Safe Navigation and Obstacle Avoidance Using Differentiable Optimization Based Control Barrier Functions

Bolun Dai[1], Rooholla Khorrambakht[1], *Student Member, IEEE*, Prashanth Krishnamurthy[1], *Member, IEEE*,
Vinícius Gonçalves[2], Anthony Tzes[2], *Senior Member, IEEE*, Farshad Khorrami[1], *Senior Member, IEEE*

Abstract—**Control barrier functions (CBFs) have been widely applied to safety-critical robotic applications. However, the construction of control barrier functions for robotic systems remains a challenging task. Recently, collision detection using differentiable optimization has provided a way to compute the minimum uniform scaling factor that results in an intersection between two convex shapes and to also compute the Jacobian of the scaling factor. In this paper, we propose a framework that uses this scaling factor, with an offset, to systematically define a CBF for obstacle avoidance tasks. We provide theoretical analyses of the continuity and continuous differentiability of the proposed CBF. We empirically evaluate the proposed CBF's behavior and show that the resulting optimal control problem is computationally efficient, which makes it applicable for real-time robotic control. We validate our approach, first using a 2D mobile robot example, then on the Franka-Emika Research 3 (FR3) robot manipulator both in simulation and experiment.**

Index Terms—**Robot safety, collision avoidance.**

I. INTRODUCTION

SAFETY is a key consideration when designing control algorithms for robotic applications [1]–[3] considering rapid integration of robotic systems into our daily lives [4]. Model predictive control (MPC) and trajectory optimization (TO) based methods have been widely used for safety-critical robot applications, e.g., obstacle avoidance. However, the computation time of MPC and TO based methods limits their deployment on systems requiring fast response time. Additionally, for MPC-based approaches, safety is only guaranteed within the preview horizon. A short preview horizon might lead to abrupt actions to ensure safety, while large preview horizons increase the computation time. A new control paradigm, CBF-based control [3], has become popular for safe robotic control since it provides a simple and computationally efficient way for safe control synthesis. Another benefit of CBFs is that CBF constraints take safety into consideration even far away from the safe set boundary.

[1]Bolun Dai, Rooholla Khorrambakht, Prashanth Krishnamurthy, and Farshad Khorrami are with Control/Robotics Research Laboratory, Electrical & Computer Engineering Department, Tandon School of Engineering, New York University, Brooklyn, NY 11201 {bd1555, rk4342, prashanth.krishnamurthy, khorrami}@nyu.edu
[2]Vinícius Gonçalves and Anthony Tzes are with Electrical Engineering, New York University Abu Dhabi, Abu Dhabi 129188, United Arab Emirates {vmg6973, anthony.tzes}@nyu.edu
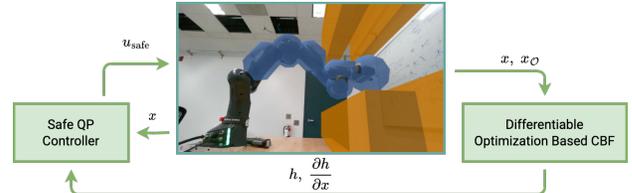
Fig. 1. The structure of safe robotic control using differentiable-optimization-based CBFs.

One of the main hurdles to applying CBF-based methods to robotic systems is the construction of a valid CBF. Although work has been done in synthesizing CBFs for robots, there still lacks a systematic approach for CBF synthesis [5]. Given this difficulty, work has been done in learning CBFs from expert data of safe and unsafe interactions [6]. CBFs for robotics systems can also be learned online using onboard sensor data [7]. Additionally, work has been done in learning CBFs starting from a handcrafted CBF [8]. Although the learning-based methods for CBF synthesis are appealing, acquiring a dataset may be costly in many cases, e.g., self-driving cars. To simplify the CBF construction, work has been done in modeling the interaction between robots and the environment using points and objects [8], spheres [9], and points and higher-order ellipsoids [10]. However, since robotic systems consist of 3D geometrical entities, these approximations either completely ignore or over-estimate the robot geometry, which leads to over-conservative control policies. Recently, work has been done in constructing CBFs for robots represented as polytopic shapes [11] and finding the safe control action using nonlinear model predictive control (NMPC). However, extending their formulation to other shapes, e.g., spheres and cylinders, is not straightforward. Another possible choice for constructing CBFs is to use a proximity measurement like the shortest distance. However, the shortest distance is not well defined when two objects overlap. To address this issue, signed distance functions (SDFs) are used instead in [12] to construct CBFs. However, algorithms used to compute the SDF, e.g., the Gilbert–Johnson–Keerthi (GJK) distance algorithm [13] and the expanding polytope algorithm (EPA) [14], are not differentiable [15] and SDFs are also nonsmooth [12]. Thus, an approximated version of the partial derivative of the SDF is used along with a looser constraint to generate safe control actions, which results in a conservative controller.

We construct our CBF in this paper to overcome the above

limitations based on differentiable optimization methods. Unlike traditional optimization solvers, differentiable optimization solvers, in addition to the optimal solution, also provide the partial derivatives of the optimal solution with respect to the problem parameters [16]. Given their differentiability, differentiable optimization solvers have gained popularity in the machine learning community by integrating them within a deep learning pipeline [17]. Recently, differentiable optimization solvers have also been used for collision detection between two convex shapes [15] by finding the minimum scaling factor for the two objects that leads to their collision.

This work builds upon the idea of using a differentiable collision detector and proposes a method to define CBFs for robotic obstacle avoidance tasks systematically. The proposed approach is efficient to compute, directly differentiable, handles a wide range of geometries, and is well-defined even in collision. The main contribution of this paper is twofold: (1) proposing and theoretically analyzing a differentiable optimization based approach to synthesize CBFs for robotic obstacle avoidance tasks that consider both the robot and scene geometry; (2) performing simulations and experiments (on FR3) to show the efficacy of our approach. This paper is structured as follows. In Section II, we briefly review CBFs and mathematical foundations of differentiable optimization solvers. In Section III, we formulate the safe robotic control problem. In Section IV, we present our approach for constructing CBFs using differentiable optimization solvers. In Section V, we show efficacy of our approach using a 2D mobile robot and on the seven degrees-of-freedom (DOF) FR3 robotic arm in both simulation and real world. Section VI concludes the paper with a discussion on future directions.

II. PRELIMINARIES

This section presents a brief introduction to CBF and differentiable optimization.

A. Control Barrier Functions

Consider a control affine system

$$\dot{x} = F(x) + G(x)u \qquad (1)$$

where the state is $x \in \mathbb{R}^n$ and the control input is $u \in \mathbb{R}^m$, with $\mathcal{U}$ being the admissible set of controls. The locally Lipschitz continuous functions $F : \mathbb{R}^n \to \mathbb{R}^n$ and $G : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ represent the drift and the control influence matrix, respectively. Additionally, we assume access to a controller $u = \pi(x)$, with $\pi : \mathbb{R}^n \to \mathbb{R}^m$ being locally Lipschitz continuous. We say the controller π can keep the system in (1) safe with respect to a set $\mathcal{C} \subset \mathbb{R}^n$ if the controller π renders the set $\mathcal{C}$ forward control invariant. In other words, the controller π keeps (1) safe with respect to $\mathcal{C}$, if for any initial state $x_0 \in \mathcal{C}$, the solution to (1), defined as $x(t)$, remains within the safe set $\mathcal{C} \; \forall t \in \mathbf{I}(x_0)$. The time interval of existence $\mathbf{I}(x_0) = [t_0, t_{\max})$ is where $x(t)$ is a unique solution to (1); the system defined in (1) is considered forward complete when $t_{\max} = \infty$. Let the set $\mathcal{C}$, with $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$, be the 0-superlevel set of a

continuously differentiable function $\mathbf{h} : \mathcal{D} \to \mathbb{R}$ that has the property $\partial \mathbf{h}/\partial x \neq 0$ for all $x \in \partial \mathcal{C}$. Then, for (1), if

$$\sup_{u \in \mathcal{U}} \left[\frac{\partial \mathbf{h}(x)}{\partial x} \Big(F(x) + G(x)u \Big) \right] \geq -\Lambda(\mathbf{h}(x)) \qquad (2)$$

holds for all $x \in \mathcal{D}$, with $\Lambda : \mathbb{R} \to \mathbb{R}$ being an extended class $\mathcal{K}_\infty$ function[1], we say that $\mathbf{h}$ is a CBF on $\mathcal{C}$.

B. Differentiable Optimization

Consider a convex optimization problem in the form of

$$\min_{y \in \mathbb{R}^n} \quad f(y \mid \psi) \qquad (3)$$
$$\text{subject to} \quad \ell(y \mid \psi) = 0$$
$$h(y \mid \psi) \leq 0$$

with $f : \mathbb{R}^n \to \mathbb{R}$ being the convex objective function, $\ell : \mathbb{R}^n \to \mathbb{R}^{n_e}$ the affine equality constraints, $h : \mathbb{R}^n \to \mathbb{R}^{n_i}$ the convex inequality constraints, and $\psi \in \mathbb{R}^{n_\psi}$ the problem configuration parameters. n_e, n_i, and n_ψ represent the number of equality constraints, inequality constraints, and problem configuration parameters, respectively. The Karush-Kuhn-Tucker (KKT) conditions for stationarity, primal feasibility, and complementary slackness are

$$0 = \partial_y \Big(f(y^\star) + (\lambda^\star)^\top \ell(y^\star) + (\nu^\star)^\top h(y^\star) \Big) \qquad (4a)$$
$$0 = \ell(y^\star) \in \mathbb{R}^{n_e} \qquad (4b)$$
$$0 = \text{diag}(\nu^\star) h(y^\star) \in \mathbb{R}^{n_i} \qquad (4c)$$

with $\lambda \in \mathbb{R}^{n_e}$ and $\nu \in \mathbb{R}^{n_i}$ being the dual variables, the superscript $\star$ representing the optimal values, and the ψ's omitted for brevity. Define $z^\star(\psi) = \begin{bmatrix} y^\star(\psi) & \lambda^\star(\psi) & \nu^\star(\psi) \end{bmatrix}^\top \in \mathbb{R}^{n_z}$, with $n_z = n + n_e + n_i$. Eq. (4) can be written as

$$g(z^\star(\psi), \psi) = 0 \qquad (5)$$

with $g : \mathbb{R}^{n_z} \times \mathbb{R}^{n_\psi} \to \mathbb{R}^{n_z}$. To compute the partial derivative of $z^\star$ w.r.t ψ, we utilize the implicit function theorem.

Theorem 1 (Implicit Function Theorem [18]). *Let $g : \mathbb{R}^{n_z + n_\psi} \to \mathbb{R}^{n_\psi}$ be a continuously differentiable function, $z_0^\star \in \mathbb{R}^{n_z}$, and $\psi_0 \in \mathbb{R}^{n_\psi}$. Let $(z_0^\star, \psi_0)$ satisfy $g(z_0^\star, \psi_0) = 0$, and assume $\partial_{z^\star} g(z^\star, \psi_0)|_{z^\star = z_0^\star}$ is invertible. Then there exist open sets $S_{z^\star} \subset \mathbb{R}^{n_z}$ and $S_\psi \subset \mathbb{R}^{n_\psi}$ containing $z_0^\star$ and ψ_0, respectively, and a unique continuously differentiable function $z^\star : \mathbb{R}^{n_\psi} \to \mathbb{R}^{n_z}$ such that $z^\star(\psi_0) = z_0^\star$, $g(z^\star(\psi_0), \psi_0) = 0$.*

Assume the implicit function Theorem holds for g, $z_0^\star$, and ψ_0. Taking the derivative on both sides of Eq. (5) yields

$$\frac{dg(z^\star, \psi)}{d\psi} \bigg|_{\substack{z^\star = z_0^\star \\ \psi = \psi_0}} = 0. \qquad (6)$$

Then, using the chain rule, we have

$$\partial_{z^\star} g(z^\star, \psi_0) \bigg|_{z^\star = z_0^\star} \partial_\psi z^\star(\psi) \bigg|_{\psi = \psi_0} + \partial_\psi g(z_0^\star, \psi) \bigg|_{\psi = \psi_0} = 0. \qquad (7)$$

---

[1]Extended class $\mathcal{K}_\infty$ functions are strictly increasing with $\Lambda(0) = 0$.

Finally, applying the implicit function Theorem yields

$$\partial_\psi z^\star(\psi)\Big|_{\psi=\psi_0} = -\partial_{z^\star}^{-1} g(z^\star, \psi_0)\Big|_{z^\star=z_0^\star} \partial_\psi g(z_0^\star, \psi)\Big|_{\psi=\psi_0}. \tag{8}$$

This provides a way to efficiently and exactly compute the gradient of the optimal solution of a convex optimization problem with respect to its problem configuration.

C. Berge's Maximum Theorem

A key theoretical result we use later in the paper is Berge's Maximum Theorem.

Theorem 2 (Berge's Maximum Theorem [19]). *Let X and Ψ be topological spaces, and $\mathcal{J} : X \times \Psi \to \mathbb{R}$ be a continuous function on $X \times \Psi$ and $\Gamma : \Psi \rightrightarrows X$ be a compact-valued correspondence[2] such that $\Gamma(\psi) \neq \emptyset$ for all $\psi \in \Psi$. Define the value function $\mathcal{J}^\star : \Psi \to \mathbb{R}$ as*

$$\mathcal{J}^\star(\psi) = \sup\{\mathcal{J}(x, \psi) \mid x \in \Gamma(\psi)\} \tag{9}$$

and the solution function $\Gamma^\star : \Psi \to X$ as

$$\Gamma^\star(\psi) = \{x \mid x \in \Gamma(\psi), \mathcal{J}(x, \psi) = \mathcal{J}^\star(\psi)\}. \tag{10}$$

If Γ is an upper and lower hemicontinuous (UHC/LHC) correspondence [19] at ψ, then $\mathcal{J}^\star$ is continuous and $\Gamma^\star$ is UHC with nonempty and compact values.

III. PROBLEM FORMULATION

We consider the problem of CBF-based obstacle avoidance for robotic systems in the form of (1). For each obstacle, we can represent the obstacle avoidance task as $c_i(x) \geq 0$, where $c_i : \mathbb{R}^n \to \mathbb{R}$ represents the i-th obstacle avoidance constraint. For each constraint function c_i, we can define its 0-superlevel set as $\mathcal{C}_i$. The safe set $\mathcal{C}$ for the robot is then the intersection of $\mathcal{C}_i$'s, i.e.,

$$\mathcal{C} = \bigcap_{i=0}^{n_i} \mathcal{C}_i = \bigcap_{i=0}^{n_i} \{x \mid x \in \mathbb{R}^n, c_i(x) \geq 0\}. \tag{11}$$

This paper aims to find a systematic way of constructing a CBF such that the robot can stay within $\mathcal{C}$.

IV. DIFFERENTIABLE OPTIMIZATION BASED CBFS

In this section, we present our proposed method of using differentiable optimization to compute CBFs for safe robotic control. First, we motivate our work by showing the limitations of SDF-based CBFs. Second, the CBF formulation is presented. Then, we show how to construct the CBF for robotic applications. Finally, we show how to construct the optimization problem for finding the safe control action.

[2]Correspondences are set-valued functions.

A. Motivation

For maintaining safety in obstacle avoidance tasks, the most straightforward approach is to use a proximity measurement, i.e., SDF, for constructing the CBF, which for two convex objects A and B is defined as $\mathrm{SDF}(A, B) = \mathrm{distance}(A, B) - \mathrm{penetration}(A, B)$. Currently, the most efficient method to compute the SDF is using GJK to compute the distance and EPA to compute the penetration. Using GJK and EPA, we get the witness points $\hat{p}_A, \hat{p}_B \in \mathbb{R}^4$ and the vector of minimal translation $\hat{n} \in \mathbb{R}^4$ in homogeneous form. Then, the CBF can be written as [20]

$$\mathbf{h}_{\mathrm{SDF}}(x_A) = \hat{n}(x_A)^\top (F_A^W(x_A)\hat{p}_A(x_A) - F_B^W \hat{p}_B(x_A)) \tag{12}$$

with $F_A^W, F_B^W \in \mathbb{R}^{4\times4}$ denoting the poses of A and B and $x_A \in \mathbb{R}^n$ representing the state of A. In (12), we assume B is static. However, due to their logical control flows and pivoting, the output of GJK and EPA, i.e., $(\hat{p}_A, \hat{p}_B, \hat{n})$, is inherently non-differentiable [15]. The work in [12] bypasses this issue by only taking the partial derivative with respect to F_A^W and treating the remainder terms as disturbance:

$$\frac{\partial \mathbf{h}_{\mathrm{SDF}}}{\partial x_A} = \hat{n}(x_A)^\top \underbrace{\frac{\partial F_A^W}{\partial x_A} \hat{p}_A(x_A)}_{J_A(x_A)} + \delta(x_A) \tag{13}$$

with $\delta : \mathbb{R}^n \to \mathbb{R}^n$ representing the remainder terms associated with the derivatives of $\hat{n}$, $\hat{p}_A$, and $\hat{p}_B$. Then, the safety constraint is written as

$$\hat{n}(x_A)^\top J_A(x_A)\dot{x}_A \geq -\gamma\mathbf{h}_{\mathrm{SDF}} + 2J_{\max}\dot{x}_{A,\max} \tag{14}$$

with $\|\delta(x)\|_\infty \leq 2\max_{x_A}\|J_A(x_A)\| = 2J_{\max}$ and $\|\dot{x}_A\|_\infty = \dot{x}_{A,\max}$. For derivation details, please refer to [12]. This is not a tight bound and therefore generates a conservative controller that only recovers a portion of the safe set. In the remainder of this section, we formulate a CBF that is directly differentiable and can recover the entire safe set.

B. CBF Formulation

Inspired by [15], we construct a CBF using the minimum scaling of two convex objects under which they collide. The scaling $\alpha \in \mathbb{R}_+$ can be computed using a conic program

$$\min_{p,\alpha} \quad \alpha \tag{15}$$
$$\text{subject to} \quad p \in \mathcal{P}_A(\alpha)$$
$$p \in \mathcal{P}_B(\alpha)$$
$$\alpha > 0.$$

where $p \in \mathbb{R}^3$ represents a point and $P_A(\alpha)$ a set that contains the interior and surface of A after scaling it uniformly using a scaling factor α. $P_B(\alpha)$ is defined in the same manner. Denote the optimal value for α as $\alpha^\star$. If $\alpha^\star > 1$, then the two convex objects are not in collision. The two convex objects collide if $\alpha^\star \leq 1$. The optimal p for (15), denoted as $p^\star$, represents the point of intersection after scaling the two convex objects. A visual illustration of the solution to (15) can be found in Fig. 2. Then, we formulate the CBF as

$$\mathbf{h}(x) = \alpha^\star(x) - \beta. \tag{16}$$
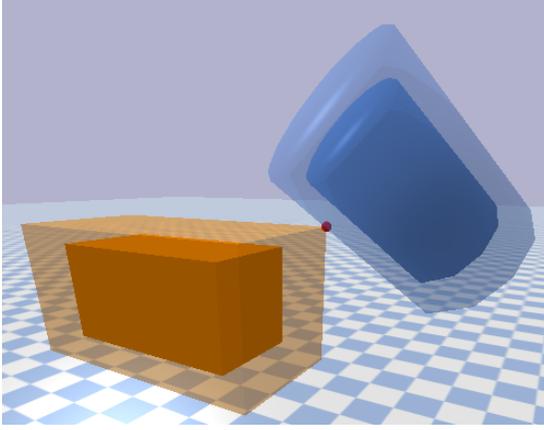
Fig. 2. This figure illustrates the outcome of the differentiable conic program in (15). The first convex object is the polygon and the second is the cylinder. The transparent regions surrounding the solid objects represent the scaled version of the objects after scaling them with a ratio of α. The red dot represents the point of intersection p of the two scaled objects.

with $\beta \in \mathbb{R}$ and $\beta \geq 1$. Using Theorem 1 and Eq. (6)-(8), we can obtain the Jacobian of $\alpha^\star$ as

$$\frac{\partial \alpha^\star}{\partial(r_1, q_1, r_2, q_1)} = \frac{\partial \mathbf{h}}{\partial(r_1, q_1, r_2, q_1)} \in \mathbb{R}^{1 \times 14} \qquad (17)$$

where $r_1, r_2 \in \mathbb{R}^3$ represents the positions of A and B, $q_1, q_2 \in \mathbb{R}^4$ represents the orientations in quaternions, and $\psi = (r_1, q_1, r_2, q_2) \in \Psi \subset \mathrm{SE}(3) \times \mathrm{SE}(3)$ represents joint pose of the two objects.

To show (16) is a valid CBF, we need to show continuous differentiability of $\alpha^\star$. In this paper, we prove continuity of $\alpha^\star$ in the general case and then prove continuous differentiability for the case of strongly convex scaling function when the gradients and Hessians exist and are continuous. We conjecture that continuous differentiability can be generalized to one object being strongly convex and the second being only convex although a formal proof is still elusive.

Theorem 3. *The optimal value of* (15)*, denoted as* $\alpha^\star$*, is a continuous function with respect to the positions and orientations of A and B.*

To prove Theorem 3, we need Lemma 1 with the constraint correspondence $\Gamma(\psi)$ defined as

$$\Gamma(\psi) = \{(p, \alpha) \mid p \in \mathcal{P}_{AB}(\alpha \mid \psi), \alpha \in S\} \qquad (18)$$

with $S \subset \mathbb{R}$ being a compact set and $\mathcal{P}_{AB}(\alpha \mid \psi) = \mathcal{P}_A(\alpha \mid \psi) \cap \mathcal{P}_B(\alpha \mid \psi)$. The term $P_A(\alpha \mid \psi)$ denotes that P_A depends on both α and ψ, where ψ is seen as the configuration parameter. $\mathcal{P}_{AB}(\alpha \mid \psi)$ and $\mathcal{P}_B(\alpha \mid \psi)$ are defined in a similar manner as $\mathcal{P}_A(\alpha \mid \psi)$. Also we define $\Gamma_i(\psi) = \{(p, \alpha) \mid p \in \mathcal{P}_i(\alpha \mid \psi), \alpha \in S\}$ with $i \in \{A, B\}$. We can then show that $\Gamma(\psi)$ is a compact-valued correspondence on $\mathbb{R}^4$.

Lemma 1. *The constraint correspondence $\Gamma(\psi)$ defined in* (18) *is a compact-valued correspondence.*

Proof. First, we show that $\Gamma_A(\psi)$ is a compact-valued correspondence. Define $K_A(\psi) = \mathcal{P}_A(1 \mid \psi) \subset \mathbb{R}^3$, and the

function $\phi : K_A(\psi) \times S \to \mathbb{R}^4$ as $\phi(p, \alpha) = (\alpha(p - r_1) + r_1, \alpha)$, which is a continuous function on $K_A(\psi) \times S$. Because both $K_A(\psi)$ and S are compact sets, we have $K_A(\psi) \times S$ as a compact set on $\mathbb{R}^4$. Since the image of ψ under Γ_A is the same as the image of $K_A(\psi) \times S$ under ϕ, using the property that continuous functions map compact sets to compact sets, and $K_A(\psi)$ is a compact set for any value of ψ, we have $\Gamma_A(\psi)$ being a compact-valued correspondence. Using the same logic, for all ψ, $\Gamma_B(\psi)$ is also a compact-valued correspondence. Since $\Gamma(\psi) = \Gamma_A(\psi) \cap \Gamma_B(\psi)$, and intersections of compact sets are compact sets, we can conclude that $\Gamma(\psi)$ is a compact-valued correspondence. $\square$

We can directly apply Theorem 2 to prove Theorem 3 if we show that $\Gamma(\psi)$ is both UHC and LHC.

Proof. **(Theorem 3)** To show UHC, we consider an arbitrary open set V such that $\Gamma(\psi) \subset V$, i.e.,

$$V = \left\{(p, \alpha) \mid \mathcal{P}_{AB}(\alpha \mid \psi) \subset V_p(\alpha), S \subset V_\alpha\right\} \qquad (19)$$

with V_p being an open set on $\mathbb{R}^3$, $p \in \mathcal{P}_{AB}(\alpha \mid \psi)$, V_α being an open set on $\mathbb{R}$, and $\alpha \in S$. In our case, UHC is equivalent to: for any ψ and α, if the intersection of the two convex shapes $\mathcal{P}_A$ and $\mathcal{P}_B$ is contained within an open set $V_p(\alpha)$, then after applying an infinitesimally small translation and rotation, the intersection is still contained within $V_p(\alpha)$. Since $V_p(\alpha)$ is an open set, $\forall p \in \mathcal{P}_{AB}(\alpha \mid \psi)$ there exists an open neighborhood around p that is a subset of $V_p(\alpha)$. Additionally, the compactness of $\mathcal{P}_{AB}(\alpha \mid \psi)$ implies that a finite subset of the union of those neighborhoods covers $\mathcal{P}_{AB}(\alpha \mid \psi)$. After applying an infinitesimally small transformation to p, it should stay in that finite cover because the change in distance between the origin and the point p is Lipschitz continuous [21]. For LHC, we consider an arbitrary open set $\bar{V}$ such that $\Gamma(\psi) \cap \bar{V} \neq \emptyset$, i.e., with $p \in \mathcal{P}_{AB}(\alpha \mid \psi)$, $\bar{V}_p(\alpha)$ being an open set on $\mathbb{R}^3$, $\bar{V}_\alpha$ an open set on $\mathbb{R}$, and $\alpha \in S$, we have

$$\bar{V} = \left\{(p, \alpha) \mid \bar{V}_\alpha \cap S \neq \emptyset, \mathcal{P}_{AB}(\alpha \mid \psi) \cap \bar{V}_p(\alpha) \neq \emptyset\right\}. \qquad (20)$$

In our case, LHC is equivalent to: for $\alpha \in \bar{V}_\alpha$, if $\mathcal{P}_{AB}(\alpha \mid \psi) \cap \bar{V}_p(\alpha) \neq \emptyset$, then after applying an infinitesimally small translation and rotation, the intersection set is not empty. Since there exists $p \in \mathcal{P}_{AB}(\alpha \mid \psi)$ with an open neighborhood that belongs to $\bar{V}_p(\alpha)$, following the logic for showing UHC, Γ is also LHC. Thus, since $\Gamma(\psi)$ is a compact-valued correspondence, using Theorem 2, we know $J^\star = \alpha^\star$ is a continuous function with respect to ψ. $\square$

Next, under sufficient conditions, we show that $\alpha^\star$ is continuously differentiable with respect to ψ.

Definition 1 (Scaling Function)**.** *We say that $\mathcal{F}_A : \mathbb{R}^3 \to \mathbb{R}$, is a scaling function for an object $A \subset \mathbb{R}^3$, if it is a continuously differentiable convex function, $\mathcal{F}_A(p) \geq 0$, $A = \{p \mid \mathcal{F}_A(p) \leq 1\}$, $\partial A = \{p \mid \mathcal{F}_A(p) = 1\}$, and $\min_p \mathcal{F}_A(p) = 0$.*

For example, the scaling function for an ellipsoid A has form $\mathcal{F}_A = (p - r)^\top \mathbf{P}(p - r)$ where $\mathbf{P} \in \mathbb{R}^3$ is a diagonal matrix with inverses of squared semi-axes lengths on its diagonal. This particular example is used in [15] to implement the

scaling constraint for the ellipsoid, and similar functions can be derived for other shapes discussed in [15].

Lemma 2. *For two objects $A, B \subset \mathbb{R}^3$, denote their scaling functions as $\mathcal{F}_A$ and $\mathcal{F}_B$, respectively. If $\mathcal{F}_A$ and $\mathcal{F}_B$ are differentiable functions, and no point p exists such that $\partial \mathcal{F}_A / \partial p = \partial \mathcal{F}_B / \partial p = \mathbf{0}$, then for the optimal solution $(p^\star, \alpha^\star)$ of (15), we have $\mathcal{F}_A(p^\star) = \mathcal{F}_B(p^\star) = \alpha^\star$ and $\nu_A^\star, \nu_B^\star \neq 0$.*

Proof. First, we rewrite the problem in (15) as

$$\min_{p, \alpha} \ \alpha \tag{21}$$
$$\text{subject to} \ \ \mathcal{F}_A(p) \leq \alpha$$
$$\mathcal{F}_B(p) \leq \alpha$$

We can write the Lagrangian for (21) as

$$\mathcal{L}(\alpha, p, \nu_A, \nu_B) = \alpha + \nu_A(\mathcal{F}_A(p) - \alpha) + \nu_B(\mathcal{F}_B(p) - \alpha) \tag{22}$$

From the stationarity condition of the KKT conditions we have

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 1 - \nu_A^\star - \nu_B^\star = 0 \tag{23a}$$

$$\frac{\partial \mathcal{L}}{\partial p} = \nu_A^\star \frac{\partial \mathcal{F}_A}{\partial p}(p^\star) + \nu_B^\star \frac{\partial \mathcal{F}_B}{\partial p}(p^\star) = 0 \tag{23b}$$

From the complementary slackness condition we have

$$\nu_A^\star(\mathcal{F}_A(p^\star) - \alpha^\star) = 0 \tag{24a}$$

$$\nu_B^\star(\mathcal{F}_B(p^\star) - \alpha^\star) = 0. \tag{24b}$$

It can be shown that neither $\nu_A^\star$ nor $\nu_B^\star$ can be 0. To show this, without loss of generality, assume $\nu_A^\star = 0$. From (23a), we have $\nu_B^\star = 1$ and from (23b) we have $\nabla_p \mathcal{F}_B(p^\star) = 0$. From definition 1, we know that $\mathcal{F}_B$ is a convex differentiable function, therefore, $p^\star$ is the global minimum of $\mathcal{F}_B$. Hence, also from definition 1, $\mathcal{F}_B(p^\star) = 0$. Then, from (24b), $\alpha^\star = \mathcal{F}_B(p^\star) = 0$. Using the constraint $\mathcal{F}_A(p) \leq \alpha$ from (21) and the condition $\mathcal{F}_A(p) \geq 0$ from definition (1), we have $\mathcal{F}_A(p^\star) = 0$, which means $p^\star$ is also a global minimizer for $\mathcal{F}_A$, i.e., $\partial \mathcal{F}_A(p^\star) / \partial p = \mathbf{0}$. Since, by the assumption of Lemma 2, no point p exists such that $\partial \mathcal{F}_A / \partial p = \partial \mathcal{F}_B / \partial p = \mathbf{0}$, thus, $\nu_A^\star \neq 0$. Given the symmetry between $\nu_A^\star$ and $\nu_B^\star$, $\nu_B^\star \neq 0$. Therefore, from (24a) and (24b), we have $\mathcal{F}_A(p^\star) = \mathcal{F}_B(p^\star) = \alpha$. $\square$

Theorem 4. *Consider two parameterized scaling functions $\mathcal{G}_A(p, \psi) = \mathcal{F}_{A(\psi)}(p)$ and $\mathcal{G}_B(p, \psi) = \mathcal{F}_{B(\psi)}(p)$ such that, for any $\psi \in \Psi$, these are scaling functions for varying sets $A(\psi), B(\psi) \subset \mathbb{R}^3$. Furthermore, for all p and ψ, assume that the conditions of Lemma 2 holds, $\mathcal{G}_A$ and $\mathcal{G}_B$ are strongly convex on p, the gradients $\partial \mathcal{G}_A / \partial \psi$ and $\partial \mathcal{G}_B / \partial \psi$ and the Hessians $\partial^2 \mathcal{G}_A / \partial p^2$, $\partial^2 \mathcal{G}_A / \partial p \partial \psi$, $\partial^2 \mathcal{G}_B / \partial p^2$, and $\partial^2 \mathcal{G}_B / \partial p \partial \psi$ are continuous in ψ, then, $\alpha^\star(\psi)$ is continuously differentiable w.r.t. ψ.*

Proof. From (23a), we have $\partial \nu_A^\star / \partial \psi + \partial \nu_B^\star / \partial \psi = 0$. Then, using Lemma 2, and taking the partial derivative with respect to ψ on both sides of (23a) and (23b), we obtain

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{c} \\ \mathbf{c}^\top & \mathbf{0} \end{bmatrix}}_{\mathbf{N}} \left[\left(\frac{\partial p^\star}{\partial \psi} \right)^\top \ \left(\frac{\partial \nu_A^\star}{\partial \psi} \right)^\top \right]^\top = \begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} \tag{25}$$

with

$$\mathbf{M} = \frac{\partial^2 \mathcal{G}_A}{\partial p^2}(p^\star, \psi) \nu_A^\star + \frac{\partial^2 \mathcal{G}_B}{\partial p^2}(p^\star, \psi) \nu_B^\star \tag{26a}$$

$$\mathbf{c} = \frac{\partial \mathcal{G}_A}{\partial p}(p^\star, \psi) - \frac{\partial \mathcal{G}_B}{\partial p}(p^\star, \psi) \tag{26b}$$

$$\Omega_1 = -\frac{\partial^2 \mathcal{G}_A}{\partial p \partial \psi}(p^\star, \psi) \nu_A^\star - \frac{\partial^2 \mathcal{G}_B}{\partial p \partial \psi}(p^\star, \psi) \nu_B^\star \tag{26c}$$

$$\Omega_2 = \frac{\partial \mathcal{G}_B}{\partial \psi}(p^\star, \psi) - \frac{\partial \mathcal{G}_A}{\partial \psi}(p^\star, \psi). \tag{26d}$$

Since $\mathcal{G}_A$ and $\mathcal{G}_B$ are strongly convex and $\nu_A^\star$ and $\nu_B^\star$ are dual feasible and thus strictly positive from Lemma 2, $\mathbf{M} \succ 0$. Note that $\mathbf{c} = \mathbf{0}$ would imply from (23a) and (23b) that $\partial \mathcal{G}_A(p^\star, \psi) / \partial p = \partial \mathcal{G}_B(p^\star, \psi) / \partial p = 0$, violating the condition of Lemma 2. Hence, $\mathbf{c} \neq \mathbf{0}$. From Schur's formula, we know that

$$\det(\mathbf{N}) = \det(\mathbf{M}) \det(-\mathbf{c}^\top \mathbf{M}^{-1} \mathbf{c}). \tag{27}$$

Since $\mathbf{M} \succ 0$ and $\mathbf{c} \neq \mathbf{0}$, we can conclude that $\det(\mathbf{N}) \neq 0$, i.e., $\mathbf{N}$ is invertible. Since, $\mathbf{M}, \mathbf{c}, \Omega_1$, and Ω_2 are continuous in ψ, $\mathbf{N}^{-1}$ is continuous on ψ, thus, $\partial p^\star / \partial \psi$ is also continuous in ψ. Then, from Lemma 2, we have

$$\alpha^\star = \mathcal{G}_A(p^\star, \psi). \tag{28}$$

Taking the derivative w.r.t. ψ on both sides of the equation yields

$$\frac{\partial \alpha^\star}{\partial \psi}(\psi) = \frac{\partial \mathcal{G}_A}{\partial p}(p^\star, \psi) \frac{\partial p^\star}{\partial \psi}(\psi) + \frac{\partial \mathcal{G}_A}{\partial \psi}(p^\star, \psi) \tag{29}$$

which, following the above derivation, is continuous. $\square$

Remark 1. *For ellipsoids/spheres, we can write a scaling function (as noted after Definition 1) that satisfies Theorem 4.*

From empirical studies, we believe that continuous differentiability holds under more general conditions motivating Conjecture 1.

Conjecture 1. *Assume we have two convex objects A and B. If A is a strongly convex object (e.g., ellipsoid, sphere) and B is a convex object that can either have a non-smooth surface (e.g., cylinder, polygon, cone) or a smooth surface, then the CBF defined in (16) is continuously differentiable.*

C. CBF Constraint

We consider the CBF formulation in Section IV-B using velocity control. For robotic systems with dynamics (1), it was shown in [22] that a derivative controller with gravity compensation could realize input-to-state safety (ISSf) when tracking a planned safe velocity command. Similar to [22], by tuning the value of β in (16), we can achieve safety with

respect to the actual safe set, even when the velocity command is not perfectly tracked. For velocity control, the system dynamics have the form of $\dot{x} = G(x)u$, with n_j denoting the number of generalized coordinates and $n = m = n_j$. The above equation is the same as (1) when $F(x) = 0$. For the CBF constraint, we need to compute the value of $\dot{\mathbf{h}}(x)$. For a single rigid body, the chain rule yields

$$\dot{\mathbf{h}}(x) = \frac{\partial \mathbf{h}}{\partial \mu} \frac{\partial \mu}{\partial x} \dot{x}. \tag{30}$$

where $\mu = [r^\top, q^\top]^\top \in \mathbb{R}^7$ represents the rigid body's position and orientation. The first term is obtained from (17). The second term can be separated into two parts: the positional part and the orientational part. For the positional part, $\partial r / \partial x = J_v(x)$, where $J_v : \mathbb{R}^{n_j} \to \mathbb{R}^{3 \times n_j}$ is the positional Jacobian matrix. The orientational Jacobian J_ω has the relationship $\omega = J_\omega(x)\dot{x}$, where $\omega \in \mathbb{R}^3$ represents the frame angular velocity and $J_\omega : \mathbb{R}^{n_j} \to \mathbb{R}^{3 \times n_j}$. Define the (vectorized) quaternion as $q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^\top \in \mathbb{R}^4$, which satisfies

$$\dot{q} = \frac{1}{2}\mathbf{Q}\omega = \frac{1}{2}\mathbf{Q}J_\omega(x)\dot{x} \tag{31}$$

$$\mathbf{Q} = \begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix} \in \mathbb{R}^{4 \times 3}. \tag{32}$$

Then, we have

$$\frac{\partial \mu}{\partial x} = \begin{bmatrix} \dfrac{\partial r}{\partial x} \\ \dfrac{\partial q}{\partial x} \end{bmatrix} = \begin{bmatrix} J_v(x) \\ \dfrac{1}{2}\mathbf{Q}J_\omega(x) \end{bmatrix} \in \mathbb{R}^{7 \times n_j}. \tag{33}$$

Finally, we have the CBF constraint as

$$\frac{\partial \mathbf{h}}{\partial x}G(x)u = \frac{\partial \mathbf{h}}{\partial \mu}\frac{\partial \mu}{\partial x}G(x)u \geq -\gamma \mathbf{h}(x). \tag{34}$$

with $\gamma \in \mathbb{R}_+$. In the CBF constraint, we use $\Lambda(a) = \gamma a$.

Remark 2. *Since the Jacobians are continuous functions, the partial derivative of α w.r.t. the general coordinates of the robot is a continuous function on $\mathcal{D}$ if the partial derivative of α w.r.t. (r, q) is a continuous function on $\mathcal{D}$.*

In the remainder of this section, we show conditions that guarantee $\partial \mathbf{h}/\partial x \neq 0$ for $x \in \partial \mathcal{C}$, which guarantees the validity of the CBF.

Lemma 3. *For $x \in \partial \mathcal{C}$, if the Jacobian matrix $J(x) = \begin{bmatrix} J_v^\top(x) & J_\omega^\top(x) \end{bmatrix}^\top$ has full row rank, then $\partial \mathbf{h}/\partial x \neq 0$.*

Proof. We can write $\partial \mathbf{h}/\partial x$ as

$$\frac{\partial \mathbf{h}}{\partial x} = \frac{\partial \mathbf{h}}{\partial \mu}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathbf{Q} \end{bmatrix}J(x) = \frac{\partial \mathbf{h}}{\partial \mu}\mathbf{A_Q}J(x). \tag{35}$$

It can be seen that if $(\partial \mathbf{h}/\partial \mu)\mathbf{A_Q} = 0$, then no infinitesimally small change in pose (including translations and rotations) exists that changes $\mathbf{h}$. It is evident that this is not the case since the distance between two convex objects can always be modified by some combination of translations and rotations.

Thus, we know that $(\partial \mathbf{h}/\partial \mu)\mathbf{A_Q} \neq 0$. Since $J(x)$ has full row rank, we have $\partial \mathbf{h}/\partial x \neq \mathbf{0}$. $\qquad\square$

Remark 3. *In addition to Lemma 3, for a fully actuated system with no input constraint, there always exists a u that satisfies the CBF constraint when $J(x)$ is non-singular.*

D. Safe Controller

For robotic applications, robots and obstacles are often rigid multibody systems. We represent each rigid body using convex primitive shapes. Then, using the proposed CBF construction method, we write a CBF for each robot-obstacle-rigid-body pair within the robot's workspace. Let the robot and obstacles be segmented into $n_\mathcal{R}$ and $n_\mathcal{O}$ convex shapes, respectively. Then, we can construct $n_\mathcal{R} \times n_\mathcal{O}$ CBFs, and the same number of CBF constraints, one for each robot-obstacle-rigid-body pair. Then, we combine all the CBF constraints and write it as an element-wise inequality

$$\frac{\partial \mathbf{H}}{\partial x}G(x)u \geq -\gamma \mathbf{H}(x), \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_{i \times j}(x) \end{bmatrix} \in \mathbb{R}^{n_\mathcal{R} \times n_\mathcal{O}} \tag{36}$$

where $i = 1, \cdots, n_\mathcal{R}$, $j = 1, \cdots, n_\mathcal{O}$, and $\mathbf{h}_{i \times j}(x)$ representing the CBF between the i-th robot segment and the j-th obstacle segment. CBF-based quadratic programs (CBFQPs) are commonly used in CBF-based methods to obtain safe control actions. CBFQP utilizes a performance controller π_{perf} to generate a reference control u_{ref}. Then, the CBFQP acts as a safety filter that alters the possibly unsafe u_{ref} in a minimally invasive fashion to find its safe counterpart, i.e.,

$$\min_u \quad \|u - u_{\text{ref}}\|_2^2 \tag{37}$$
$$\text{subject to} \quad \frac{\partial \mathbf{H}}{\partial x}G(x)u \geq -\gamma \mathbf{H}(x).$$

This method works well when the performance controller is non-optimization-based, e.g., PID or control Lyapunov function (CLF) based controllers. For optimization-based controllers, we add the CBF constraints and solve it as a single optimization problem

$$\min_u \quad \mathcal{J}(u) \tag{38}$$
$$\text{subject to} \quad \frac{\partial \mathbf{H}}{\partial x}G(x)u \geq -\gamma \mathbf{H}(x)$$

where $\mathcal{J} : \mathbb{R}^m \to \mathbb{R}$ is the objective function. We will demonstrate the use of these two control methods in Section V. The controller computation flow is shown in Fig. 1. Note that for two obstacles at similar distances with different scales, the CBF values would be smaller for the larger obstacle. However, the CBFQP controller would not favor one obstacle over the other since a feasible solution of the CBFQP would need to satisfy the CBF constraint of each individual obstacle.

V. EXPERIMENTS

This section shows the efficacy of our proposed approach. First, we show the performance of our method on a simulated mobile robot example. Then, we show the application of our method to a 7-degree-of-freedom (DOF) robot manipulator both in simulation and real life. In all our experiments, we
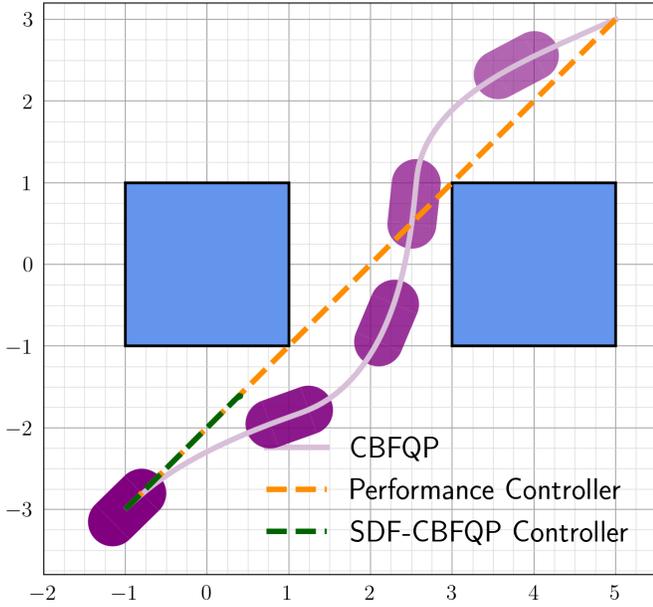
Fig. 3. Generated motion for the mobile robot (purple capsule) in presence of obstacles (blue boxes). The orange dashed curve is the motion generated by the performance controller u_{ref}. The light purple curve represents the generated trajectory of the proposed CBFQP and the green dashed curve represents the generated trajectory of the SDF-CBFQP controller [9] which prematurely terminates (i.e., without reaching the target state).

set $\beta = 1.03$ and $\gamma = 5.0$, though a larger β value would also work. All experiments are performed on a PC with 32GB of RAM and an Intel Core i7 11700 processor.

A. Mobile Robot Example

We demonstrate the efficacy of our approach on a mobile robot example having a capsule shape and the obstacles are represented as polygons. We assume that we can directly control the robot's linear and angular velocity, i.e.,

$$\begin{bmatrix} \dot{p}_x & \dot{p}_y & \dot{\phi} \end{bmatrix}^\top = \begin{bmatrix} v \cos\phi \\ v \sin\phi \\ \omega \end{bmatrix}. \tag{39}$$

The area the robot occupies after scaling it with α can be described using the constraints [15]:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0_{1\times3} & -l/2 & 1 \\ 0_{1\times3} & -l/2 & -1 \end{bmatrix} \begin{bmatrix} p \\ \alpha \\ \delta \end{bmatrix} \in \mathbb{R}_+^2 \tag{40a}$$

$$\begin{bmatrix} 0 \\ -r \end{bmatrix} - \begin{bmatrix} 0_{1\times3} & -R & 0 \\ -I_{3\times3} & 0_{1\times3} & \hat{b}_x \end{bmatrix} \begin{bmatrix} p \\ \alpha \\ \delta \end{bmatrix} \in \mathcal{Q}_4 \tag{40b}$$

with $r \in \mathbb{R}^3$ being the position of the capsule, $\mathcal{Q}_4 \subset \mathbb{R}^4$ the second-order cone, $R \in \mathbb{R}_+$ the radius of the capsule, $L \in \mathbb{R}_+$ the length of the capsule, $l \in \mathbb{R}_+$ the line segment distance, $\delta \in [-\alpha L/2, \alpha L/2]$ a slack variable, $\hat{b}_x = \mathbf{R}[1,0,0]^\top$, and $\mathbf{R}$ being the rotation matrix representing the orientation of the robot. For any point p that satisfies (40), it will belong to the scaled version of the robot body. The area occupied

by the scaled version of the obstacles is represented using the constraint [15]:

$$A_o R_o r_o - \begin{bmatrix} A_o R_o^\top & -b_o \end{bmatrix} \begin{bmatrix} p \\ \alpha \end{bmatrix} \in \mathbb{R}_+, \tag{41}$$

with $A_o \in \mathbb{R}^{n_o \times 2}$ and $b_o \in \mathbb{R}^{n_o}$ being the halfspace constraints, $r_o \in \mathbb{R}^2$ being the position of the obstacle, $R_o \in \mathbb{R}^{2\times2}$ being the rotation matrix of the obstacle, and n_o representing the number of edges for the obstacle. Then, if we use (40) and (41) as the constraints in (15), we can solve for the CBF defined in (16). We use a proportional controller as the performance controller

$$v = K_v \sqrt{(p_{t,x} - p_x)^2 + (p_{t,y} - p_y)^2} \tag{42a}$$

$$\omega = K_\omega \Big[\text{atan2}(p_{t,y} - p_y, p_{t,x} - p_x) - \phi \Big] \tag{42b}$$

with $K_v = 0.5$, $K_\omega = 2.0$, and the target position $(p_{t,x}, p_{t,y}) = (5.0, 3.0)$. The control is obtained by solving (37). The generated motion is shown in Fig. 3. Using our proposed CBF with a simple performance controller, the controller in (37) can generate fairly complex maneuvers to ensure safety. When testing the SDF-based CBF proposed in [9], the mobile robot gets stuck when getting close to the obstacle, and the CBFQP often fails to find a feasible solution. This is due to a reduced feasible set caused by the conservativeness of the approximated partial derivative of the SDF in the CBF constraint. On average, the SDF-CBF takes $9\mu s$ to compute, while our proposed CBF takes $34\mu s$. However, SDF-based CBFs failed to solve the task.

B. 7-DOF Robotic Arm

For the FR3 experiments, we constructed two settings: two walls with a gap on each of them (referred as the `two-walls` task) and three blocks scattered in the workspace (referred as the `three-blocks` task). We encapsulate the links with ellipsoid-shaped bounding boxes for the `three-blocks` task and capsule-shaped bounding boxes for the `two-walls` task. In both settings, the end-effector is encapsulated with a sphere. Since the CoM of the two base links does not have relative translation with respect to the base, we do not need to encapsulate them. This gives us, in total, seven bounding boxes (two bounding boxes encapsulate the fifth link) and $7n_{\mathcal{O}}$ CBFs for each of the experiments. We use a resolved rate controller with joint centering to obtain the desired joint velocity subject to the CBF constraint. The resolved rate controller cost $\mathcal{J}_r$ and the joint-centering cost $\mathcal{J}_c$ are

$$\mathcal{J}_r = \left\| J(\theta)\dot{\theta}_{\text{des}} - \Big[K_p(p_{\text{des}} - p) + \dot{p}_{\text{des}} \Big] \right\|_2^2 \tag{43a}$$

$$\mathcal{J}_c = \left\| \mathcal{N}(\theta)[\dot{\theta}_{\text{des}} - K_p'(\theta_{\text{nominal}} - \theta)] \right\|_2^2 \tag{43b}$$

with $\theta \in \mathbb{R}^7$ representing the joint angles, $\mathcal{N} \in \mathbb{R}^{n_j \times n_j}$ being the nullspace projection matrix $\mathcal{N}(\theta) = \mathbf{I} - J^\dagger(\theta)J(\theta)$ and $J^\dagger \in \mathbb{R}^{n_j \times 6}$ representing the pseudo-inverse of J. Then, we solve for $u = \dot{\theta}_{\text{des}}$ using

$$\min_u \quad \mathcal{J}_r + \epsilon\mathcal{J}_c \tag{44}$$

$$\text{subject to} \quad \frac{\partial \mathbf{H}}{\partial x} u \geq -\gamma\mathbf{H}(x)$$
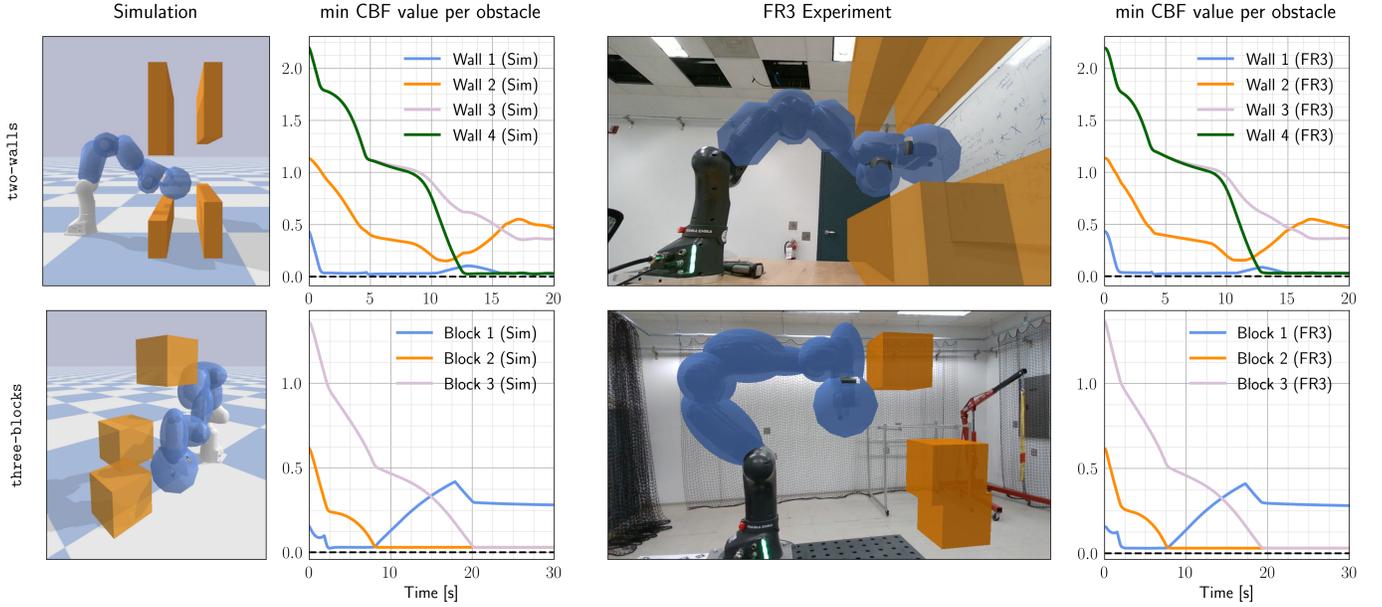
Fig. 4. This figure shows the proposed CBF-based obstacle avoidance controller can ensure safety on an FR3 robot, both in simulation and on the real robot. The simulation is performed using PyBullet with a time step of 1 ms. The walls are numbered in the front upper, front lower, back upper, and back lower order. The blocks are numbered in the top, middle, and bottom order. For each experiment, we record the CBF values of each link-obstacle pair, and for each obstacle, we show the minimum CBF value among all the links. For the real robot experiments, we synthetically show the bounding boxes as blue overlays.
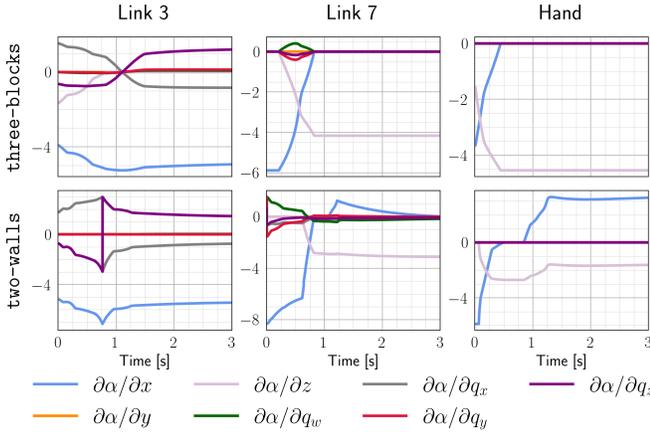
Fig. 5. The computed partial derivatives of the CBF in the real-world robot experiment on the three-blocks and two-walls task. The partial derivatives for link 4, link 5, and link 6 are similar to link 7, thus, in interest of space, they are not shown in the figure above.

with $\epsilon \in \mathbb{R}_+$ being a weighting parameter and $G(x) = \mathbf{I}_{n_j \times n_j}$. The torque command to track $\dot{\theta}_{\text{des}}$ is generated using a derivative controller with gravity compensation

$$\tau = K_d(\dot{\theta}_{\text{des}} - \dot{\theta}) + \mathbf{G}_g(x) \in \mathbb{R}^{n_j} \qquad (45)$$

with $K_d \in \mathbb{R}^{n_j \times n_j}$ being a diagonal matrix and $\mathbf{G}_g : \mathbb{R}^n \to \mathbb{R}^n_j$ the generalized gravitational vector. For the two-walls task, the goal is to reach a target point within the gap of the wall in the back. For the three-blocks task, the robot to reach a point within the lowest block. In both cases, the desired end-effector position is set to be the target point $p_{\text{des}} = p_{\text{target}}$, and the desired end-effector velocity to be zero $\dot{p}_{\text{des}} = \mathbf{0}_{3 \times 1}$.

The results of our proposed method on these two tasks are shown in Fig. 4. For two-walls, the end-effector reaches the target position while ensuring safety. The end-effector reaches the closest point to the target that avoids collision for three-blocks. At each time step, our method computes $7n_{\mathcal{O}}$ CBFs, seven for each obstacle, and we show the change in the minimum value among the seven CBFs for each obstacle in Fig. 4. For two-walls and three-blocks, the average CBF computation time is 0.24ms and 0.20ms, respectively. For both tasks, in simulations and on the real robot, the velocity control is updated at 100Hz.

We empirically show that the proposed CBF is continuously differentiable. The individual elements of $\partial\alpha/\mu$ for the real-world experiment on the three-blocks task are plotted in Fig. 5. We see that all partial derivatives are continuous, echoing our claim in Conjecture 1. For the two-walls task, we see that when continuous differentiability does not hold globally, the approach is still viable in practice and safety can be achieved using our proposed CBF. This is because even when the CBF is not globally continuously differentiable, such discontinuities would intuitively be expected to be over a sparse set (e.g., a set of measure zero). Hence, in practice, the likelihood of encountering the exact poses for which the CBF is not continuously differentiable will be minimal. This is corroborated by the empirical observation that we did not observe any undesired behaviors, e.g., no unsafe behavior or large control commands. The simulation and experimental results can be found at https://youtu.be/WhfFZT1oyJE.

VI. CONCLUSION

This paper presents a systematic and computationally simple approach to construct CBFs using differentiable optimization

based collision detectors. We showed continuous differentia-
bility for strongly convex scaling functions when the gradients
and Hessians exist and are continuous. We conjecture that the
continuous differentiability can be generalized to one object
being strongly convex and the second being only convex. We
experimentally demonstrated the efficacy of our approach on a
mobile robot in simulation and a 7-DOF robot manipulator in
both simulations and on the real robot. In the future, we plan
to extend our approach to consider input and state constraints
and apply them to multi-robot collaboration tasks.

References

[1] B. Dai, H. Huang, P. Krishnamurthy, and F. Khorrami, "Data-efficient control barrier function refinement," in *Proceedings of American Control Conference, San Diego, CA*, May 2023, pp. 3675–3680.
[2] B. Dai, P. Krishnamurthy, A. Papanicolaou, and F. Khorrami, "State constrained stochastic optimal control for continuous and hybrid dynamical systems using DFBSDE," *Automatica*, vol. 155, p. 111146, 2023.
[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proceedings of European Control Conference, Naples, Italy*, June 2019, pp. 3420–3431.
[4] B. Dai, P. Krishnamurthy, A. Papanicolaou, and F. Khorrami, "State constrained stochastic optimal control using LSTMs," in *Proceedings of American Control Conference, New Orleans, LA*, May 2021, pp. 1294–1299.
[5] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *Proceedings of IEEE Conference on Decision and Control, Las Vegas, NV*, December 2016, pp. 827–834.
[6] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *Proceedings of IEEE Conference on Decision and Control, Jeju Island, South Korea*, December 2020, pp. 3717–3724.
[7] C. Li, Z. Zhang, A. Nesrin, Q. Liu, F. Liu, and M. Buss, "Instantaneous local control barrier function: An online learning approach for collision avoidance," *CoRR*, vol. abs/2106.05341, 2021.
[8] B. Dai, P. Krishnamurthy, and F. Khorrami, "Learning a better control barrier function," in *Proceedings of IEEE Conference on Decision and Control, Cancún, Mexico*, December 2022, pp. 945–950.
[9] A. Singletary, S. Kolathaya, and A. D. Ames, "Safety-critical kinematic control of robotic systems," *IEEE Control Systems Letters*, vol. 6, pp. 139–144, 2022.
[10] M. A. Murtaza, S. Aguilera, M. Waqas, and S. Hutchinson, "Safety compliant control for robotic manipulator with task and input constraints," *IEEE Robotics & Automation Letters*, vol. 7, no. 4, pp. 10 659–10 664, 2022.
[11] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *Proceedings of IEEE International Conference on Robotics and Automation, Philadelphia, PA*, May 2022, pp. 286–292.
[12] A. Singletary, W. Guffey, T. G. Molnár, R. W. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics & Automation Letters*, vol. 7, no. 4, pp. 10 954–10 961, 2022.
[13] 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 of Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
[14] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3d game objects," in *Proceedings of Game Developers Conference, San Jose, CA*, vol. 170, 2001.
[15] K. Tracy, T. A. Howell, and Z. Manchester, "Differentiable collision detection for a set of convex primitives," in *Proceedings of IEEE International Conference on Robotics and Automation, London, United Kingdom*, May 2023, pp. 3663–3670.
[16] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *Proceedings of International Conference on Machine Learning, Sydney, Australia*, vol. 70, August 2017, pp. 136–145.
[17] A. Agrawal, B. Amos, S. T. Barratt, S. P. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Proceedings of Annual Conference on Neural Information Processing Systems, Vancouver, Canada*, December 2019, pp. 9558–9570.
[18] U. Dini, *Lezioni di analisi infinitesimale*. Stabilimento Tipografico Successori Fratelli Nistri, Pisa, Italy, 1907, vol. 1.
[19] K. C. Border, *Fixed Point Theorems with Applications to Economics and Game Theory*. Cambridge Univ. Press, Cambridge, U.K., 1985.
[20] J. Schulman, Y. Duan, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.
[21] E. G. Gilbert and D. W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *IEEE Journal of Robotics and Automation*, vol. 1, no. 1, pp. 21–30, 1985.
[22] T. G. Molnár, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics & Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022.