Control Barrier Functions via Minkowski Operations for Safe Navigation among Polytopic Sets

Yi-Hsuan Chen[1], Shuo Liu[2], Wei Xiao[3], Calin Belta[4], Michael Otte[1]

Abstract— **Safely navigating around obstacles while respecting the dynamics, control, and geometry of the underlying system is a key challenge in robotics. Control Barrier Functions (CBFs) generate safe control policies by considering system dynamics and geometry when calculating safe forward-invariant sets. Existing CBF-based methods often rely on conservative shape approximations, like spheres or ellipsoids, which have explicit and differentiable distance functions. In this paper, we propose an optimization-defined CBF that directly considers the exact Signed Distance Function (SDF) between a *polytopic* robot and *polytopic* obstacles. Inspired by the Gilbert-Johnson-Keerthi (GJK) algorithm, we formulate both (i) minimum distance and (ii) penetration depth between polytopic sets as convex optimization problems in the space of Minkowski difference operations (the $\mathcal{M}_\mathcal{D}$-space). Convenient geometric properties of the $\mathcal{M}_\mathcal{D}$-space enable the derivatives of implicit SDF between two polytopes to be computed via differentiable optimization. We demonstrate the proposed framework in three scenarios including pure translation, initialization inside an unsafe set, and multi-obstacle avoidance. These three scenarios highlight the generation of a non-conservative maneuver, a recovery after starting in collision, and the consideration of multiple obstacles via pairwise CBF constraints, respectively. A video of all simulation animations can be found at** `https://youtu.be/3Dh0gtDW8bE` **.**

I. INTRODUCTION

Safely navigating an environment with obstacles is a fundamental challenge in robotics. Navigational solutions must respect the dynamics and control constraints of the vehicle and the geometric constraints of the vehicle and obstacles. Control Barrier Functions (CBFs) guarantee safety for control-affine systems by providing forward-invariant safe sets and combine naturally with CLFs, yielding a sequence of pointwise quadratic programs (QP) [1], [2]. QPs are computationally efficient and well-suited to real-time applications. Further advancements in safety-critical control with CBFs have introduced formulations for high relative degree constraints [3]–[7].

CBFs are used for obstacle avoidance across a range of applications [8]–[14], yet most CBFs approximate robots and/or obstacles by smooth shapes, i.e., hyperspheres [15] and ellipsoids [16], [17]. Hyperspheres and ellipsoids are
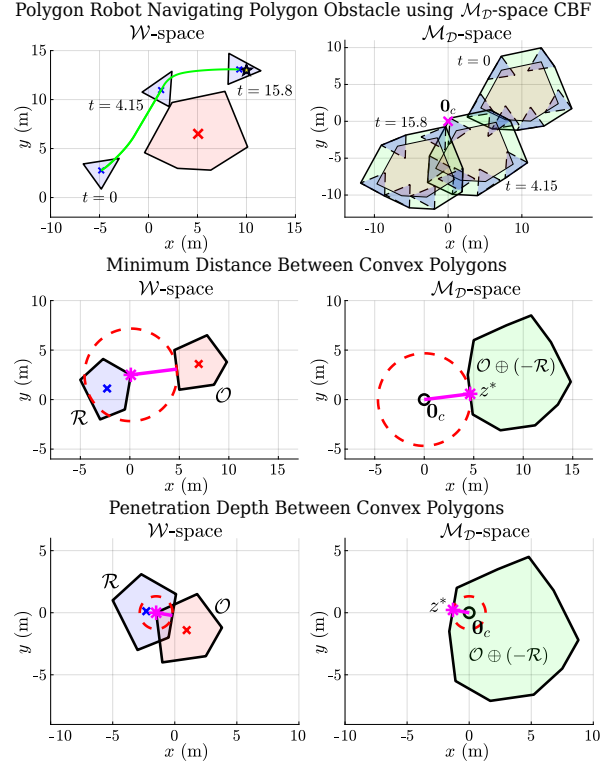
[1]Y. Chen and M. Otte are with the Dept. of Aerospace Engineering, University of Maryland, College Park, MD, USA. {yhchen91, otte}@umd.edu

[2]S. Liu is with the Dept. Mechanical Engineering, Boston University, Brookline, MA, 02215, USA. liushuo@bu.edu

[3]W. Xiao is with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA. weixy@mit.edu

[4]C. Belta is with the Dept. Electrical and Computer Engineering and Dept. Computer Science, University of Maryland, College Park, MD, USA. calin@umd.edu

Fig. 1: **Top:** A polytopic robot navigating around a polytopic obstacle in $\mathcal{W}$-space (Left) and the corresponding Minkowski differences in $\mathcal{M}_\mathcal{D}$-space (Right). The proposed CBF method leverages the fact that signed distance between polytopes in $\mathcal{W}$-space can be derived in $\mathcal{M}_\mathcal{D}$-space. **Middle/Bottom:** Distances between disjoint (Middle) and intersecting (Bottom) convex sets in $\mathcal{W}$-space (left), and the same distances in $\mathcal{M}_\mathcal{D}$-space (Right). In $\mathcal{M}_\mathcal{D}$-space the origin $\mathbf{0}_c$ lies outside or inside of the Minkowski difference if the signed distance is positive or negative, respectively. z^* is the critical point, whose derivative is used to enforce CBF constraints.

popular because they admit closed-form, differentiable robot-obstacle distance, but they are often overly conservative approximations of real-world objects; such objects are, for example, more accurately approximated by polytopes.

In this paper we present a CBF-based framework that considers *polytopic* robots and *polytopic* obstacles directly. Our method computes the exact signed distance function (SDF) between two polytopes by solving a convex optimization problem in the dual space that is defined by the Minkowski difference between the two polytopes (see Fig. 1). The resulting robot-to-obstacle SDF is differentiable almost everywhere (a.e.).

Our work builds on a foundation of prior work [18]–[23] that considers continuous-time CBFs for collision avoidance between polytopes/polygons. The first study of CBFs [18]

uses the Minimum Distance Function (MDF), solved by duality-based convex optimization, to construct a nonsmooth CBF (NCBF). In [18], [19] the exact derivative is applied in the case of strongly convex sets (but not polytopes, which are weakly convex), while a conservative lower bound on the CBF derivative is used for polytopes. A CBF-based filtering strategy for manipulators [20] uses the SDF and a local linear approximation of the CBF gradient (approximation error considered to ensure safety). An optimization-free smooth approx-SDF CBF is proposed in [23], where conservatism is reduced via hyperparameter tuning. In [22] CBFs are constructed using differentiable optimization of the minimum scaling factor (another metric for collision detection), and polygons are approximated by strictly convex shapes. Polygons are similarly approximated by polynomial shapes using logistic regression in [21]. Our work differs from prior work [18]–[23] in that we solve the SDF via convex optimization and compute its derivative using differentiable optimization *without approximation* by leveraging the configuration obstacle defined via the Minkowski difference. Table I provides a conceptual comparison with existing works.

The Minkowski difference is a fundamental tool in computational geometry for collision detection. It was first used in the Gilbert-Johnson-Keerthi (GJK) algorithm [24] to compute the distance between convex sets, and later extended by the Expanding Polytope Algorithm (EPA) [25] to compute penetration depth. Due to its effectiveness in encoding robot's geometry and configuration with obstacles, it has been widely used in sampling-based [26], [27] and optimization-based motion planning [28]–[30]. In this article, we refer to the space in which Minkowski difference lies as the "*Minkowski-difference-space*"[1], denoted as $\mathcal{M}_{\mathcal{D}}$-space.

While CBFs have previously considered $\mathcal{M}_{\mathcal{D}}$-space for collision avoidance [23], prior work relies on an approximation of the SDF by its lower bound. The integration of exact SDF computation via convex optimization with CBFs remains an open problem. To the best of our knowledge, we are the first to bridge this gap by combining the Minkowski difference, convex optimization, and CBFs in a unified framework that considers exact safety constraints.

The main contributions of this paper are as follows:

- We propose a novel approach to synthesize CBFs for obstacle avoidance between polytopic sets using the exact signed distance. This is achieved by solving the distance problem in the $\mathcal{M}_{\mathcal{D}}$-space. The derivatives are computed via differentiable optimization and are used to evaluate the derivatives of CBFs.
- We present an exact formulation for computing penetration depth as convex optimization. This reformulation allows us to use SDF as a CBF, as both minimum distance and penetration depth share the same differentiable optimization framework.

[1]While some authors refer to $\mathcal{M}_{\mathcal{D}}$-space as "configuration space," we use the term "Minkowski-difference-space" to distinguish it from the separate "configuration space" ($\mathcal{C}$-space) typically considered in motion planning. It is also chosen to be similar to the "collision space" defined in [29]. The more succinct "Minkowski space" has a specific different meaning in physics.

- We demonstrate the proposed method on single-integrator and unicycle models navigating around obstacles, including scenarios with initial collisions to validate the penetration depth formulation.

II. PRELIMINARIES

In this section we present background on CBFs (II-A), Minkowski operations (II-B), and signed distance (II-C).

A. Control Barrier Functions

Consider a nonlinear affine control system:

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

where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ ($\mathcal{X}$ denotes a closed state constraint set), $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times q}$ are locally Lipschitz, and $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^q$ ($\mathcal{U}$ denotes a closed control constraint set).

Definition 1 *(Forward invariant set):* A set $C \subset \mathbb{R}^n$ is forward invariant for system (1) if its solutions starting at $\boldsymbol{x}(t_0) \in C$ satisfy $\boldsymbol{x}(t) \in C, \ \forall t \geq t_0$.

Let C be the safe set, defined as the 0-superlevel set of continuously differentiable $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$:

$$C = \{\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}. \qquad (2)$$

Definition 2 *(CBF [2]):* Given a set C as in (2), $h(\boldsymbol{x})$ is a control barrier function (CBF) for system (1) if there exists an extended class $\mathcal{K}_\infty$ function[2] α such that, for all $\boldsymbol{x} \in \mathcal{X}$,

$$\sup_{\boldsymbol{u} \in \mathcal{U}}[L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x}))] \geq 0, \qquad (3)$$

where $L_f(\cdot), L_g(\cdot)$ denote the Lie derivatives along f and g, respectively. It is assumed that $L_g h(\boldsymbol{x}) \neq 0$ when $h(\boldsymbol{x}) = 0$.

Theorem 1 (Safety Guarantee from [2])**.** *Given a CBF h with the associated set C from (2), any Lipschitz continuous controller $\boldsymbol{u}(t) \in \mathcal{U}, t \geq t_0$ that satisfies (3) renders the set C forward invariant for (1).*

Remark 1. *Our SDF is continuously differentiable w.r.t. spatial variables but (only) differentiable a.e. w.r.t. orientation (due to active-set changes) and (only) when the robot and obstacle are not in collision. Similarly, in the cases of contact/penetration, the SDF is locally Lipschitz, and thus (only) differentiable a.e., such that non-differentiability occurs on a measure-zero set (vertices, interior medial axis, rotation-induced active-set switches). The classical CBF in Def. 2 does not apply at those (measure-zero set of) points. Consequently, the proofs in this paper (which build from the classical approach) fully apply to the non-collision case for single-integrator dynamics. We conjecture that a parallel analysis can be obtained a.e. for the collision and orientation-dependent cases; however, a rigorous nonsmooth analysis such as in [19], [31] is planned for future work.*

Definition 3 *(Control Lyapunov function [32]):* A continuously differentiable function $V : \mathcal{X} \to \mathbb{R}$ is a globally and exponentially stabilizing control Lyapunov function (CLF)

[2]An extended class $\mathcal{K}_\infty$ function is a function $\alpha : \mathbb{R} \to \mathbb{R}$ that is strictly increasing and $\alpha(0) = 0$, see, e.g., [1].

for system (1) if there exist $c_1 > 0, c_2 > 0, c_3 > 0$ such that $\forall \boldsymbol{x} \in \mathcal{X}, c_1 \|\boldsymbol{x}\|^2 \le V(\boldsymbol{x}) \le c_2 \|\boldsymbol{x}\|^2$ and

$$\inf_{\boldsymbol{u} \in \mathcal{U}} [L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x})] \le 0. \qquad (4)$$

Many existing works [2], [4], [33], unify CBF and CLF within a QP framework by relaxing the CLF with a slack variable penalized in the cost function to improve feasibility.

B. Collision Detection via Minkowski Operation

The workspace $\mathcal{W}$ of a robot is the physical (2D or 3D Euclidean) environment in which it operates, consisting of the space where the robot can move and the obstacles it must avoid. The configuration space ($\mathcal{C}$-space) represents all possible positions, orientations, etc, with each point defining a unique configuration. In contrast, the $\mathcal{M}_{\mathcal{D}}$-space is a dual of $\mathcal{W}$, and is a Euclidean space with the same (2D or 3D) dimensionality as $\mathcal{W}$. In $\mathcal{M}_{\mathcal{D}}$-space, the robot is shrunk to a point at the origin, and the obstacles are transformed to encode the robot's geometry and configuration (see Fig. 1).

Minkowski operations map problems defined in $\mathcal{W}$ to $\mathcal{M}_{\mathcal{D}}$-space, where distance computation (and thus collision detection) can be formulated directly [24], [25], [34], [35]. Given two sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$, their Minkowski sum is defined as the set of all pairwise sums of points taken from each set:

$$\mathcal{A} \oplus \mathcal{B} = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\}, \qquad (5)$$

where $\boldsymbol{a}, \boldsymbol{b}$ are position vectors in sets of $\mathcal{A}, \mathcal{B}$, respectively.

Definition 4 (*Configuration Obstacle (CO) [36]*): Given a robot $\mathcal{R}$ and an obstacle $\mathcal{O}$, the corresponding CO is:

$$\mathcal{C}_{\mathcal{R}}(\mathcal{O}) = \mathcal{O} \oplus (-\mathcal{R}), \qquad (6)$$

where $-\mathcal{R} = \{-\boldsymbol{p} \mid \boldsymbol{p} \in \mathcal{R}\}$ is the reflection of $\mathcal{R}$ through the origin, where $\boldsymbol{p}$ is a position vector of $\mathcal{R}$. Note that we refer to the Minkowski sum of $\mathcal{O}$ and $-\mathcal{R}$ as the *Minkowski difference*, following collision-detection convention.[3]

Lemma 1 (Collision between Convex Sets [24]). *A robot $\mathcal{R}$ and an obstacle $\mathcal{O}$ intersect (are in collision) if and only if their CO, $\mathcal{C}_{\mathcal{R}}(\mathcal{O})$, contains the origin $\boldsymbol{0}_c$.*

Remark 2. *While the Minkowski sum (5) and CO computation (6) are general operations that apply to any shape, it is worth emphasizing that Lem. 1 holds only for convex sets. For well-behaved non-convex sets (e.g., non-fractal) that can be decomposed into convex subsets, a CO is computed for each convex-robot-convex-obstacle pair, and Lem. 1 applied iteratively. In this paper, we consider nonempty convex polygons for both robot and obstacle.*

Lemma 2 (Properties of the CO [37]). *Given convex, nonempty polygons $\mathcal{R}$ and $\mathcal{O}$ with ℓ_r and ℓ_o edges. Their CO, $\mathcal{O} \oplus (-\mathcal{R})$, is a convex, nonempty polygon with $\le \ell_r + \ell_o$ edges, and can be computed in $O(\ell_r + \ell_o)$ time.*

By Lem. 1 and Lem. 2, the collision detection between two convex polygons reduces to checking whether the origin lies inside their CO, as illustrated in Fig. 1.

[3]In fields like mathematics and image processing, the Minkowski difference has a different (alternative) definition, where $\mathcal{A} \ominus \mathcal{B}$ is the set that, when added to $\mathcal{B}$, recovers $\mathcal{A}$.

C. Signed distance for Collision Avoidance

While it may be sufficient to use the minimum distance to enforce safety in some scenarios, this measure alone is incapable of addressing situations where the robot is already in contact with an obstacle. To fully characterize the spatial relationship between two sets—whether disjoint or intersecting—the notion of signed distance[4] [38] is required.

Definition 5 (*Signed Distance via CO*): The signed distance between a robot $\mathcal{R}$ and an obstacle $\mathcal{O}$ is defined as

$$\text{sd}(\mathcal{R}, \mathcal{O}) = \text{dist}(\mathcal{R}, \mathcal{O}) - \text{pd}(\mathcal{R}, \mathcal{O}), \qquad (7)$$

where $\text{dist}(\cdot, \cdot)$ and $\text{pd}(\cdot, \cdot)$ are the minimum distance and penetration depth, and defined [39] as

$$\begin{aligned} \text{dist}(\mathcal{R}, \mathcal{O}) &= \min\{\|\boldsymbol{t}\| \mid (\mathcal{R} + \boldsymbol{t}) \cap \mathcal{O} \ne \varnothing\} \\ &= \min\{\|\boldsymbol{t}\| \mid (\boldsymbol{0}_c + \boldsymbol{t}) \cap \mathcal{C}_{\mathcal{R}}(\mathcal{O}) \ne \varnothing\}, \\ \text{pd}(\mathcal{R}, \mathcal{O}) &= \min\{\|\boldsymbol{t}\| \mid (\mathcal{R} + \boldsymbol{t}) \cap \mathcal{O} = \varnothing\} \\ &= \min\{\|\boldsymbol{t}\| \mid (\boldsymbol{0}_c + \boldsymbol{t}) \cap \mathcal{C}_{\mathcal{R}}(\mathcal{O}) = \varnothing\}. \end{aligned} \qquad (8)$$

The signed distance is positive when $\mathcal{R}$ and $\mathcal{O}$ are disjoint and negative when they intersect. Thus, collision avoidance can be enforced by ensuring $\text{sd}(\mathcal{R}, \mathcal{O}) > 0$. The signed distance between two sets can equivalently be expressed as the signed distance between the origin $\boldsymbol{0}_c$ and their CO. The closest related work [18] constructs CBFs based on implicit MDFs by posing a convex program with dual variables, and enforces the resulting NCBF with a linear program (LP) that imposes a lower bound on the CBF derivative. Moreover, [18] considers only minimum distance and not penetration depth (cases where the system starts in collision). In contrast, we propose a new framework for SDF-based CBFs that incorporates penetration depth, as detailed in Sec. IV.

III. PROBLEM FORMULATION

The robot's state is denoted $\boldsymbol{x}$. Both robot and obstacles are defined using the halfspace-representation (H-rep), $\mathcal{R}(\boldsymbol{x}) := \{y \in \mathbb{R}^d \mid A_r(\boldsymbol{x})y \le b_r(\boldsymbol{x})\}$ and $\mathcal{O}_i := \{y \in \mathbb{R}^d \mid A_{o_i} y \le b_{o_i}\}$, with $A_r \in \mathbb{R}^{\ell_r \times d}$, $b_r \in \mathbb{R}^{\ell_r}$, $A_{o_i} \in \mathbb{R}^{\ell_{o_i} \times d}$, $b_{o_i} \in \mathbb{R}^{\ell_{o_i}}$, and $d \in \{2, 3\}$ for 2D and 3D spaces. ℓ_r and ℓ_{o_i} denote the number of edges of the robot and the i-th obstacle, respectively. The corresponding COs are denoted as $\mathcal{O}_i^{\mathcal{C}}(\boldsymbol{x}) = \{y \in \mathbb{R}^d \mid A_i^{\mathcal{C}}(\boldsymbol{x})y \le b_i^{\mathcal{C}}(\boldsymbol{x})\}$, omitting the subscript $\mathcal{R}$ for brevity, where $A_i^{\mathcal{C}} \in \mathbb{R}^{\ell_{\mathcal{C}_i} \times d}$, $b_i^{\mathcal{C}} \in \mathbb{R}^{\ell_{\mathcal{C}_i}}$, and $\ell_{\mathcal{C}_i}$ denotes the number of edges of the i-th CO.

By abuse of notation, we write $\mathcal{R} = \mathcal{R}(\boldsymbol{x})$ and $\mathcal{O}_i^{\mathcal{C}} = \mathcal{O}_i^{\mathcal{C}}(\boldsymbol{x})$. The CO is specific to the obstacle and the robot's current configuration, and thus depends on $\boldsymbol{x}$. In our framework, the CO is used to compute either the minimum distance or the penetration depth, depending on whether the origin lies outside or inside the CO, respectively.

We make the following assumptions:
1. The robot $\mathcal{R}$ is a rigid body, is uniformly bounded, and has nonempty interior, $\forall \boldsymbol{x} \in \mathcal{X}$.
2. $A_i^{\mathcal{C}}(\boldsymbol{x}), b_i^{\mathcal{C}}(\boldsymbol{x})$ are differentiable a.e., and the inequality constraints provide a minimal representation of the CO.

[4]Similar notion appears as the minimal translational distance in [34].

TABLE I: Conceptual comparison of existing CBF-based collision avoidance methods for polytopic sets.

Approaches	Metric	Explicitness	Smoothness	Conservatism	Obstacle Defined in	Collision Recovery Demonstrated
MDF-based CBF [18], [19]	Distance	Implicit	Nonsmooth	Nonconservative (strongly convex sets) Conservative (polytopes)	Workspace	No
SDF-linearized CBF [20]	Distance	Implicit	Smooth	Conservative	Workspace	No
Polynomial CBF [21]	Distance	Explicit	Smooth	Tunable	Workspace	No
Diff-opti CBF [22]	Scaling factor	Implicit	Smooth	Tunable	Workspace	No
Opti-free CBF [23]	Distance	Explicit	Smooth	Tunable	$\mathcal{M}_\mathcal{D}$-space	No
Ours	Distance	Implicit	Nonsmooth	Nonconservative	$\mathcal{M}_\mathcal{D}$-space	Yes

Implicit indicates that CBFs are defined via optimization [18], [21], [22], or computed algorithmically [20]. *Conservatism* indicates whether the method uses approximations or lower bounds in its formulation. *Collision recovery* refers to the ability to recover from an initial collision with an obstacle; this property is theoretically enabled by SDF-based CBFs [20], [23] but has not been demonstrated in prior work. Abbreviations: diff = differentiable, opti = optimization.

For almost all $\boldsymbol{x} \in \mathcal{X}$, $\ell_{\mathcal{C}_i} = \ell_r + \ell_{o_i}$. Note that, if $\mathcal{R}$ and $\mathcal{O}_i$ have parallel edges, the associated constraints $A_i^\mathcal{C}$, $b_i^\mathcal{C}$ may lose one/more rows, depending on the number of parallel edges. Yet, such degeneracy occurs only on a measure zero set and thus does not affect the optimization for almost all $\boldsymbol{x}$ (as reported in [38]).

3. All obstacles are static, uniformly bounded, and have nonempty interior. Also, the information about obstacles is accessible to the robot.

We define the problem using the robot $\mathcal{R}$ and a single obstacle $\mathcal{O}$, omitting the subscript i for simplicity.

Problem 1. *Given a robot $\mathcal{R}$ with known affine dynamics in the form of* (1) *and a static obstacle $\mathcal{O}$, find a state-feedback optimal controller $\boldsymbol{u}^*(\boldsymbol{x})$ that both*

1) *achieves objectives: minimizes energy consumption $J = \int_0^T \boldsymbol{u}^\top \boldsymbol{u} dt$, and ensures the robot reaches the desired position (x_d, y_d),*

2) *satisfies constraints: safety — requiring $sd(\mathcal{R}, \mathcal{O}) > d_{safe}$, where $d_{safe} \geq 0$ is the user-defined safety margin, and actuator limitations $\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}$.*

Remark 3. *This formulation extends to multiple obstacles by applying the same safety constraint independently to each pair $(\mathcal{R}, \mathcal{O}_i)$. Additionally, non-convex polygons can be represented as a union of multiple convex polygons.*

We now develop a framework for constructing CBFs based on the SDF without introducing geometric approximations.

IV. SDF-BASED CBFs FOR POLYTOPES IN $\mathcal{M}_\mathcal{D}$-SPACE

This section presents our main result of constructing CBFs based on the "exact" SDF by recasting its computation as convex optimization in $\mathcal{M}_\mathcal{D}$-space. The derivative of the SDF is then computed using differentiable convex optimization.

A. Optimization-Based SDF Formulation in $\mathcal{M}_\mathcal{D}$-space

The minimum distance between $\mathcal{R}$ and $\mathcal{O}$ when the two shapes are disjoint ($\mathcal{R} \cap \mathcal{O} = \varnothing$) can be computed by solving a QP in the workspace:

$$\min \ \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{subject to} \quad A_r \mathbf{x} \leq b_r, \ A_o \mathbf{y} \leq b_o$$
$$\text{with} \ \text{dist}(\mathcal{R}, \mathcal{O}) = \|\mathbf{x}^* - \mathbf{y}^*\|_2, \tag{9}$$

where $\|\cdot\|_2$ is the Euclidean norm and $(\mathbf{x}^*, \mathbf{y}^*)$ is the optimal solution pair to (9). The distance between sets can be reformulated as the distance between a point and a set [40],

$\text{dist}(\mathcal{R}, \mathcal{O}) = \text{dist}(\mathbf{0}_c, \mathcal{O} \oplus (-\mathcal{R}))$, where $\mathcal{O} \oplus (-\mathcal{R})$ is their CO. We formulate the equivalent problem in $\mathcal{M}_\mathcal{D}$-space as

$$\min \ \|\boldsymbol{z}\|_2^2 \quad \text{subject to} \quad A^\mathcal{C} \boldsymbol{z} \leq b^\mathcal{C}$$
$$\text{with} \ \text{dist}(\mathbf{0}_c, \mathcal{O}^\mathcal{C}) = \|\boldsymbol{z}^*\|_2, \tag{10}$$

where $\boldsymbol{z}^*$ is the optimal solution of (10) and it can also be interpreted as the projection[5] of the origin $\mathbf{0}_c$ on $\mathcal{O}^\mathcal{C}$.

Similarly, the penetration depth $\text{pd}(\mathcal{R}, \mathcal{O})$ can be interpreted as the problem of finding the minimum translation vector (MTV) that separates two intersecting convex sets. It can be conceptually formulated as an optimization problem in the workspace (a similar formulation appears in [41]) as

$$\text{pd}(\mathcal{R}, \mathcal{O}) = \min \ \|\boldsymbol{z}\|_2 \quad \text{subject to} \quad \mathcal{R}' \cap \mathcal{O} = \varnothing, \tag{11}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is the MTV, and $\mathcal{R}' = \{y \in \mathbb{R}^d \mid A_r y \leq b_r + A_r \boldsymbol{z}\}$ is the translation of $\mathcal{R}$ by $\boldsymbol{z}$. However, unlike in the minimum distance case, it is unclear how to represent the separation constraint $\mathcal{R}' \cap \mathcal{O} = \varnothing$ as a convex constraint.

To address this issue, we propose a reformulation of (11) as LP in $\mathcal{M}_\mathcal{D}$-space by leveraging the notion of the depth[6] of a point relative to a convex set [40]

$$\text{depth}(\mathbf{0}_c, \mathcal{O}^\mathcal{C}) = \max \ s$$
$$\text{subject to} \ s \geq 0, \ s\|A_k^\mathcal{C}\|_2 \leq b_k^\mathcal{C}, \ k = 1, \ldots, \ell_\mathcal{C}, \tag{12}$$

where $s \in \mathbb{R}$ is the minimal translational distance required to separate two overlapping sets. The vector $A_k^\mathcal{C}$ and scalar $b_k^\mathcal{C}$, denoting the k-th row $A^\mathcal{C}$ and $b^\mathcal{C}$, respectively, define the bounding hyperplanes of $\mathcal{O}^\mathcal{C}$. It is shown in [25], [34] that $\text{pd}(\mathcal{R}, \mathcal{O}) = \text{depth}(\mathbf{0}_c, \mathcal{O}^\mathcal{C})$.

However, while (12) gives the penetration depth (scalar), we also need the direction of separation to recover the optimal solution $\boldsymbol{z}^*$ of (11), which cannot be solved directly. This direction can be obtained by identifying the active constraint from the dual variable of (12). Let k_{act} be the set of active indices, and $A_{k_{\text{act}}}^\mathcal{C}$ be the matrix containing the rows corresponding to the active constraints at the optimal solution s^*. By strong duality in LP, complementary slackness holds and implies that nonzero dual variables $\lambda_{k_{\text{act}}}$ correspond to the active constraints.

In most cases, a single constraint is active, so $A_{k_{\text{act}}}^\mathcal{C}$ reduces to one row vector, denoted as $a_{k_{\text{act}}}^\mathcal{C}$, which uniquely determines the separation direction. When multiple constraints are

---

[5] The projection of a point $\boldsymbol{x}$ on a set C is any point $\boldsymbol{z} \in C$ that is closest to $\boldsymbol{z}$, i.e., satisfies $\|\boldsymbol{z} - \boldsymbol{x}\| = \text{dist}(\boldsymbol{x}, C)$.

[6] The depth of a point $\boldsymbol{x}$ on a convex set C is the radius of the largest ball, centered at $\boldsymbol{x}$, that lies entirely within C.

active, $A_{k_{\mathrm{act}}}^{\mathcal{C}}$ becomes a matrix. As the origin is equidistant from multiple active planes, the separating direction lies in the convex cone of their normals. The presence of multiple active constraints occurs only for a measure-zero set of configurations. Moreover, if such a case occurs in practice, any active constraint yields a valid separating direction. Once the active constraint of (12) is identified, corresponding to the hyperplane $\mathcal{H} = \{y \in \mathbb{R}^d \mid a_{k_{\mathrm{act}}}^{\mathcal{C}} y = b_{k_{\mathrm{act}}}^{\mathcal{C}}\}$, $\boldsymbol{z}^*$ can be computed by projecting the origin $\mathbf{0}_c$ onto $\mathcal{H}$ [40] as

$$\boldsymbol{z}^* = \mathrm{proj}_{\mathcal{H}}(\mathbf{0}_c) = b_{k_{\mathrm{act}}}^{\mathcal{C}} (a_{k_{\mathrm{act}}}^{\mathcal{C}})^\top / \|a_{k_{\mathrm{act}}}^{\mathcal{C}}\|_2^2. \tag{13}$$

The signed distance (7) is computed by solving either the QP in (10) or the LP in (12), with feasibility guaranteed by the assumption 1. and Lem. 2. The equivalence of computing the signed distance in both $\mathcal{W}$- and $\mathcal{M}_{\mathcal{D}}$-spaces is shown in Fig. 1. The key advantages of formulating the optimization problems in $\mathcal{M}_{\mathcal{D}}$-space are: (1) it enables convex optimization (LP) formulation for penetration depth computation; and (2) the minimum distance and penetration depth can be viewed as companion problems that provide a generalized measure of distance. This allows us to compute derivative of the signed distance using the same differentiable optimization framework, see Sec. IV-C.

B. CBF Construction Using Signed Distance

The point $\boldsymbol{z}^*$, defined in $\mathcal{M}_{\mathcal{D}}$-space and obtained from our optimization-based signed distance formulation, is referred to as the critical point, as its derivative is used in computing the derivative of CBF. Geometrically, $\boldsymbol{z}^*$ lies on the boundary of the CO, $\partial \mathcal{O}^{\mathcal{C}}$, and is the closest point to the origin $\mathbf{0}_c$ in $\mathcal{M}_{\mathcal{D}}$-space. Thus, the signed distance is equal to $\pm \|\boldsymbol{z}^*\|_2$.

Based on this, we propose the following CBF candidate:

$$h(\boldsymbol{x}) = \mathrm{sd}(\mathcal{R}(\boldsymbol{x}), \mathcal{O}) - d_{\mathrm{safe}} = \mathrm{sd}(\mathbf{0}_c, \mathcal{O}^{\mathcal{C}}(\boldsymbol{x})) - d_{\mathrm{safe}},$$

$$\text{where } \mathrm{sd}(\mathcal{R}(\boldsymbol{x}), \mathcal{O}) = \begin{cases} \|\boldsymbol{z}^*(\boldsymbol{x})\|_2 & \text{if } \mathbf{0}_c \notin \mathcal{O}^{\mathcal{C}} \\ -\|\boldsymbol{z}^*(\boldsymbol{x})\|_2 & \text{if } \mathbf{0}_c \in \mathcal{O}^{\mathcal{C}} \end{cases}, \tag{14}$$

where $\boldsymbol{z}^*(\boldsymbol{x}) \in \mathbb{R}^d$ is the optimal solution to (10) if $\mathcal{R} \cap \mathcal{O} = \varnothing$, and can be indirectly found by solving (12) with its dual variable analysis in the case of $\mathcal{R} \cap \mathcal{O} \neq \varnothing$. Recall that both the critical point $\boldsymbol{z}^*(\boldsymbol{x})$ and the CO, $\mathcal{O}^{\mathcal{C}}(\boldsymbol{x})$, depend on the robot's state $\boldsymbol{x}$. The associated safe set of the system is defined as $C = \{\boldsymbol{x} \in \mathbb{R}^n : \mathrm{sd}(\mathbf{0}_c, \mathcal{O}^{\mathcal{C}}(\boldsymbol{x})) \geq d_{\mathrm{safe}}\}$. Note that the SDF-based CBF (14) is implicit and is obtained from an optimization that depends on system state $\boldsymbol{x}$. It is important to note that the SDF in Euclidean space is differentiable a.e. [20], which means it is differentiable at all points except for a set of (Lebesgue) measure zero.

Lemma 3. *Let $\mathcal{X}_f = \{x \in \mathcal{X} \mid \mathcal{R}(\boldsymbol{x}) \cap \mathcal{O} = \varnothing\}$. The optimal solution $\boldsymbol{z}^*(\boldsymbol{x})$ of (10) is unique for every $\boldsymbol{x} \in \mathcal{X}_f$, and $\boldsymbol{z}^*(\cdot)$ is continuous and differentiable a.e. on $\mathcal{X}_f$.*

Proof. By Lem. 2, the CO set $\mathcal{O}^{\mathcal{C}}(\boldsymbol{x})$ is nonempty, closed, and convex, and therefore, the Euclidean projection of the origin $\mathbf{0}_c$ on the set $\mathcal{O}^{\mathcal{C}}(\boldsymbol{x})$ is unique [40, Chap. 8]. As (10) is a strictly convex QP, its solution $\boldsymbol{z}^*(\boldsymbol{x})$ is continuous and differentiable a.e. based on [42, Thm. 1]. ∎

Remark 4. *Extending Lemma 3 to the penetration depth case is deferred to future work. The min-distance case is a strictly convex QP, while penetration depth is an LP, so the differentiability results of [42] do not directly apply. The min-distance and penetration depth cases are complementary, and we focus on the min-distance case in the next subsection. While our LP formulation performs well empirically and yields stable derivative estimates under the same $\mathcal{M}_{\mathcal{D}}$-space framework, a rigorous differentiability analysis of the penetration depth case is planned for future work.*

C. Derivative Calculation Using Differentiable Optimization

To enforce the CBF constraint on system dynamics, we need to find the derivative of $h(\boldsymbol{x})$. However, the CBF $h(\boldsymbol{x})$ (14) depends on the solution to an optimization problem $\boldsymbol{z}(\boldsymbol{x})$, defined implicitly to state $\boldsymbol{x}$. Therefore, we cannot differentiate it w.r.t. $\boldsymbol{x}$ directly. Given this implicit dependence, we first apply the chain rule to express $\partial h / \partial \boldsymbol{x}$ as:

$$\frac{\partial h}{\partial \boldsymbol{x}} = \frac{\partial h}{\partial \boldsymbol{z}^*} \frac{\partial \boldsymbol{z}^*}{\partial \boldsymbol{x}}, \tag{15}$$

where $\partial h / \partial \boldsymbol{z}^* = \boldsymbol{z}^{*\top} / (h + d_{\mathrm{safe}}) \in \mathbb{R}^{1 \times d}$. The second term $\partial \boldsymbol{z}^* / \partial \boldsymbol{x} \in \mathbb{R}^{d \times n}$ is the Jacobian of the optimal solution $\boldsymbol{z}^*$ to (10), which describes how it changes as inequality constraints of CO vary with the state $\boldsymbol{x}$.

Corollary 1. *Let $\boldsymbol{z}^*(\boldsymbol{x})$ be the optimal solution to the QP (10) for $\boldsymbol{x} \in \mathcal{X}_f$. Its Clarke generalized subdifferential [43] $\partial \boldsymbol{z}^*(\boldsymbol{x})$ is nonempty for all $\boldsymbol{x} \in \mathcal{X}_f$, and has a single unique Jacobian for all but a measure zero set of points.*

Proof. This follows directly from [42, Thm. 1], as the objective function of QP (10) is strictly convex. ∎

Since (10) is a QP in standard form, the Jacobian can be computed using the framework introduced in [42]. We start by finding the Karush-Kuhn-Tucker (KKT) conditions of (10), where the stationarity and complementary slackness conditions can be compactly expressed in matrix form as:

$$G(\boldsymbol{z}, \boldsymbol{\lambda}) = \begin{bmatrix} 2I_d \boldsymbol{z} + (A^{\mathcal{C}})^\top \boldsymbol{\lambda} \\ D(\boldsymbol{\lambda})(A^{\mathcal{C}} \boldsymbol{z} - b^{\mathcal{C}}) \end{bmatrix},$$

where I_d is the d-dimensional identity matrix, $\boldsymbol{\lambda} \in \mathbb{R}^{\ell_c}$ is the dual variable of (10), and $D(\cdot)$ creates a diagonal matrix from a vector. Let $\boldsymbol{\xi} := (\boldsymbol{z}, \boldsymbol{\lambda})$. Thus, a convex optimization solver can be viewed as a root-finding method that seeks the optimal solution $\boldsymbol{\xi}^*(\boldsymbol{x})$ by solving $G(\boldsymbol{\xi}^*) = 0$ (suppressing the dependency of $\boldsymbol{\xi}$ on $\boldsymbol{x}$ for brevity). The partial derivative of $\boldsymbol{z}^*$ w.r.t. $\boldsymbol{x}$ can be computed by differentiating the KKT conditions using the implicit function theorem [44]:

$$D_{\boldsymbol{x}} G(\boldsymbol{\xi}^*) = 0 \ \Rightarrow \ \partial_{\boldsymbol{\xi}} G(\boldsymbol{\xi}^*) \, \partial_{\boldsymbol{x}}(\boldsymbol{\xi}^*) + \partial_{\boldsymbol{x}} G(\boldsymbol{\xi}^*) = 0.$$

Rearranging the matrix equation yields

$$\partial_{\boldsymbol{x}}(\boldsymbol{\xi}^*) = -[\partial_{\boldsymbol{\xi}} G(\boldsymbol{\xi}^*)]^{-1} \partial_{\boldsymbol{x}} G(\boldsymbol{\xi}^*), \tag{16}$$

$$\partial_{\boldsymbol{\xi}} G(\boldsymbol{\xi}^*) = \begin{bmatrix} 2I_d & (A^{\mathcal{C}})^\top \\ D(\boldsymbol{\lambda}^*) A^{\mathcal{C}} & D(A^{\mathcal{C}} \boldsymbol{z}^* - b^{\mathcal{C}}) \end{bmatrix},$$

$$\partial_{\boldsymbol{x}} G(\boldsymbol{\xi}^*) = \begin{bmatrix} d_{\boldsymbol{x}}(A^{\mathcal{C}})^\top \boldsymbol{\lambda}^* \\ D(\boldsymbol{\lambda}^*) \left(d_{\boldsymbol{x}} A^{\mathcal{C}} \boldsymbol{z}^* - d_{\boldsymbol{x}} b^{\mathcal{C}} \right) \end{bmatrix}, \tag{17}$$

where $d_{\boldsymbol{x}}A^{\mathcal{C}} \in \mathbb{R}^{\ell_{\mathcal{C}} \times d \times n}$ is a 3D tensor and $d_{\boldsymbol{x}}b^{\mathcal{C}} \in \mathbb{R}^{\ell_{\mathcal{C}} \times n}$ represents how the bounding constraints of CO, $A^{\mathcal{C}}, b^{\mathcal{C}}$, changes w.r.t. state $\boldsymbol{x}$. Note that $\boldsymbol{z}^*, \boldsymbol{\lambda}^*, A^{\mathcal{C}}, b^{\mathcal{C}}$ all depend on the state $\boldsymbol{x}$, which we denote compactly as $\boldsymbol{z}^*(\boldsymbol{x}), \boldsymbol{\lambda}^*(\boldsymbol{x}), A^{\mathcal{C}}(\boldsymbol{x}), b^{\mathcal{C}}(\boldsymbol{x})$. The Jacobian of the optimal solution $\boldsymbol{z}^*$ can be extracted from the associated block (16), which also provides the derivatives of dual variables. The partial derivative of h can be computed using (15) to (17).

Remark 5. *As (17) is derived by differentiating the KKT conditions; complementary slackness holds. Thus, it is sufficient to consider only the active constraints when computing the derivatives. That is, we can replace $A^{\mathcal{C}}, b^{\mathcal{C}}$ with $A^{\mathcal{C}}_{k_{act}}, b^{\mathcal{C}}_{k_{act}}$.*

D. Configuration Obstacle Computation

Consider a polygonal robot modeled as a rigid body in SE(2), with states (position and orientation) is $\boldsymbol{x} = [x, y, \theta]^\top$, and a nonzero initial configuration $\boldsymbol{x}_0 = [x_0, y_0, \theta_0]^\top$. Its shape in H-rep can be parameterized by its configuration in $\mathcal{W}$-space as $\mathcal{R}(\boldsymbol{x}) = \{y \mid A(\boldsymbol{x})y \leq b(\boldsymbol{x})\}$, where $A(\boldsymbol{x}) = AR(\theta - \theta_0)^\top$, $b(\boldsymbol{x}) = b + A(\boldsymbol{x})\boldsymbol{p} - A\boldsymbol{p}_0$ with $\boldsymbol{p} = [x, y]^\top$, $\boldsymbol{p}_0 = [x_0, y_0]^\top$, and $A(\boldsymbol{x}_0), b(\boldsymbol{x}_0)$ defining the robot's shape at the initial configuration. $R(\theta) = [\cos\theta, -\sin\theta; \sin\theta, \cos\theta]$ is a rotation matrix.

However, this formulation does not extend directly to the CO in $\mathcal{M}_{\mathcal{D}}$-space. In the special case of pure translation, the CO's shape remains unchanged, and its bounding hyperplanes translate in the opposite direction of the robot's motion by definition (6). Thus, the derivatives of the CO bounding constraints w.r.t. position $\boldsymbol{p}$, have a closed-form expression: $d_{\boldsymbol{x}}A^{\mathcal{C}} = \boldsymbol{0}_{\ell_{\mathcal{C}} \times 2 \times 2}$, $d_{\boldsymbol{x}}b^{\mathcal{C}} = -A^{\mathcal{C}}$.

In contrast, with rotation the bounding hyperplanes change both slope and order, causing nontrivial CO deformations. While the Minkowski sum can be computed via linear projection [45], this method still does not provide an analytical expression for CO. Deriving an explicit relationship between the robot-obstacle representations in $\mathcal{W}$-space and the CO in $\mathcal{M}_{\mathcal{D}}$-space remains an open problem for future research.

E. Controller Synthesis Using CLF-CBF-QP

We implement (18) by discretizing the horizon $[0, T]$ into equal intervals of Δt, holding the state fixed at the start of each step. At time step $t = k\Delta t$, the control is held constant over $[t_k, t_{k+1})$, rendering the constraints affine in $\boldsymbol{u}$. We then solve the standard CLF-CBF-QP [2] with $\boldsymbol{x}(t_k)$ to compute the optimal control $\boldsymbol{u}(t_k) = \boldsymbol{u}_k$, and apply $\boldsymbol{u}_k$ on $[t_k, t_{k+1})$ to propagate the dynamics (1).

$$\min_{\boldsymbol{u}(t), \delta(t)} \int_0^T \boldsymbol{u}(t)^\top \boldsymbol{u}(t) + p\delta(t)^2$$

$$\text{subject to} \quad L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \gamma h(\boldsymbol{x}) \geq \epsilon, \quad (18)$$
$$L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + cV(\boldsymbol{x}) \leq \delta,$$
$$\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max},$$

where δ is a slack variable, used to relax the CLF constraint and $p > 0$ is a penalty. $\epsilon > 0$ is a small constant to ensure the CBF will change sign when starting in the unsafe set in

finite time. We refer the interested readers to [6] for detailed discussion on finite-time convergence guarantee.

V. CASE STUDY AND IMPLEMENTATION

We demonstrate the proposed CBFs by solving Problem 1 in three case studies. We consider single-integrators (V-A) and unicycles that start in collision (V-B.1) or involve multiple obstacles (V-B.2). All simulations are conducted in MATLAB, using 'quadprog' to solve both signed distance and CLF-CBF-QP problems, and 'ode45' to integrate system dynamics at a sampling rate of 100 Hz, on a PC with an Intel i7-8700 6-core of 3.2 GHz CPU and 32 GB of RAM. (See video at https://youtu.be/3Dh0gtDW8bE .)

A. Case I: Single-Integrator Model (Pure Translation)

We first consider the scenario of a triangular robot $\mathcal{R}$ with single-integrator dynamics translating to the desired position $\boldsymbol{x}_d$ while avoiding a polygonal obstacle $\mathcal{O}$: $\boldsymbol{x} = [x, y]^\top$, $\dot{\boldsymbol{x}} = \boldsymbol{u} = [u_1, u_2]^\top$, where x, y denote the position, and u_1, u_2 are the control inputs (linear velocities). The dynamics can be rewritten in form of (1) with $f(\boldsymbol{x}) = \boldsymbol{0}_{2\times1}$ and $g(\boldsymbol{x}) = \boldsymbol{I}_2$. We use the proposed SDF-CBF (14) to impose safety and a CLF $V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_d)^2$ to reach $\boldsymbol{x}_d = (5, 8)$. The other parameters are $\boldsymbol{x}(0) = (-4.83, 0.77), \boldsymbol{u}_{\max} = -\boldsymbol{u}_{\min} = $ 5m/s, $d_{\text{safe}} = 0, p = 10, c = 1, \gamma = 3, \epsilon = 0$. The derivative of the CBF is computed using (16) and (17). Each control loop involves computing the CO (6) and solving a QP for minimum distance (10) and CLF-CBF-QP (18). Loops run at 100 Hz and take 7.38 ms on average. The simulation results (Fig. 2) demonstrate the non-conservativeness of the proposed CBF, i.e., the robot safely moves in close proximity to the obstacle boundary.

B. Case II: Unicycle Model

Now we consider the unicycle dynamics defined by $\dot{x} = v\cos\theta$, $\dot{y} = v\sin\theta$, $\dot{\theta} = u_1$, and $\dot{v} = u_2$, where x, y
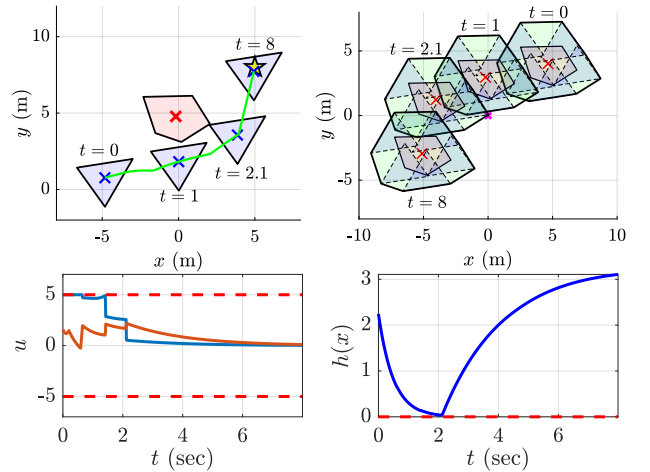

Fig. 2: **Top-Left:** The robot's trajectory (green) moving left-to-right from the start $(t = 0)$ to the goal $(t = 8)$. Snapshots show the robot (blue) avoiding the obstacle (red). **Top-Right:** The corresponding motion of the CO (green) moving right-to-left from $t = 0$ to $t = 8$ in $\mathcal{M}_{\mathcal{D}}$-space (see video https://youtu.be/3Dh0gtDW8bE). **Bottom:** Control input $\boldsymbol{u}(t)$, and CBF $h(\boldsymbol{x})$ profiles. Both control inputs satisfy actuator limits and safety constraints.
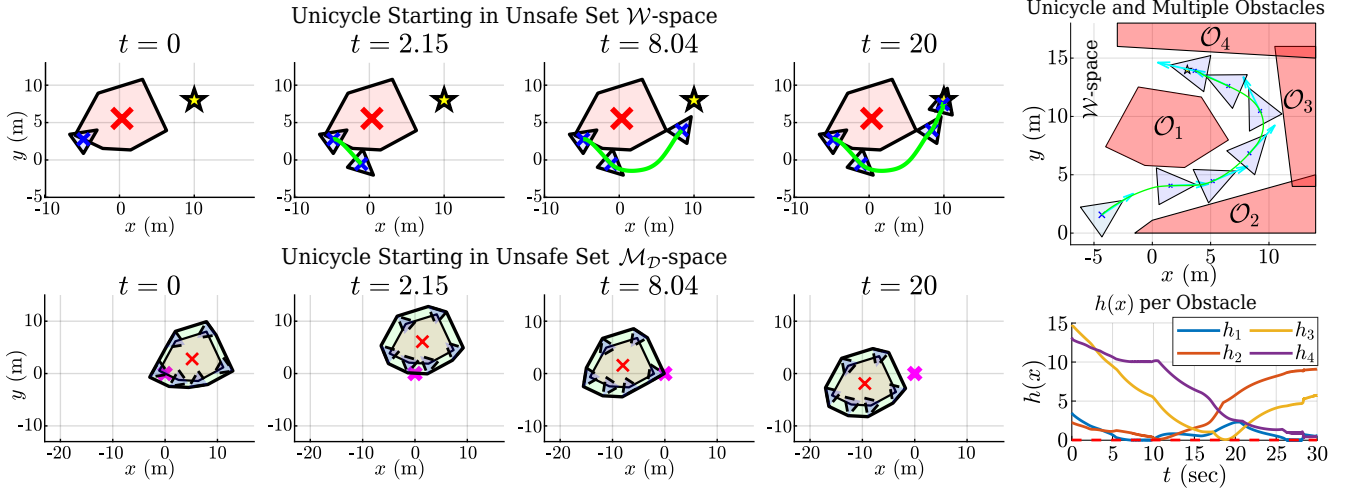
Fig. 3: **Left**: Four columns depict movement over time. At $t = 0$ The robot starts in collision with the obstacle in $\mathcal{W}$-space (top), and the corresponding CO contains the origin in $\mathcal{M}_\mathcal{D}$-space (bottom). As the robot moves out of the obstacle, the CO moves away from the origin ($t = 0$ to $t = 2.15$ top and bottom). The collision is resolved by time $t = 2.15$ seconds. **Right-Top:** Snapshots of robot motion in $\mathcal{W}$-space demonstrate the non-conservativeness of the proposed framework. **Right-Bottom:** The CBFs $h_i(\boldsymbol{x})$ for each obstacle stay above the safety margin over time, thereby guaranteeing safety.

denote the position, θ is the orientation, v is the linear speed, and u_1, u_2 are the two control inputs (turning rate and acceleration). The dynamics can be rewritten in form of (1) with $\boldsymbol{x} = [x, y, \theta, v]^\top, \boldsymbol{u} = [u_1, u_2]^\top, f(\boldsymbol{x}) = [v\cos\theta, v\sin\theta, 0, 0]^\top$ and $g(\boldsymbol{x}) = [\mathbf{0}_{2\times2}; I_2]$.

We highlight that the CO depends on both position and orientation, unlike $\mathcal{W}$-space obstacles, which depend only on position. Thus, high-order CBF [4] or auxiliary-variable CBF [5] is unnecessary to handle the high relative degree constraint. We use two CLFs $V_1(\boldsymbol{x}) = (\theta - \tan^{-1}(\frac{y_d - y}{x_d - x}))^2$, $V_2(\boldsymbol{x}) = (v - v_d)^2$ (similar to [4]) to reach the goal.

As discussed in Sec. IV-D, unlike the purely translational case, we do not have an analytical expression of CO in terms of system state $\boldsymbol{x}$. Instead, at each time step, we compute CO from the robot's current configuration (linear time; see Lem. 2), and approximate the derivative of its bounding constraints numerically. We consider two scenarios: (1) the robot starts in an unsafe set (Fig. 3-Left), and (2) the robot navigates among multiple obstacles (Fig. 3-Right).

1) Unicycle Starting in Unsafe Set (In Collision)

The robot starts in collision with the obstacle; thus, the origin lies within the CO in $\mathcal{M}_\mathcal{D}$-space (see 1st-column of Fig. 3). Note that the CBF's use of SDF between two polygonal sets enables the robot to leave the unsafe set. The parameters used in this case are $\boldsymbol{x}_d = (10, 8), v_d = 2$m/s, $u_{1,\max} = -u_{1,\min} = 5$ rad/s, $u_{2,\max} = -u_{2,\min} = 8$m/s^2, $d_{\text{safe}} = 0.02, p = 8, c = 5, \gamma = 0.8, \epsilon = 10^{-6}$. The robot trajectory is shown in Fig. 3, which demonstrates the proposed framework is able to bring the robot starting in the unsafe set back to the safe set. The computation time of each control loop is 7.85 ms on average.

2) Unicycle Collision Avoidance with Multiple Obstacles

For the multiple obstacles scenario, the safety is guaranteed by ensuring that safety constraint is satisfied w.r.t. each Obstacle $\mathcal{O}_i, i \in \{1, \ldots, 4\}$, as mentioned in Rem. 3, i.e., $h_i(\boldsymbol{x}) = \text{sd}(\mathcal{R}(\boldsymbol{x}), \mathcal{O}_i) - d_{\text{safe}} = \text{sd}(\mathbf{0}_c, \mathcal{O}_i^\mathcal{C}(\boldsymbol{x})) - d_{\text{safe}} \geq 0$.

The associated CBF constraints are enforced in the CLF-CBF-QP as $L_f h_i(\boldsymbol{x}) + L_g h_i(\boldsymbol{x})\boldsymbol{u} + \gamma h_i(\boldsymbol{x}) \geq \epsilon$ The parameters used are the same as in the previous setting, except for $\boldsymbol{x}_d = (3, 13.5), v_d = 1.8$m/s, $p = 10, c = 6, \gamma = 0.3, \epsilon = 0$.

Our CBF framework allows the robot to navigate safely through a simple passage while reaching the goal position (Fig. 3-Right-Top). The time evolution of each CBF $h_i(\boldsymbol{x})$ (Fig. 3-Right-Bottom) shows that all values remain positive throughout the trajectory, indicating that the robot is collision-free w.r.t each obstacle over time. The average computation time per iteration is 19.63 ms.

VI. CONCLUSION AND FUTURE WORK

This paper presents a new CBF framework based on the exact SDF between polygons, obtained via convex optimization in $\mathcal{M}_\mathcal{D}$-space. We have shown that, in the minimum distance case, the proposed CBF is differentiable a.e. and always has a well-defined subdifferential. For penetration depth, we empirically show that the proposed optimization reformulation is compatible with our CBF framework. We validate our approach in three scenarios that highlight non-conservative maneuvering, collision recovery, and applicability to multi-obstacle environment. Future work includes deriving analytical expressions of CO bounding constraints for rotational cases and performing a formal nonsmooth analysis for the penetration depth case.

REFERENCES

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, pp. 3420–3431, IEEE, 2019.

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[3] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, pp. 322–328, 2016.

[4] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Trans. on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[5] S. Liu, W. Xiao, and C. A. Belta, "Auxiliary-variable adaptive control barrier functions for safety critical systems," in *Conference on Decision and Control*, pp. 8602–8607, IEEE, 2023.

[6] S. Liu, W. Xiao, and C. A. Belta, "Auxiliary-variable adaptive control lyapunov barrier functions for spatio-temporally constrained safety-critical applications," in *Conference on Decision and Control*, pp. 8098–8104, IEEE, 2024.

[7] S. Liu, W. Xiao, and C. A. Belta, "Auxiliary-variable adaptive control barrier functions," *arXiv preprint arXiv:2502.15026*, 2025.

[8] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation.," in *Robotics: Science and Systems*, vol. 13, pp. 1–10, Cambridge, MA, USA, 2017.

[9] E. H. Thyri, E. A. Basso, M. Breivik, K. Y. Pettersen, R. Skjetne, and A. M. Lekkas, "Reactive collision avoidance for asvs based on control barrier functions," in *Conference on Control Technology and Applications*, pp. 380–387, IEEE, 2020.

[10] C. Khazoom, D. Gonzalez-Diaz, Y. Ding, and S. Kim, "Humanoid self-collision avoidance using whole-body control with control barrier functions," in *International Conference on Humanoid Robots*, pp. 558–565, IEEE, 2022.

[11] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *International Conference on Robotics and Automation*, pp. 286–292, 2022.

[12] 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*, pp. 3722–3727, IEEE, 2024.

[13] S. Liu, Y. Mao, and C. A. Belta, "Safety-critical planning and control for dynamic obstacle avoidance using control barrier functions," *arXiv preprint arXiv:2403.19122*, 2024.

[14] T. G. Molnar, S. K. Kannan, J. Cunningham, K. Dunlap, K. L. Hobbs, and A. D. Ames, "Collision avoidance and geofencing for fixed-wing aircraft with control barrier functions," *IEEE Trans. on Control Systems Technology*, pp. 1–16, 2025.

[15] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," *IEEE Trans. on Control Systems Technology*, vol. 26, no. 1, pp. 194–206, 2018.

[16] C. K. Verginis and D. V. Dimarogonas, "Closed-form barrier functions for multi-agent ellipsoidal systems with uncertain lagrangian dynamics," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 727–732, 2019.

[17] R. Funada, K. Nishimoto, T. Ibuki, and M. Sampei, "Collision avoidance for ellipsoidal rigid bodies with control barrier functions designed from rotating supporting hyperplanes," *IEEE Trans. on Control Systems Technology*, vol. 33, no. 1, pp. 148–164, 2025.

[18] A. Thirugnanam, J. Zeng, and K. Sreenath, "Duality-based convex optimization for real-time obstacle avoidance between polytopes with control barrier functions," in *American Control Conference*, pp. 2239–2246, 2022.

[19] A. Thirugnanam, J. Zeng, and K. Sreenath, "Nonsmooth control barrier functions for obstacle avoidance between convex regions," *arXiv preprint arXiv:2306.13259*, 2023.

[20] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10954–10961, 2022.

[21] C. Peng, O. Donca, G. Castillo, and A. Hereid, "Safe bipedal path planning via control barrier functions for polynomial shape obstacles estimated using logistic regression," in *International Conference on Robotics and Automation*, pp. 3649–3655, IEEE, 2023.

[22] S. Wei, B. Dai, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "Diffocclusion: Differentiable optimization based control barrier functions for occlusion-free visual servoing," *IEEE Robotics and Automation Letters*, 2024.

[23] S. Wu, Y. Fang, N. Sun, B. Lu, X. Liang, and Y. Zhao, "Optimization-free smooth control barrier function for polygonal collision avoidance," *arXiv preprint arXiv:2502.16293*, 2025.

[24] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[25] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3d game objects," in *Game Developers Conference*, vol. 170, p. 209, 2001.

[26] J.-M. Lien, "Hybrid motion planning using minkowski sums," *Proceedings of Robotics: Science and Systems*, 2008.

[27] S. Ruan, K. L. Poblete, H. Wu, Q. Ma, and G. S. Chirikjian, "Efficient path planning in narrow passages for robots with ellipsoidal components," *IEEE Trans. on Robotics*, vol. 39, no. 1, pp. 110–127, 2023.

[28] M. Lutz and T. Meurer, "Efficient formulation of collision avoidance constraints in optimization based trajectory planning and control," in *Conference on Control Technology and Applications*, pp. 228–233, IEEE, 2021.

[29] L. Montaut, Q. Le Lidec, V. Petrik, J. Sivic, and J. Carpentier, "Collision detection accelerated: An optimization perspective," in *Robotics: Science and Systems*, 2022.

[30] J. Guthrie, "A differentiable signed distance representation for continuous collision avoidance in optimization-based motion planning," in *Conference on Decision and Control*, pp. 7214–7221, IEEE, 2022.

[31] 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.

[32] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control lyapunov functions and hybrid zero dynamics," in *Conference on Decision and Control*, pp. 6837–6842, IEEE, 2012.

[33] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2019.

[34] S. Cameron and R. Culley, "Determining the minimum translational distance between two convex polyhedra," in *International Conference on Robotics and Automation*, vol. 3, pp. 591–596, IEEE, 1986.

[35] C. Ericson, *Real-time collision detection*. Crc Press, 2004.

[36] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[37] M. De Berg, *Computational geometry: algorithms and applications*. Springer Science & Business Media, 2000.

[38] "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[39] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[40] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[41] W. Gao, "Efficient incremental penetration depth estimation between convex geometries," *arXiv preprint arXiv:2304.07357*, 2023.

[42] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*, pp. 136–145, PMLR, 2017.

[43] F. H. Clarke, "Generalized gradients and applications," *Trans. of the American Mathematical Society*, vol. 205, pp. 247–262, 1975.

[44] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings*, vol. 543. Springer, 2009.

[45] S. Sadraddini and R. Tedrake, "Linear encodings for polytope containment problems," 2019.