

# Duality-based Convex Optimization for Real-time Obstacle Avoidance between Polytopes with Control Barrier Functions

Akshay Thirugnanam, Jun Zeng, and Koushil Sreenath

**Abstract**—Developing controllers for obstacle avoidance between polytopes is a challenging and necessary problem for navigation in tight spaces. Traditional approaches can only formulate the obstacle avoidance problem as an offline optimization problem. To address these challenges, we propose a duality-based safety-critical optimal control using nonsmooth control barrier functions for obstacle avoidance between polytopes, which can be solved in real-time with a QP-based optimization problem. A dual optimization problem is introduced to represent the minimum distance between polytopes and the Lagrangian function for the dual form is applied to construct a control barrier function. We validate the obstacle avoidance with the proposed dual formulation for L-shaped (sofa-shaped) controlled robot in a corridor environment. We demonstrate real-time tight obstacle avoidance with non-conservative maneuvers on a moving sofa (piano) problem with nonlinear dynamics.

## I. INTRODUCTION

### A. Motivation

Achieving safety-critical navigation for autonomous robots in an environment with obstacles is a vital problem in robotics research. Recently, control barrier functions (CBFs) together with quadratic program (QP) based optimizations have become a popular method to design safety-critical controllers. In this paper, we propose a novel duality-based approach to formulate the obstacle avoidance problem between polytopes into QPs in the continuous domain using CBFs, which could then be deployed in real-time.

### B. Related Work

1) *Control Barrier Functions*: One approach to provide safety guarantees for obstacle avoidance in control problems is to draw inspiration from control barrier functions. CBF-QPs [2] permit us to find the minimum deviation from a given feedback control input to guarantee safety. The method of CBFs can also be generalized for high-order systems [3], [4], discrete-time systems [5]–[7] and input-bounded systems [8]–[11]. It must be noted that early work on nonovershooting control in [12] could also have been used to obtain results similar to control barrier functions. Specifically, CBFs are widely used for obstacle avoidance [13]–[18] with a variety of applications for autonomous robots,

This work was supported by National Science Foundation Grant CMMI-1931853.

All authors are with Hybrid Robotics Group at the Department of Mechanical-Engineering, UC Berkeley, USA. {akshay\_t, zengjunsjtu, koushils}@berkeley.edu

A descriptive video with animations can be found at <https://youtu.be/rpwre3IYPJE>. The appendix for proofs is excluded in this version and can be found in the full version of this paper [1].

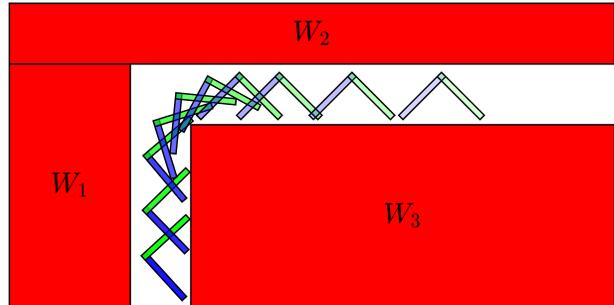


Fig. 1: Snapshots of solving the moving sofa (piano) problem using our proposed formulation. It's shown that the controlled object can maneuver through a tight corridor whose width is smaller than the diagonal length of the controlled object, which cannot be achieved if we over-approximate these rectangle-shaped regions into spheres.

including autonomous cars [19], aerial vehicles [20] and legged robots [21]. The shapes of robots and obstacles are usually approximated as points [19], paraboloids [22] or hyper-spheres [23], where the distance function can be calculated explicitly as an analytic expression from their geometric configuration. The distance functions for these shapes are differentiable and can be used as control barrier functions to construct a safety-critical optimal control problem.

However, these approximations usually over-estimate the dimensions of the robot and obstacles, e.g., a rectangle is approximated as the smallest circle that contains it. When a tight-fitting obstacle avoidance motion is expected, as shown in Fig. 1, robots and obstacles are usually approximated as polytopes. While this makes maneuvers less conservative for obstacle avoidance, computing the distance between two polytopes requires additional effort [24]. Moreover, since this distance is not in an explicit form, it cannot be used directly as a CBF. Furthermore, the distance between polytopes is non-differentiable [25], which necessitates the use of nonsmooth control barrier functions (NCBFs) to guarantee safety [26], [27].

2) *Obstacle Avoidance between Polytopes*: We narrow our discussions about obstacle avoidance between polytopes into optimization-based approaches. In [28], obstacle avoidance between rectangle-shaped objects in an offline planning problem is studied, where collision avoidance is ensured by keeping all vertices of the controlled object outside the obstacle. Generally, when controlled objects are polyhedral, the collision avoidance constraints can be reformulated with integer variables [29]. This method applies well for linear systems using mixed-integer programming but cannot be deployed as real-time controllers for general nonlinear systems

due to the complexity arising from integer variables. The obstacle avoidance problem between convex regions could also be solved by using sequential programming [30], where penalizing collisions with a hinge loss is considered through an offline optimization problem.

Recently, a duality-based approach [31] was introduced to non-conservatively reformulate obstacle avoidance constraints as a set of smooth non-convex ones, which is validated on navigation problems in tight environments [32]–[35]. This idea does optimize the computational time compared with other ideas, but nonlinear non-convex programming is still involved. Moreover, this approach can only be used for offline planning for nonlinear systems. This philosophy has been extended into discrete-time control barrier functions (DCBFs) to enforce obstacle avoidance constraints between polytopes in real-time [36]. However, the resulting DCBF formulation is still non-convex with non-convex DCBF or nonlinear system dynamics, and the computation time of the DCBF formulation might not be sufficiently low enough to be real-time. On the other hand, a continuous-time formulation can result in a convex optimization formulation even for nonlinear systems, which leads to faster computation times. Thus, continuous-time obstacle avoidance between polytopes requires a computationally efficient implementation, such as CBF-QPs and proper analysis on the nonsmooth nature of distance between polytopes to guarantee safety. To summarize, real-time obstacle avoidance between polytopes with convex programming for general nonlinear systems is still a challenging problem.

### C. Contributions

The contributions of this paper are as follows:

- We propose a novel approach to reformulate a minimization problem for obstacle avoidance between polytopes for nonlinear affine systems into a duality-based quadratic program with CBFs.
- We establish the obstacle avoidance algorithm for the minimum distance between polytopes under a QP-based control law that guarantees safety, where the nonsmooth nature of the minimum distance between polytopes is resolved in the dual space. This formulation is used for real-time safety-critical obstacle avoidance.
- Our proposed algorithm demonstrates real-time obstacle avoidance at 50 Hz in the *moving sofa (piano) problem* [37] with nonlinear dynamics, where an L-shaped controlled object can maneuver safely in a tight L-shaped corridor, whose width is less than the diagonal length of the controlled object.

## II. BACKGROUND

We consider  $N$  robots, with the  $i$ -th robot having states  $x^i \in \mathcal{X} \subset \mathbb{R}^n$  and nonlinear, control affine dynamics:

$$\dot{x}^i(t) = f^i(x^i(t)) + g^i(x^i(t))u^i(t), \quad i \in [N], \quad (1)$$

where,  $u^i(t) \in \mathcal{U} \subset \mathbb{R}^m$ ,  $f^i : \mathcal{X} \rightarrow \mathbb{R}^n$  and  $g^i : \mathcal{X} \rightarrow \mathbb{R}^{n \times m}$  are continuous, and  $[N] = \{1, \dots, N\}$ . We assume  $\mathcal{X}$  to be a connected set and  $\mathcal{U}$  a convex, compact set. While the

dimensions of the states and inputs for each system can be different, we assume them to be the same across the systems for simplicity. Throughout the paper, superscripts of variables denote the robot index and subscripts denote the row index of vectors or matrices.

### A. Closed-loop Trajectory for Discontinuous Inputs

Polytopes have non-differentiable surfaces, and the minimum distance between polytopes could be non-differentiable at these points of non-differentiability. Hence, enforcing safety constraints with the nonsmooth distance could result in the loss of continuity property of the feedback control. Since  $f^i$  and  $g^i$  might not be Lipschitz continuous and  $u^i(t)$  may be discontinuous, the solution of (1) need not be unique or even well-defined. In this case, to have a well-defined notion of a solution to (1), the dynamical system (1) is turned into a differential inclusion. Let  $u^i : \mathcal{X} \rightarrow \mathcal{U}$  be some measurable feedback control law. A valid solution for the closed loop trajectory is defined via the Filippov map [27] as

$$\begin{aligned} \dot{x}^i(t) &\in F[f^i + g^i u^i](x^i(t)) \\ &:= \text{co}\left\{ \lim_{k \rightarrow \infty} (f^i + g^i u^i)(x_{(k)}) : x_{(k)} \rightarrow x^i(t), x_{(k)} \notin \mathcal{Q}_f, \mathcal{Q} \right\} \end{aligned} \quad (2)$$

where ‘co’ stands for convex hull,  $x_{(k)}$  denotes the  $k$ -th element of the sequence  $\{x_{(k)}\}$ ,  $\mathcal{Q}_f$  is a system-dependent zero-measure set, and  $\mathcal{Q}$  is any zero-measure set. The resulting map  $F[f^i + g^i u^i] : \mathcal{X} \rightarrow 2^{\mathbb{R}^n}$  is a non-empty convex, compact, and upper semi-continuous set-valued map. Here  $2^{\mathbb{R}^n}$  denotes the power set of  $\mathbb{R}^n$ . For a set-valued map  $\Gamma : \mathcal{X} \rightarrow 2^{\mathbb{R}^n}$  to be upper semi-continuous at  $a \in \mathcal{X}$ , we require,  $\forall \{a_{(m)}\} \in \mathcal{X}$  and  $\{b_{(m)}\}$  such that  $b_{(m)} \in \Gamma(a_{(m)})$ ,  $\lim_{m \rightarrow \infty} a_{(m)} = a$  and  $\lim_{m \rightarrow \infty} b_{(m)} = b \Rightarrow b \in \Gamma(a)$ . Then for all  $x_0^i \in \mathcal{X}$ , a Filippov solution exists for the differential inclusion (2) with  $x^i(0) = x_0^i$  [38, Prop. 3]. A Filippov solution on  $[0, t]$  is an absolutely continuous map  $x^i : [0, T] \rightarrow \mathcal{X}$  which satisfies (2) for almost all  $t \in [0, T]$ . Throughout the paper, the term “almost all” means for all but on a set of measure zero.

### B. Minimum Distance between Polytopes

For robot  $i$ , we define the polytope  $\mathcal{P}^i(x^i)$  as the  $l$ -dimensional physical domain associated to the robot at state  $x^i \in \mathcal{X}$  with

$$\mathcal{P}^i(x^i) := \{z \in \mathbb{R}^l : A^i(x^i)z \leq b^i(x^i)\}, \quad (3)$$

where  $A^i : \mathcal{X} \rightarrow \mathbb{R}^{r^i \times l}$  and  $b^i : \mathcal{X} \rightarrow \mathbb{R}^{r^i \times 1}$  represent the half spaces that define the geometry of robot  $i$  at some state, shown in Fig. 2. We assume that the following properties hold for  $A^i$  and  $b^i$ :

- $\exists R, r > 0$  such that  $\forall x^i \in \mathcal{X}$ ,  $\exists c \in \mathbb{R}^l$  such that  $B_r(c) \subset \mathcal{P}^i(x^i) \subset B_R(c)$ , where  $B_r(c)$  represents the open ball with radius  $r$  centered at  $c$ .
- $A^i, b^i$  are continuously differentiable, and  $\forall x \in \mathcal{X}$ , the set of inequalities  $A^i(x)z \leq b^i(x)$  does not contain any redundant inequality.
- $\forall x \in \mathcal{X}$ , the set of active constraints at any vertex of  $\mathcal{P}^i(x)$  are linearly independent.

The first assumption requires that the geometry of the robot  $\mathcal{P}^i(x)$  be uniformly bounded with a non-empty interior for all  $x \in \mathcal{X}$ . This assumption guarantees regularity conditions [39, Def. 2.3] are met for minimum distance computations, which in turn guarantee the essential conditions of continuity [39, Thm. 2.3] and differentiability [39, Thm. 2.4] of minimum distance. The last assumption requires that no more than  $l$  half spaces intersect at any vertex. As an example, a square pyramid in 3D does not satisfy this criterion. Any polytope that does not satisfy this assumption can be tessellated into smaller polytopes, such as tetrahedra.

The square of the minimum distance between  $\mathcal{P}^i(x^i)$  and  $\mathcal{P}^j(x^j)$  is defined as  $h^{ij}(x^i, x^j)$ , where  $h^{ij} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  can be computed using the following QP:

$$\begin{aligned} h^{ij}(x^i, x^j) &:= \min_{\{z^i, z^j\}} \|z^i - z^j\|_2^2 \\ \text{s.t. } &A^i(x^i)z^i \leq b^i(x^i), A^j(x^j)z^j \leq b^j(x^j), \\ &z^i, z^j \in \mathbb{R}^l. \end{aligned} \quad (4)$$

Note that compared to the prior work on CBFs, the distance  $h^{ij}$  is implicit and is a solution of a minimization problem. By the regularity and smoothness assumptions on the polytopes,  $h^{ij}$  is locally Lipschitz continuous [40, Lem. 1]. Variables  $z^i, z^j \in \mathbb{R}^l$  denote points inside the polytopes  $\mathcal{P}^i(x^i)$  and  $\mathcal{P}^j(x^j)$  respectively. Since the feasible sets of (4) are non-empty (by assumption), convex, and compact, the solution to the QP (4) always exists and is non-negative. When the robots intersect with each other the minimum distance is uniformly zero, and there is no measure of penetration between the polytopes.

### C. Nonsmooth Control Barrier Functions

For obstacle avoidance, we want to design a controller such that the minimum distance between any pair of robots  $i$  and  $j$  should be strictly greater than 0. We define a safe set of states  $\mathcal{S}^{ij}$  as the zero-superlevel set of the minimum distance between robots  $i$  and  $j$ ,  $h^{ij}$  [2],

$$\mathcal{S}^{ij} := \{(x^i, x^j) : h^{ij}(x^i, x^j) > 0\}^c, \quad (5)$$

where  $(\cdot)^c$  denotes closure of a set. Note that since  $h^{ij}(x^i, x^j) = 0$  for intersecting robot geometries, we use closure to obtain the correct safe set. The closed-loop system is considered safe if  $(x^i(t), x^j(t)) \in \mathcal{S}^{ij} \forall t \in [0, T]$ , where  $T$  is time till which the solution is defined.

Let  $u^i(x^i), u^j(x^j) \in \mathcal{U}$  be feedback control laws that are measurable, with corresponding Filippov solutions  $x^i(t), x^j(t)$  for  $t \in [0, T]$ . Since  $h^{ij}$  is locally Lipschitz and the state trajectories are absolutely continuous,  $h^{ij}(t) := h^{ij}(x^i(t), x^j(t))$  is an absolutely continuous function, and is thus differentiable at almost all  $t \in [0, T]$ .

**Lemma 1.** [26, Lem. 2] Let  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  be a locally Lipschitz class- $\mathcal{K}$  function. If

$$\dot{h}^{ij}(t) \geq -\alpha(h(t)) \quad (6)$$

for almost all  $t \in [0, T]$  and  $h(0) > 0$ , then  $h(t) > 0 \forall t \in [0, T]$ , making the system safe.

In this case the absolutely continuous function  $h^{ij}(t)$  is called a Nonsmooth Control Barrier Function (NCBF), which is a generalization of Control Barrier Functions (CBFs) to nonsmooth functions. The constraint (6) is called the NCBF constraint. In the following section we derive a safety-critical feedback control law that satisfies this property.

**Remark 1.** For simplicity of discussion, the later analysis will be illustrated for a pair of robots  $i$  and  $j$ , and results will be generalized where necessary. Further, for simplicity of notation, we denote  $x := (x^i, x^j)$ ,  $h(x) := h^{ij}(x^i, x^j)$ ,  $\mathcal{S} := \mathcal{S}^{ij}$ , and  $u := (u^i, u^j)$  for the pair of systems  $i$  and  $j$ .

## III. NCBFS FOR POLYTOPES

In this section, we will illustrate a general approach for imposing NCBF constraints for polytopes. In order to enforce the NCBF constraints, we need to be able to write  $\dot{h}(t)$  explicitly in terms of  $\dot{A}^i(t), \dot{A}^j(t), \dot{b}^i(t), \dot{b}^j(t)$ , which in turn depend on  $u$ . In the following sub-section, we attempt to construct an explicit formula for  $\dot{h}(t)$ .

### A. Primal Approach

To explicitly compute  $\dot{h}(t)$ , we first need to show that it exists. Let  $x(t), t \in [0, T]$  be the Filippov solution corresponding to a feedback control law  $u(x)$ . Let  $\mathcal{O}(t)$  be the set of all optimal solutions to (4) at time  $t$ . For any pair of optimal solutions  $(z^{*i}(t), z^{*j}(t)) \in \mathcal{O}(t)$ , let  $s^*(t) := z^{*i}(t) - z^{*j}(t)$  be the separating vector.  $s^*(t)$  is fundamental for the distance formulation between polytopes since it is related to the minimum distance  $h^{ij}(t) = \|s^*(t)\|_2^2$ . We will next prove that  $s^*(t)$  is unique, continuous, and right-differentiable.

**Lemma 2.** For all  $t \in [0, T]$ , the separating vector  $s^*(t)$  is unique for all pairs of primal optimal solutions  $(z^{*i}(t), z^{*j}(t))$ .

*Proof.* The feasible set of (4) is convex and compact, and the cost function  $\|z^i - z^j\|_2^2$  is convex. Projecting the feasible set onto the  $z^i + z^j = 0$  surface, the resulting set is also convex and compact. Note that  $s = z^i - z^j \in \mathbb{R}^l$  represents the projected co-ordinates on the  $z^i + z^j = 0$  surface. The resulting cost function is  $\|s\|_2^2$ , is strictly convex, and therefore, the optimal solution  $s^*(t)$  of the projected problem exists and is unique. Since for any pair of primal optimal solutions  $(z^{*i}(t), z^{*j}(t))$ ,  $z^{*i}(t) - z^{*j}(t)$  is an optimal solution to the projected problem,  $z^{*i}(t) - z^{*j}(t)$  equals  $s^*(t)$  for all pairs of optimal solutions of (4). ■

For all  $(z^{*i}(t), z^{*j}(t)) \in \mathcal{O}(t)$ , we define  $\text{Act}^i(t) \subset [r^i]$  ( $\text{Act}^j(t) \subset [r^j]$ ) as the set of indices of constraints that are active for all  $z^{*i}(t)$  ( $z^{*j}(t)$ ) for  $\mathcal{P}^i(t)$  ( $\mathcal{P}^j(t)$ ). Then,

$$\begin{aligned} \text{Aff}^i(t) &:= \{z^i : A_{\text{Act}^i(t)}^i(t)z^i = b_{\text{Act}^i(t)}^i(t)\} \\ \text{Aff}^j(t) &:= \{z^j : A_{\text{Act}^j(t)}^j(t)z^j = b_{\text{Act}^j(t)}^j(t)\} \end{aligned} \quad (7)$$

represent two parallel affine spaces such that the minimum distance between robots  $i$  and  $j$  at  $t$  is the distance between  $\text{Aff}^i(t)$  and  $\text{Aff}^j(t)$ . These affine spaces can be points, hyperplanes, or even the entire space if the two polytopes intersect. An example is pictorially depicted in Fig. 2.

We now assume that for almost all  $t \in [0, T]$ ,  $\exists \epsilon > 0$  such that the  $\dim(\text{Aff}^i(t))$ ,  $\dim(\text{Aff}^j(t))$ , and dimension of the orthogonal subspace common between  $\text{Aff}^i(t)$  and  $\text{Aff}^j(t)$  are constant for  $\tau \in [t, t + \epsilon)$ . This is true when the set of times when states of the system oscillate infinitely fast has zero-measure. In practice the states of the system do not oscillate infinitely fast due to limited control frequency and inertia of the system. Then, under this assumption, we have the following result:

**Lemma 3.** The separating vector  $s^*(t)$  is continuous and right-differentiable for almost all  $t \in [0, T]$ .

*Proof.* Let  $t$  be a time when the dimensions of the affine spaces and null space is constant for  $\tau \in [t, t + \epsilon)$ . The separating vector  $s^*(\tau)$  is the unique vector from  $\text{Aff}^i(\tau)$  to  $\text{Aff}^j(\tau)$  that is perpendicular to both of them. These three constraints (that define  $s^*(\tau)$  and establish orthogonality of  $s^*(\tau)$  to  $\text{Aff}^i(\tau)$  and  $\text{Aff}^j(\tau)$ ) can be written in the form of a system of linear equalities with the matrix right-differentiable at  $\tau = t$ . By the assumption on constant dimensions of the above spaces ( $\text{Aff}^i(t)$ ,  $\text{Aff}^j(t)$ , and their common orthogonal subspace), this matrix has constant rank for  $\tau \in [t, t + \epsilon)$ . By Lem. 2, there is at least one solution to this system  $\forall \tau \in [t, t + \epsilon)$ . A right-differentiable solution to  $s^*(\tau)$  can be found from this linear system using Gauss elimination. Since  $s^*(t)$  is unique by Lem. 2,  $s^*(t)$  must be right-differentiable, and thus continuous, at  $t$ . ■

To impose the NCBF constraint we then compute  $\dot{h}(t) = \lim_{\delta \rightarrow 0^+} 1/\delta(h(t + \delta) - h(t))$  as:

$$\dot{h}(t^+) = \frac{d}{d\tau} \|z^{*i}(\tau) - z^{*j}(\tau)\|_2^2 \Big|_{\tau=t^+}. \quad (8)$$

Although,  $h(t) = \|s^*(t)\|_2^2$  is right-differentiable for almost all times, the primal optimal solutions  $z^{*i}(t)$ ,  $z^{*j}(t)$  could be non-differentiable or even discontinuous. So,  $\dot{h}(t)$  cannot be written explicitly as a minimization problem from (8), since  $\dot{z}^{*i}(t)$  and  $\dot{z}^{*j}(t)$  need not be well-defined. This leads us to consider the dual formulation instead.

As a motivation, consider enforcing the collision avoidance constraint  $\mathcal{P}^i \cap \mathcal{P}^j = \emptyset$  explicitly. Constraining the distance between any two points in  $\mathcal{P}^i$  and  $\mathcal{P}^j$  to be greater than 0 is not sufficient, since they may not be the closest points. This is due to the fact that (4) is a minimization problem. However, constraining a plane to separate  $\mathcal{P}^i$  and  $\mathcal{P}^j$  is sufficient to guarantee  $\mathcal{P}^i \cap \mathcal{P}^j = \emptyset$  even if it is not the maximal separating plane. The dual formulation of (4) allows us to explicitly compute this separating plane constraint, which can be used in the NCBF constraint (4). The reason for using the dual formulation instead of the primal is further elaborated upon later in Remark 2.

## B. Dual Formulation

The dual program of a minimization problem is a maximization problem in terms of the corresponding dual variables. For a quadratic optimization problem, as in (4), the dual program has the same optimal solution as that of (4).

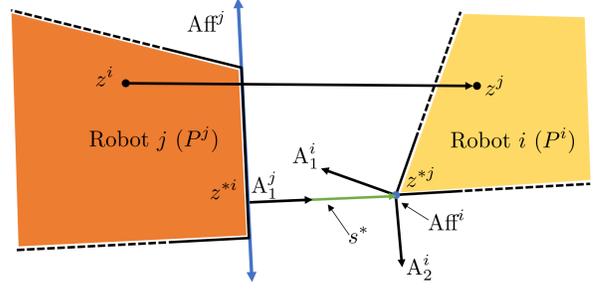


Fig. 2: At any two configurations, the minimum distance between the robots  $i$  and  $j$  is the same as the minimum distance between the affine spaces  $\text{Aff}^i$  and  $\text{Aff}^j$ , illustrated in blue. These spaces are affine extensions of some two faces of the robots. The points on robot  $i$  and  $j$  with the least distance are  $z^{*i}$  and  $z^{*j}$  respectively, and  $s^*$  represents the vector with the smallest norm. The dynamics of the minimum distance between the robots is a hybrid system, with the discrete states being the pair of faces of the robots generating these affine spaces. For each discrete state, the minimum distance varies smoothly as the distance between the two affine spaces.

So, differentiating the dual program will give  $\dot{h}(t)$  as a maximization problem. Since any feasible solution to a maximization problem gives us a lower bound of the optimum, we use the dual problem of (4) to get a lower bound of  $\dot{h}(t)$ . This enables us to express the NCBF constraint as a feasibility problem rather than an optimization problem.

To obtain the dual program, we first transform the constrained optimization problem (4) to an unconstrained one by adding the constraints to the cost with weights  $\lambda^i$  and  $\lambda^j$ , which are called the dual variables. The unconstrained problem is optimized in terms of  $z^i$  and  $z^j$  to obtain the Lagrangian function  $L(\lambda^i, \lambda^j)$  as

$$L(\lambda^i, \lambda^j) = -\frac{1}{4} \lambda^i A^i(x^i) A^i(x^i)^T \lambda^i T - \lambda^i b^i(x^i) - \lambda^j b^j(x^j). \quad (9)$$

The dual program is then defined as the maximization of the Lagrangian function.

**Lemma 4.** The dual program corresponding to (4) is:

$$\begin{aligned} h(x) &= \max_{\{\lambda^i, \lambda^j\}} L(\lambda^i, \lambda^j) \\ \text{s.t. } & \lambda^i A^i(x^i) + \lambda^j A^j(x^j) = 0, \lambda^i, \lambda^j \geq 0. \end{aligned} \quad (10)$$

*Proof.* The proof is provided in Appendix A in the full version of this paper [1]. ■

Since an optimal solution to (4) always exists, an optimal solution to (10) also always exists. Let  $(z^{*i}(t), z^{*j}(t)) \in \mathcal{O}(t)$  and the active set of constraints at  $z^{*i}(t)$  be  $\text{Act}^i(z^{*i}(t), t)$ . Linear independence constraint qualification (LICQ) is said to be held at  $z^{*i}(t)$  if  $A_{\text{Act}^i(z^{*i}(t), t)}^i$  is full rank [41, Def. 2.1]. By the linear independence assumption on  $\mathcal{P}^i(t)$ , since the set of the active constraints at any vertex of  $\mathcal{P}^i(t)$  are linearly independent,  $A_{\text{Act}^i(z^{*i}(t), t)}^i$  is also full-rank for all  $z^{*i}(t)$  [41, Lem. 2.1]. Then, the primal problem (4) is considered non-degenerate and there exists a unique dual optimal solution for (10) [41, Lem. 2.2]. The dual optimal solution  $(\lambda^{*i}(t), \lambda^{*j}(t))$  along with any primal

optimal solution  $(z^{*i}(t), z^{*j}(t))$  must then satisfy the KKT optimality conditions at  $t$ :

$$\begin{aligned} 2\lambda^{*i}(t)A^i(t) &= z^{*i}(t)^T - z^{*j}(t)^T = s^*(t)^T, \quad (11) \\ 2\lambda^{*j}(t)A^j(t) &= z^{*j}(t)^T - z^{*i}(t)^T = -s^*(t)^T, \\ \lambda_k^{*i}(t) &= 0 \quad \text{for } k \notin \text{Act}^i(z^{*i}(t), t), \\ \lambda_k^{*j}(t) &= 0 \quad \text{for } k \notin \text{Act}^j(z^{*j}(t), t). \end{aligned}$$

Then, by LICQ,  $A_{\text{Act}^i(t)}^i$  and  $A_{\text{Act}^j(t)}^j$  have full rank and the non-zero components of the dual optimal solutions at  $t$  can be written explicitly as:

$$\lambda^{*i}(t) = s^*(t)^T A_{\text{Act}^i(t)}^{i\dagger}, \quad \lambda^{*j}(t) = -s^*(t)^T A_{\text{Act}^j(t)}^{j\dagger}, \quad (12)$$

where  $(\cdot)^\dagger$  is the generalized inverse. By assumption  $\text{Aff}^i(\tau)$  and  $\text{Aff}^j(\tau)$  have full rank for  $\tau \in [t, t + \epsilon)$  for almost all  $t \in [0, T]$ , and thus  $\lambda^{*i}(t)$  and  $\lambda^{*j}(t)$  are right-differentiable at almost all  $t \in [0, T]$ . Since  $\lambda^{*i}(t), \lambda^{*j}(t), A^i(t)$  and  $A^j(t)$  are right-differentiable, we can differentiate the constraints of (10). Finally, we can explicitly write a linear program, which is obtained by differentiating the cost and constraints of (10), to calculate  $\dot{h}(t)$  as:

**Lemma 5.** Let,

$$\begin{aligned} g(t) &= \max_{\{\dot{\lambda}^i, \dot{\lambda}^j\}} \dot{L}(t, \lambda^{*i}(t), \lambda^{*j}(t), \dot{\lambda}^i, \dot{\lambda}^j) \\ \text{s.t.} \quad &\dot{\lambda}^i A^i(t) + \lambda^{*i}(t) \dot{A}^i(t) \\ &+ \dot{\lambda}^j A^j(t) + \lambda^{*j}(t) \dot{A}^j(t) = 0, \quad (13) \\ &\dot{\lambda}_k^i \geq 0 \quad \text{if } \lambda^{*i}(t)_k = 0, \\ &\dot{\lambda}_k^j \geq 0 \quad \text{if } \lambda^{*j}(t)_k = 0. \end{aligned}$$

where  $\dot{L}(t, \lambda^i, \lambda^j, \dot{\lambda}^i, \dot{\lambda}^j)$  represents the time-derivative of Lagrangian function  $L(\lambda^i, \lambda^j)$  and is as follows,

$$\begin{aligned} \dot{L} &= -\frac{1}{2} \lambda^i A^i(t) A^i(t)^T \dot{\lambda}^i - \frac{1}{2} \lambda^j A^j(t) A^j(t)^T \dot{\lambda}^j \\ &- \dot{\lambda}^i b^i(t) - \dot{\lambda}^j b^j(t) - \dot{\lambda}^i b^j(t) - \dot{\lambda}^j b^i(t). \quad (14) \end{aligned}$$

Then, for almost all  $t \in [0, T]$ ,  $\dot{h}^{ij}(t) = g(t)$ .

*Proof.* Since  $(\lambda^{*i}(t), \lambda^{*j}(t))$  is the optimal solution to (10), its derivative  $(\dot{\lambda}^{*i}(t), \dot{\lambda}^{*j}(t))$  is a feasible solution to (13) for almost all  $t \in [0, T]$ . So,  $\dot{h}^{ij}(t) = \frac{d}{dt} L(\lambda^{*i}(t), \lambda^{*j}(t)) \leq g(t)$  for almost all  $t \in [0, T]$ . Let  $(\bar{\lambda}^i, \bar{\lambda}^j)$  be a feasible solution to (13). We can integrate the constraints of (13) to find  $(\bar{\lambda}^i(\tau), \bar{\lambda}^j(\tau)), \tau \in [t, t + \epsilon)$  which satisfy:

$$\begin{aligned} (\bar{\lambda}^i(t), \bar{\lambda}^j(t)) &= (\lambda^{*i}(t), \lambda^{*j}(t)) \quad (15) \\ (\bar{\lambda}^i(t), \bar{\lambda}^j(t)) &\text{ is dual feasible for (10)} \end{aligned}$$

So,  $(\bar{\lambda}^i(\tau), \bar{\lambda}^j(\tau))$  are dual feasible and have cost less than  $h^{ij}(\tau)$ , i.e.  $L(\bar{\lambda}^i(\tau), \bar{\lambda}^j(\tau)) \leq h^{ij}(\tau) \quad \forall \tau \in [t, t + \epsilon)$  and  $L(\bar{\lambda}^i(t), \bar{\lambda}^j(t)) = h^{ij}(t)$ . Differentiating the cost yields  $\dot{h}^{ij}(t) \geq \dot{L}(t, \lambda^{*i}(t), \lambda^{*j}(t), \dot{\lambda}^i, \dot{\lambda}^j)$  for all feasible  $(\dot{\lambda}^i, \dot{\lambda}^j)$ , and thus  $\dot{h}^{ij}(t) \geq g(t)$ . So,  $g(t) = \dot{h}^{ij}(t)$  for almost all  $t \in [0, T]$ . ■

Based on the linear program in (13), we can conservatively implement the NCBF constraint by enforcing, for some  $(\dot{\lambda}^i, \dot{\lambda}^j)$  feasible for (13),

$$\dot{L}(t, \lambda^{*i}(t), \lambda^{*j}(t), \dot{\lambda}^i, \dot{\lambda}^j) \geq -\alpha(h(t)). \quad (16)$$

Lem. 5 then guarantees that

$$\dot{h}(t) \geq \dot{L}(t, \lambda^{*i}(t), \lambda^{*j}(t), \dot{\lambda}^i, \dot{\lambda}^j) \geq -\alpha(h(t)) \quad (17)$$

which is the required NCBF constraint.

**Remark 2.** Due to the direction of inequality required for the NCBF constraint,  $\dot{h}(t)$  needs to be expressed as a maximization problem. This is the primary motivation for considering the dual problem, since writing  $\dot{h}(t)$  using the primal problem results in a minimization problem (8).

We use (16) to motivate a feedback control law to guarantee safety of the system. The input  $u$  implicitly affects  $\dot{L}$  via the derivatives of the boundary matrices  $A^i, A^j, b^i, b^j$ . Note that  $\dot{L}$  is affine in  $\dot{\lambda}^i, \dot{\lambda}^j$ , and  $u$ . So, (16) is a linear constraint in  $\dot{\lambda}^i, \dot{\lambda}^j$ , and  $u$ . So,  $\forall x \in \mathcal{S}$ , (10) is used to compute  $h(x)$ ,  $\lambda^{*i}(x)$ , and  $\lambda^{*j}(x)$  and the optimal solution of following quadratic program is used as the feedback control:

$$u^*(x) = \underset{\{u, \dot{\lambda}^i, \dot{\lambda}^j\}}{\text{argmin}} \|u - u^{nom}(x)\|_Q^2 \quad (18a)$$

$$\text{s.t.} \quad \dot{L}(t, \lambda^{*i}(x), \lambda^{*j}(x), \dot{\lambda}^i, \dot{\lambda}^j, u) \geq -\alpha(h(x) - \epsilon_1^2) \quad (18b)$$

$$\begin{aligned} \dot{\lambda}^i A^i(x) + \lambda^{*i}(x) (\mathcal{L}_{f^i} A^i(x) + \mathcal{L}_{g^i} A^i(x) u) \\ + \lambda^{*j}(x) (\mathcal{L}_{f^j} A^j(x) + \mathcal{L}_{g^j} A^j(x) u) = -\dot{\lambda}^j A^j(x) \end{aligned} \quad (18c)$$

$$\dot{\lambda}_k^i \geq 0 \text{ if } \lambda^{*i}(x)_k < \epsilon_2, \quad \dot{\lambda}_k^j \geq 0 \text{ if } \lambda^{*j}(x)_k < \epsilon_2, \quad (18d)$$

$$|\dot{\lambda}^i| \leq M, |\dot{\lambda}^j| \leq M, \quad (18e)$$

$$u \in \mathcal{U}, \quad (18f)$$

where  $u^{nom}(x)$  is a non-safe nominal feedback control law,  $\mathcal{L}_{(\star)}(\cdot)$  represents the Lie derivative of  $(\cdot)$  along  $(\star)$ ,  $Q \succ 0$  is the cost matrix,  $M$  is a large number, and  $\epsilon_1$  and  $\epsilon_2 > 0$  are small constants.  $u^{nom}(x)$  can be obtained by a control Lyapunov function or by any tracking controller. Note that (18) is a feedback law which does not assume existence of Filippov solutions, since it is only a function of  $x$  and not  $t$ .

**Remark 3.** Since  $h(x)$  is quadratic in nature, its gradient (if it exists) can be zero at  $\partial\mathcal{S}$ , which can affect forward invariance of  $\mathcal{S}$  [2, Rem. 5]. For example, in 1D, let  $\dot{x} = u$  and  $h(x) = x^2$ . Then at  $x = 0$ , the NCBF constraint reduces to  $0 \cdot u \geq 0$ , which is true  $\forall u \in \mathcal{U}$ . This would imply that the system remains safe irrespective of the input, which is not true. This problem can be solved by setting  $h(x) = x^2 - \epsilon_1^2$ , which would result in non-zero gradient at  $x = \epsilon_1$ . So, we consider an  $\epsilon_1 > 0$  and the  $\epsilon_1^2$ -level set of  $h$  as

$$\Omega_{\epsilon_1^2} = \{x : h(x) = \epsilon_1^2\}. \quad (19)$$

We can then redefine  $h(x)$  as  $(h(x) - \epsilon_1^2)$ . If the new  $h$  satisfies (6), then the  $\epsilon_1^2$ -superlevel set is safe.

**Remark 4.** The constant  $\epsilon_2$  is used to ensure upper semi-continuity of the feasible set of control inputs for (18). This is similar to the almost-active gradient method used to prove safety in [27, Thm. 3].

We now prove that (18) results in system safety.

**Theorem 1.** Let  $x(0) \in \mathcal{S}$  and (18) be feasible  $\forall x \in \mathcal{S}$  for some locally Lipschitz class- $\mathcal{K}$  function  $\alpha$ . Then, using

the feedback control law (18), the system remains safe irrespective of the cost function.

*Proof.* The proof is provided in Appendix B in the full version of this paper [1]. ■

**Remark 5.** This formulation can be extended to more than 2 robots by introducing a new pair of dual variables for each pair of robots, and the corresponding constraints in (18). Note that the dual variable for robot  $i$  corresponding to robot  $j$  is different from that for robot  $k$ . Thus, the analysis in Sec. III remains valid. Additionally, static or dynamic obstacles can be represented as uncontrolled robots and non-convex shaped robots can be represented through unions of different polytopes with each using the same states and inputs.

#### IV. NUMERICAL EXAMPLES

In this section, we consider the problem of moving an L-shaped sofa through an L-shaped corridor. The approach presented in Sec. III is applied to solve the problem in real-time.

##### A. Simulation Setup

The sofa is the controlled object, with the side length of the L-shape as 1 m and the width as 0.1 m, as illustrated in Fig. 1. To implement our controller, we consider the sofa as the union of two  $1\text{ m} \times 0.1\text{ m}$  arms perpendicular to each other. The width of the corridor is 1 m.

The states of the sofa are  $x = (z_1, z_2, \theta) \in \mathbb{R}^2 \times \mathbb{R}$ , where  $(z_1, z_2)$  is the position of the vertex at the intersection of the two arms, and  $\theta$  is the angle of rotation. The inputs to the sofa are  $(v, \omega)$ , where  $v$  is the speed of the sofa and  $\omega$  the angular velocity. The velocity of the sofa is assumed to be along a  $\frac{\pi}{4}$  angle to the arm. The nonlinear control affine dynamics of the sofa is then,

$$\dot{z}_1 = v \cos(\theta + \frac{\pi}{4}), \quad \dot{z}_2 = v \sin(\theta + \frac{\pi}{4}), \quad \dot{\theta} = \omega. \quad (20)$$

Further, we impose input bounds as  $|v| \leq 0.3\text{ m/s}$  and  $|\omega| \leq 0.2\text{ rad/s}$ . The corridor is represented as the union of three rectangular walls, shown in Fig. 1. The polytopes corresponding to these walls are defined as follows:

$$\mathcal{W}^i := \{z \in \mathbb{R}^2 : A^{W^i} z \leq b^{W^i}\}, \quad i \in [3] \quad (21)$$

We note the general problem description in Sec. II allows for such static obstacles by eliminating the dependence of  $A^{W^i}$  and  $b^{W^i}$  on the state, as mentioned in Rem. 5.

Similarly, the sofa is represented as:

$$\mathcal{P}^j(x) := \{z \in \mathbb{R}^2 : A^{S^j}(x)z \leq b^{S^j}(x)\}, \quad j \in [2], \quad (22)$$

where  $\mathcal{P}^j(x)$  is denoted as Arm  $j$ . Arm 1 is the blue-colored polytope in Fig. 1, whereas Arm 2 is the green-colored one. Again, our description allows for this by choosing the same states and inputs for the two polytopes.

The NCBF is chosen as the square of the minimum distance as in (4). Notice that the QP-based program (18) will always have a solution, as  $v = 0, \omega = 0$  is always a feasible solution. Since we have static obstacles and the

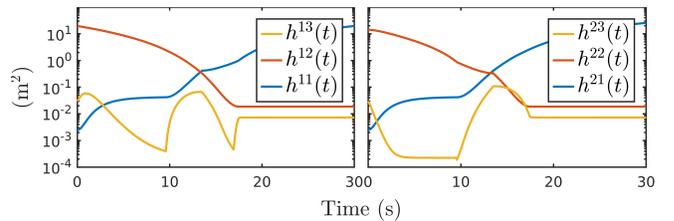


Fig. 3: Square of minimum distance (NCBF) between the arms of the sofa and the walls, where Wall  $W^1$  (left figure) is the left wall, Wall  $W^2$  (right figure) the upper one, and Wall  $W^3$  the inner wall of the corridor. Since the minimum distance is always greater than zero, the L-shaped sofa never collides with the obstacles over its trajectory. Both plots have log-scale on the y-axis.

controlled object consists of two polytopes, we only enforce NCBF constraints between  $\mathcal{W}^i$  and  $\mathcal{P}^j$  for  $i \in [3], j \in [2]$ .

A control Lyapunov function  $V_{clf}$  [2] is introduced in the final QP formulation in place of the nominal controller with the form as follows:

$$V_{clf}(x) = (z_1 - z_1^d)^2 + (z_2 - z_2^d)^2 + k(\theta - \theta^d)^2, \quad (23)$$

where  $\theta^d = -\frac{\pi}{4}$ . The desired position  $(z_1^d, z_2^d)$  is chosen to be at the end of the corridor, and the initial orientation is  $\frac{\pi}{4}$ . The CLF constraint  $\dot{V}_{clf} \geq -\alpha_1 V_{clf} - s$  is then enforced, where  $s \geq 0$  is a slack variable. The slack variable ensures that the feasibility of the QP-based program is not affected by the CLF constraint. The margin  $\epsilon_1$  as defined in Rem. 3 is chosen as 1.5 cm and  $\epsilon_2$  is chosen as  $10^{-5}$ .

##### B. Results

The simulations are performed on a Virtual Machine with 4 cores of a 2.20 GHz Intel Core i7 processor, running IPOPT [42] on MATLAB, and the visualization is generated by MPT3 [43]. The snapshots of the sofa trajectory is shown in Fig. 1 and also illustrated in the multimedia attachment.

1) *Enforcement of NCBF constraint:* The system remains safe throughout the simulation since the NCBF  $h^{ij}$  is greater than  $\epsilon_1^2$  for all possible robot-obstacle interactions as depicted in Fig. 3. In Fig. 4, the values of  $\dot{h}(t)$ , and LHS and RHS terms in the NCBF constraints from (18) are shown. We can see that by Lem. 5, the LHS term in the NCBF constraint is always a lower bound to  $\dot{h}(t)$ . This verifies the safety property of our duality-based approach.

TABLE I: Statistical analysis of computation time (ms) per iteration

Timing (ms)	mean $\pm$ std	p50	p99	max
Distance QPs (10)	1.07 $\pm$ 0.29	0.99	1.94	20.1
Polytope-NCBF-QP (18)	14.5 $\pm$ 1.55	14.2	19.3	37.6
Total ( $2 \times 3 + 1 = 7$ QPs)	21.1 $\pm$ 1.69	20.8	26.9	41.6

2) *Computation time:* From Table I, we can see that there are large outliers in computation time per iteration, but they occur in less than 0.1% iterations. Nevertheless, from Table I, we can apply our controller at 50Hz. The fast computation time allows us to directly implement the control inputs from (18) on the robot in real-time.

The optimization problem for the sofa problem has 51 variables and at most 72 constraints. In general consider

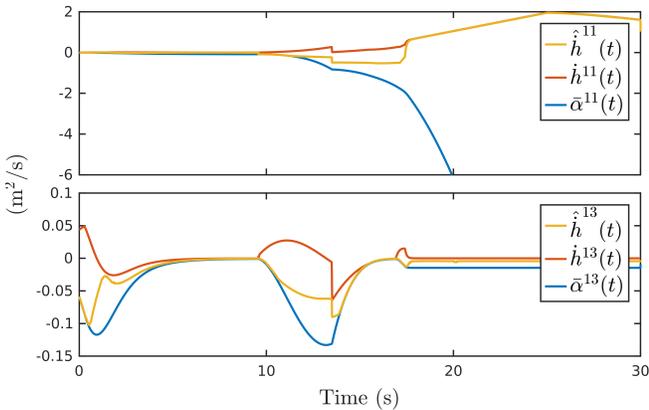


Fig. 4: NCBF constraint enforcement between Arm 1 and Wall 1 (top figure) and Arm 1 and Wall 3 (bottom figure). Illustration of safety performance of the system. The red lines are  $\hat{h}(t)$ , the yellow lines are the lower bounds of  $\hat{h}(t)$ , and blue line is the RHS of the NCBF constraint.  $\bar{\alpha}^{1j}(t) = -\alpha(h^{1j}(t) - \epsilon_1^2)$ , where  $j$  is the wall. The system safety is guaranteed with the red line being always above the blue line.

$N$  controlled robots in a  $d$ -dimensional space with  $f$  facets and  $m$  control inputs. Then, the polytope-NCBF-QP (18) has  $(1+d)\frac{N(N-1)}{2}$  constraints along with at most  $2f\frac{N(N-1)}{2}$  non-negativity constraints and  $N(m+2f)$  variables, whereas a CBF-QP formulation using spherical over-approximation would have  $\frac{N(N-1)}{2}$  constraints and  $Nm$  variables.

3) *Continuity of  $\hat{h}(t)$  and  $\lambda^*(t)$* : Fig. 4 also shows that both  $\hat{h}(t)$  and the lower bound of  $\hat{h}(t)$  can have discontinuities. For the case of the moving sofa problem, a discontinuity in  $\hat{h}(t)$  can arise when the sofa is rotating and the point of minimum distance on the sofa with any wall jumps from one vertex to another, since the end points of the sofa arm need not have the same velocity when rotating. The dual optimal variables  $\lambda^{*i}(t)$  and  $\lambda^{*j}(t)$  are always continuous and right-differentiable as shown in Sec. III. The dual variables are plotted in Fig. 5, which demonstrate this property.

4) *Deadlocks*: For various initial conditions, the sofa can get stuck in a deadlock in the corridor. This can happen when the arms of the sofa are so large that it cannot turn at the corner. It can also happen when the two arms of the sofa get too close to the wall and the sofa cannot turn because it would cause one of its arms to penetrate the wall. Our controller still ensures safety in this case. A high-level planner could help by generating a deadlock free trajectory at low frequency that then serves as input to our control law.

### C. Discussions

1) *Nonlinearity of the system dynamics*: The duality-based formulation (18) is a convex quadratic program even when the system dynamics is nonlinear, as long as it is control affine. This allows us to achieve dynamically-feasible obstacle avoidance with QPs for polytopes.

2) *Optimality of solution vs computation time*: As noted in Sec. III, the cost function of the QP-based program (18) does not affect the safety of the system. If the optimization solver does not converge to the optimal solution, the current solution can be used if it is feasible. This can be useful

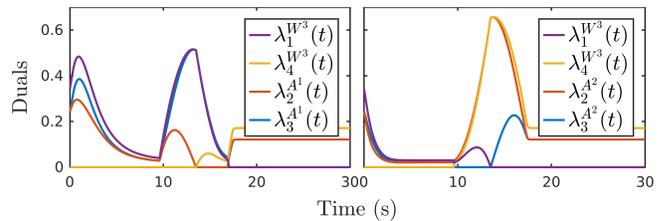


Fig. 5: Dual optimal solutions for Arm 1 (left figure) and Arm 2 (right figure) of the robot corresponding to the obstacle wall 3. The dual optimal solutions are unique, continuous, and right-differentiable as shown in Sec. III. Note that pair of dual variables for Arm 1-Wall 3 are different from that of Arm 2-Wall 3, i.e. the vector  $\lambda^{W^3}(t)$  is different for Arms 1 and 2. The subscripts represent the components of the vectors, and only the non-zero components have been plotted.

in real-time implementations where both control frequency and safety matter. A feasible solution to (18) can be directly found in some cases, such as when the NCBF is constructed using a safe backup controller [44].

3) *Trade-off between computation speed and tight maneuvering*: A polytope with  $f^i$  faces would require  $f^i$  dual variables in the dual formulation (18) and additional constraints. Using a hyper-sphere as an over-approximation to the polytope would require fewer dual variables, as in [4]. However, such an approximation can be too conservative and completely ignore the rotational geometry of the polytope. Using the full polytope structure can prove beneficial in dense environments, such as the sofa problem, where a spherical approximation cannot work. So, there is a trade-off between computation speed and maneuverability. In practice, a hybrid approach should be used: a hyper-sphere approximation when two obstacles are far away, and the polytope structure when closer, which could find a good trade-off between computation speed and maneuverability for tight obstacle avoidance in dense environments.

4) *Robustness with respect to the safe set*: Since we use the minimum distance between two polytopes as the NCBF and not the signed distance, the minimum distance is uniformly zero when two polytopes intersect. So, the proposed controller is not robust in the sense that if the state leaves the safe set, it will not converge back to it.

## V. CONCLUSION

In this paper, we have presented a general framework for obstacle avoidance between polytopes using the dual program of a minimum distance QP problem. We have shown that the control input using our method can be computed using a QP for systems with control affine dynamics, enabling real-time implementation. We have numerically verified the safety performance of our controller for the problem of moving an L-shaped sofa through a tight corridor. We also have explored properties such as robustness and deadlock avoidance. Further work in this topic would involve extending our method to other convex shapes, and deriving safety guarantees for more general classes of systems.

## ACKNOWLEDGEMENT

The authors gratefully acknowledge Somayeh Sojoudi at the University of California, Berkeley for her valuable comments on properties of parametric optimization problems.

## REFERENCES

- [1] A. Thirugnanam, J. Zeng, and K. Sreenath, "Duality-based convex optimization for real-time obstacle avoidance between polytopes with control barrier functions," *arXiv preprint arXiv:2107.08360*, 2021.
- [2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, 2019, pp. 3420–3431.
- [3] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016, pp. 322–328.
- [4] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *IEEE Conference on Decision and Control*, 2019, pp. 474–479.
- [5] 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*, 2017.
- [6] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *IEEE Conference on Decision and Control*, 2021, pp. 6137–6144.
- [7] H. Almubarak, K. Stachowicz, N. Sadegh, and E. Theodorou, "Safety embedded differential dynamic programming using discrete barrier states," *IEEE Robotics and Automation Letters*, 2022.
- [8] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed pointwise feasibility," in *2021 American Control Conference*, 2021, pp. 3856–3863.
- [9] D. R. Agrawal and D. Panagou, "Safe control synthesis via input constrained control barrier functions," in *IEEE Conference on Decision and Control*, 2021, pp. 6113–6118.
- [10] A. Katriniok, "Control-sharing control barrier functions for intersection automation under input constraints," *arXiv preprint arXiv:2111.10205*, 2021.
- [11] J. Breeden and D. Panagou, "High relative degree control barrier functions under input constraints," in *IEEE Conference on Decision and Control*, 2021, pp. 6119–6124.
- [12] M. Krstic and M. Bement, "Nonovershooting control of strict-feedback nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1938–1943, 2006.
- [13] M. Yue, X. Wu, L. Guo, and J. Gao, "Quintic polynomial-based obstacle avoidance trajectory planning and tracking control framework for tractor-trailer system," *International Journal of Control, Automation and Systems*, vol. 17, no. 10, pp. 2634–2646, 2019.
- [14] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 7139–7145.
- [15] Y. Huang and Y. Chen, "Switched control barrier function with applications to vehicle safety control," in *ASME Dynamic Systems and Control Conference*, 2020.
- [16] M. Marley, R. Skjetne, and A. R. Teel, "Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles," in *2021 American Control Conference*, 2021, pp. 243–249.
- [17] S. He, J. Zeng, B. Zhang, and K. Sreenath, "Rule-based safety-critical control design using control barrier functions with application to autonomous lane change," in *2021 American Control Conference*, 2021, pp. 178–185.
- [18] M. Marley, R. Skjetne, E. Basso, and A. R. Teel, "Maneuvering with safety guarantees using control barrier functions," in *13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2021*, vol. 54, no. 16, 2021, pp. 370–377.
- [19] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 194–206, 2017.
- [20] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *American Control Conference*, 2016, pp. 2252–2258.
- [21] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conference*, 2015, pp. 4542–4548.
- [22] F. Ferraguti, M. Bertuletti, C. T. Landi, M. Bonfè, C. Fantuzzi, and C. Secchi, "A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5921–5928, 2020.
- [23] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference*, 2021, pp. 3882–3889.
- [24] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, 1988.
- [25] F. B. Adda and J. Cresson, "About non-differentiable functions," *J. Math. Anal. Appl.*, vol. 263, no. 2, pp. 721–737, 2001.
- [26] 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.
- [27] —, "Boolean composability of constraints and control synthesis for multi-robot systems via nonsmooth control barrier functions," in *IEEE Conf. on Control Technology and Applications*, 2018, pp. 897–902.
- [28] B. Li and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowledge-Based Systems*, vol. 86, pp. 11–20, 2015.
- [29] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and engineering*, vol. 3, no. 3, pp. 227–252, 2002.
- [30] J. Schulman, Y. Duan, J. Ho, A. 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.
- [31] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, 2020.
- [32] J. Zeng, P. Kotaru, M. W. Mueller, and K. Sreenath, "Differential flatness based path planning with direct collocation on hybrid modes for a quadrotor with a cable-suspended payload," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3074–3081, 2020.
- [33] X. Shen, E. L. Zhu, Y. R. Stürz, and F. Borrelli, "Collision avoidance in tightly-constrained environments without coordination: a hierarchical control approach," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 2674–2680.
- [34] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, and K. Sreenath, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering*, 2021, pp. 2132–2139.
- [35] R. Firoozi, X. Zhang, and F. Borrelli, "Formation and reconfiguration of tight multi-lane platoons," *Control Engineering Practice*, vol. 108, p. 104714, 2021.
- [36] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *IEEE International Conference on Robotics and Automation*, 2022.
- [37] W. E. Howden, "The sofa problem," *The computer journal*, vol. 11, no. 3, pp. 299–301, 1968.
- [38] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.
- [39] M. J. Best and N. Chakravarti, "Stability of linearly constrained convex quadratic programs," *Journal of Optimization Theory and Applications*, vol. 64, no. 1, pp. 43–53, 1990.
- [40] D. Klatté and B. Kummer, "Stability properties of infima and optimal solutions of parametric optimization problems," in *Nondifferentiable Optim.: Motivations and Applications*. Springer, 1985, pp. 215–229.
- [41] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [42] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [43] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510.
- [44] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *IEEE Conf. Control Technol. Appl.*, 2018, pp. 1656–1661.