews in Control*, vol. 57, p. 100945, 2024.

[7] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

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

[9] M. Tayal, B. G. Goswami, K. Rajgopal, R. Singh, T. Rao, J. Keshavan, P. Jagtap, and S. Kolathaya, "A Collision Cone Approach for Control Barrier Functions," in *arXiv:2403.07043*, 2024.

[10] J. Huang, J. Zeng, X. Chi, K. Sreenath, Z. Liu, and H. Su, "Dynamic Collision Avoidance Using Velocity Obstacle-Based Control Barrier Functions," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 5, pp. 1601–1615, 2025.

[11] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Safe Policy Synthesis in Multi-Agent POMDPs via Discrete-Time Barrier Functions," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 4797–4803.

[12] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 812–818.

[13] S. He, J. Zeng, B. Zhang, and K. Sreenath, "Rule-Based Safety-Critical Control Design using Control Barrier Functions with Application to Autonomous Lane Change," in *American Control Conference (ACC)*, 2021, pp. 178–185.

[14] P. Thontepu, B. G. Goswami, M. Tayal, N. Singh, S. S. P I, S. S. M G, S. Sundaram, V. Katewa, and S. Kolathaya, "Collision Cone Control Barrier Functions for Kinematic Obstacle Avoidance in UGVs," in *Indian Control Conference (ICC)*, 2023, pp. 293–298.

[15] M. Tayal, R. Singh, J. Keshavan, and S. Kolathaya, "Control Barrier Functions in Dynamic UAVs for Kinematic Obstacle Avoidance: A Collision Cone Approach," in *American Control Conference (ACC)*, 2024, pp. 3722–3727.

[16] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, "ClearPath: highly parallel collision avoidance for multi-agent simulation," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009, pp. 177–187.

[17] M. Kim and J.-H. Oh, "Study on optimal velocity selection using velocity obstacle (OVVO) in dynamic and crowded environment," *Autonomous Robots*, vol. 40, no. 8, pp. 1459–1470, 2016.

[18] S. Samavati, M. Zarei, and M. T. Masouleh, "An optimal motion planning and obstacle avoidance algorithm based on the finite time velocity obstacle approach," in *Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, pp. 250–255.

[19] S. Sadraddini, S. Sivaranjani, V. Gupta, and C. Belta, "Provably Safe Cruise Control of Vehicular Platoons," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 262–267, 2017.

[20] A. Haraldsen, M. S. Wiig, A. D. Ames, and K. Y. Pattersen, "Safety-Critical Control of Nonholonomic Vehicles in Dynamic Environments Using Velocity Obstacles," in *American Control Conference (ACC)*, 2024, pp. 3152–3159.

[21] J. Breeden and D. Panagou, "Compositions of Multiple Control Barrier Functions Under Input Constraints," in *American Control Conference (ACC)*, 2023, pp. 3688–3695.

[22] A. S. Roncero, R. I. C. Muchacho, and P. Ögren, "Multi-Agent Obstacle Avoidance using Velocity Obstacles and Control Barrier Functions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 6638–6644.

[23] X. Wang, T. Kim, B. Hoxha, G. Fainekos, and D. Panagou, "Safe Navigation in Uncertain Crowded Environments Using Risk Adaptive CVaR Barrier Functions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 7669–7676.