Safe Control of Second-Order Systems with Linear Positional Constraints

Mohammed Alyaseen   Nikolay Atanasov   Jorge Cortés

***Abstract*—Control barrier functions (CBFs) offer a powerful tool for enforcing safety specifications in control synthesis. This paper deals with the problem of constructing valid CBFs. Given a second-order system and any desired safety set with linear boundaries in the position space, we construct a provably control-invariant subset of this desired safety set. The constructed subset does not sacrifice any positions allowed by the desired safety set, which can be nonconvex. We show how our construction can also meet safety specification on the velocity. We then demonstrate that if the system satisfies standard Euler-Lagrange systems properties then our construction can also handle constraints on the allowable control inputs. We finally show the efficacy of the proposed method in a numerical example of keeping a 2D robot arm safe from collision.**

***Index Terms*—safety critical control, nonsmooth control barrier functions, optimization-based control design.**

I. Introduction

Control barrier functions (CBFs) provide a flexible framework to certify the forward invariance of a desired set with respect to system trajectories and design feedback controllers that ensure it. Because of their versatility, CBFs have made their way into numerous applications in robotics, transportation, power systems, and beyond. By definition, every boundary point of the CBF's 0-superlevel set admits a control value that holds the system's trajectory from instantaneously leaving it. This point-wise condition is known as the CBF condition. Finding CBFs is a challeging task: it amounts to finding a set whose states can be made safe, i.e., for which control actions ensuring safety can be identified. This is not trivial given the complexity of the dynamics and limitations on the control inputs. After clearing this challenge, one must still figure out whether a well-behaved control law can be synthesized out of all the point-wise safe control actions. In this work, we construct valid CBFs that enforce any positional safety requirements with linear boundaries for second-order dynamics and provide an associated continuous safe controller.

Literature review: Whether from Nagumo's theorem [1] or from comparison results in the theory of differential equations [2], the CBF condition was first derived for smooth functions [3]–[5]. This condition was extended to non-smooth functions in multiple works, such as [6]–[8]. Many approaches have been proposed to construct a CBF or verify whether a given function is a CBF. One approach applies learning methods to construct CBFs [9]–[12]. Another uses reachability analysis to construct the maximal invariant set and use it in safe control design [13]. Another class uses backstepping to design CBFs for cascaded systems [14]. Still, another group of works, most closely related to the treatment here, utilizes properties of specific systems to construct suitable CBFs. For instance, [15]–[17] constructs CBFs for polynomial systems using sum-of-square optimization. The work [18] constructs non-smooth CBFs for fully actuated Euler-Lagrange systems, with constraints on position, velocity, and inputs given by hypercubes. The work [19] proposes a method to construct a safe subset of a hyper-sphere in the position space, assuming no input constraints. Both these works consider convex constraint sets. All these approaches have advantages and disadvantages: learning methods are general but approximate; reachability analysis is general and exact but computationally intractable for high dimensions; backstepping is tractable and exact, but limited to cascaded systems. Finally, the works that consider specific classes of systems are exact and tractable, but limited to those classes and their respective safe set structures. Our work here contributes to the last line of work by significantly enlarging the class of sets that can be rendered safe with second-order systems. We render safe a highly expressive class of positional (potentially nonconvex) safety constraints defined by linear boundaries rather than hypercubes and hyperspheres as in [18], [19]. Once a valid CBF is constructed, safe feedback controllers are usually synthesized via state-parameterized optimization programs [4], [20] due to their flexibility, convenience, and computational lightness. This motivated the study of the regularity properties of such controllers, see e.g. [21]–[23]. Our recent work [24] synthesizes a provably feasible optimization-based safe feedback controller for safe sets given by arbitrarily nested unions and intersections of superlevel sets of differentiable functions. By constructing a valid safe set, we show the applicability of the general control techniques in [24] to the class of systems considered here.

Statement of Contributions: We consider[1] second-order system dynamics and positional constraints specified by nested unions and intersections of half-spaces. We construct a control-invariant subset of the full state space which contains all positions allowed by the original positional constraints. We derive a general condition which, if satisfied, proves that our constructed set is safe for general, possibly not fully actuated, second-order systems and provide an associated QP safeguarding controller. We then show that this safety condition is always satisfied for fully actuated systems. We further show that a compact allowable controls set suffices for

M. Alyaseen, N. Atanasov, and J. Cortés are with the Contextual Robotics Institute, UC San Diego, {malyasee,natanasov,cortes}@ucsd.edu. M. Alyaseen is also affiliated with Kuwait University as a holder of a scholarship. This work was supported by AFOSR Award FA9550-23-1-0740.

[1]We let $\mathbf{1_{n\times m}}$ and $\mathbf{0_{n\times m}}$ denote the $n\times m$ matrices of ones and zeroes, resp. Likewise, $\mathbf{1_n}$ and $\mathbf{0_n}$ denote the n vectors of ones and zeroes, resp. The boundary, interior, and convex hull of $\mathcal{S}$ are $\partial\mathcal{S}$, $\mathrm{int}(\mathcal{S})$, and $\mathrm{co}(\mathcal{S})$, resp. The projection of $\mathcal{C}\subset\mathbb{R}^n$ on the first n' components is denoted $\mathrm{Proj}_{n'}(\mathcal{C})$. The vectors of the standard basis of $\mathbb{R}^n$ are denoted $\{e_\ell\}_{\ell=1}^n$. We denote the 2-norm of a vector x by $\|x\|$. The norm of a matrix A is the induced 2-norm, $\|A\| := \sup_{x\neq 0}\frac{\|Ax\|}{\|x\|}$. We refer to the 0-superlevel set of a function simply as superlevel set. Table I provides a summary of key notation used throughout the technical exposition.

safety when the desired safe set is bounded. We show how our method can be utilized to incorporate velocity and input constraints when the dynamics satisfy standard Euler-Lagrange system properties. Finally, we apply our method to design safe controls for a 2D robotic arm.

$\mathcal{C}, \mathcal{L}, \mathcal{I}^\ell$	Desired safe set and its sets of indices, cf. (3)
$\mathcal{C}^s, \bar{\mathcal{L}}, \bar{\mathcal{I}}^\ell$	Safe set and its sets of indices, cf. (5)
$\mathcal{I}_{\text{act}}(x)$	Active constraints at boundary point x, cf. Condition 1
ϵ, γ	Design parameters for safe set, cf. (6)

TABLE I: Summary of key notation.

II. Problem Statement

Consider the second-order dynamics

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

where $x = (x_1, x_2)$, $x_1, x_2 \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f(x) = (x_2, f_2(x))$ is the drift, and $G(x) = (\mathbf{0}_{n\times m}, G_2(x))$ correspond to the input vector fields. Here, $f_2 : \mathbb{R}^{2n} \to \mathbb{R}^n$ and $G_2 : \mathbb{R}^{2n} \to \mathbb{R}^{n\times m}$ are Lipschitz functions. Consider also the half-spaces parameterized by $i \in \mathcal{I} := \{1, \ldots, r\}$

$$\mathcal{C}_i := \{x \in \mathbb{R}^{2n} \mid h_i(x) := a_i^\top x_1 + b_i \geq 0\}, \tag{2}$$

with $\|a_i\| \neq 0$. We require that if $i \neq i'$ then the augmented vectors (a_i, b_i) and $(a_{i'}, b_{i'})$ are linearly independent. The states x_1 and x_2 can represent mechanical, electrical or other quantities: we refer to x_1 and x_2 as generalized position and generalized velocity, resp., for convenience. Our goal is to design a control law for (1) that keeps invariant a desired set $\mathcal{C}$ given as a union of intersections of the half-spaces $\mathcal{C}_i$'s,

$$\mathcal{C} = \bigcup_{\ell\in\mathcal{L}} \bigcap_{i\in\mathcal{I}^\ell} \mathcal{C}_i, \tag{3}$$

where $\mathcal{L} \subset \mathbb{N}$ and the sets $\mathcal{I}^\ell \subset \mathcal{I}$ are sets of indices. This set corresponds to the superlevel set of the function

$$h(x) = \max_{\ell\in\mathcal{L}} \min_{i\in\mathcal{I}^\ell} h_i(x). \tag{4}$$

Note that the safety set $\mathcal{C}$ constrains only the states x_1 corresponding to the generalized position. This setup is common in many problems, such as collision avoidance [25], where the main concern is to avoid the physical locations occupied by obstacles. This form for $\mathcal{C}$ is flexible enough to capture the safety requirement of staying in any set of positions with linear boundaries while simultaneously avoiding obstacles of linear boundaries. An example of such a set is shown later in Figure 1c. In [24, Lemma 4.8], we provide a procedure of turning any set given by arbitrary nested unions and intersections of $\mathcal{C}_i$'s into the form (3). This is applicable in a wide range of situations for autonomous robotic systems from geofencing to protect a human collaborator to avoiding collisions in human environments.

We impose the following structural assumptions on the safety requirement. Let $\mathcal{S}_\cap$ be the collection of sets of indices whose corresponding half-spaces intersect in $\mathcal{C}$: that is, if $I \subseteq \mathcal{I}$ is such that $(\cap_{i\in I}\mathcal{C}_i) \cap \mathcal{C} \neq \emptyset$, then $I \in \mathcal{S}_\cap$.

Assumption 1. The set $\text{Proj}_n(\mathcal{C})$ is compact. Furthermore, for any $I \in \mathcal{S}_\cap$, one point $y_I \in \mathcal{C}$ can be chosen to satisfy $h_i(y_I) > 0$ for all $i \in I$. •

Assumption 1 is reasonable. Assuming the compactness of $\text{Proj}_n(\mathcal{C})$ is the same as saying that the set of safe positions x_1 is compact. A sufficient condition for this is the boundedness of $\cap_{i\in\mathcal{I}^\ell}\mathcal{C}_i$ for every ℓ. The last part of the assumption only requires that any nonempty intersection $(\cap_{i\in I}\mathcal{C}_i) \cap \mathcal{C}$ has a nonempty interior. For instance, in our example later in Section V, the safety constraints $x_1 \geq a$ and $x_1 \leq b$ with some $a < b$, satisfy Assumption 1 since the interior of their intersection in the totality of $\mathcal{C}$ is non-empty. We discuss the impact of the specific choice of points $\{y_I\}$ in Remark III.2.

Since our system (1) is second order, the control is only available in the order of the generalized acceleration $\dot{x}_2$. Thus, $\mathcal{C}$, which only constrains x_1, is the superlevel set of a function that does not satisfy the CBF condition [4]. Therefore, $\mathcal{C}$ is generally not control-invariant due to the lack of constraints on the velocity: e.g., initial conditions starting exactly at the boundary of $\mathcal{C}$ with a velocity heading outwards are unsafe, with no possible control value to counter it. Hence, there is a need to identify a control-invariant set $\mathcal{C}^s \subseteq \mathcal{C}$ that constrains the velocity. This construction should contain as much of $\text{Proj}_n(\mathcal{C})$ as possible.

Problem 1. Given the dynamics (1) and $\mathcal{C}$ defined by (3) satisfying Assumption 1, construct $\mathcal{C}^s$ such that:

(i) $\mathcal{C}^s \subseteq \mathcal{C}$,
(ii) $\text{Proj}_n(\mathcal{C}^s) = \text{Proj}_n(\mathcal{C})$,
(iii) $\mathcal{C}^s$ is control-invariant,

and design a continuous controller that renders $\mathcal{C}^s$ invariant.

III. Construction of Control-Invariant Set

In this section, we solve Problem 1 by constructing a function B of the form $B(x) = \max_{\ell\in\bar{\mathcal{L}}} \min_{i\in\bar{\mathcal{I}}^\ell} B_i(x)$, and then proving that its superlevel set

$$\mathcal{C}^s := \{x \in \mathbb{R}^{2n} \mid B(x) \geq 0\} = \bigcup_{\ell\in\bar{\mathcal{L}}} \bigcap_{i\in\bar{\mathcal{I}}^\ell} \mathcal{C}_i^s, \tag{5}$$

satisfies the requirements in Problem 1, where $\mathcal{C}_i^s$'s are the superlevel sets of B_i's. Constructing B amounts to defining the functions B_i and the index sets $\bar{\mathcal{L}}$ and $\bar{\mathcal{I}}^\ell$.

Definition of functions: For each $1 \leq i \leq r$, define $B_i(x) = h_i(x)$ and

$$B_{i+r}(x) = a_i^\top x_2 + \gamma(a_i^\top x_1 + b_i) - \epsilon, \tag{6}$$

where ϵ and γ are two positive design parameters, each with a special role in customizing the design and proving safety.

Definition of sets: Let $\bar{\mathcal{L}} = \mathcal{L}$ and $\bar{\mathcal{I}}^\ell = \mathcal{I}^\ell \cup (\{r\} + \mathcal{I}^\ell)$.

The choice of B follows this logic: for $i \in \{1, \ldots, r\}$, $\mathcal{C}_i$ is the superlevel set of B_i; the function B_{i+r} is then chosen such that its superlevel set contains the points on the boundary of $\mathcal{C}_i$ only if f points towards the interior of the safe set at those boundary points. Thus, the system is safe without requiring any inputs at the boundary points defined by $B_i = 0$ with $i \in \{1, \ldots, r\}$ if these boundary points are in the superlevel set of B_{i+r}. This choice of B_{i+r} is inspired by the concept of high-order control barrier function [26].

The following result shows that this construction satisfies requirements (i) and (ii) in Problem 1.

Lemma III.1 (No Positions Lost). *Let $\mathcal{C}^s$ be the superlevel set of B. Then $\mathcal{C}^s \subset \mathcal{C}$. Under Assumption 1, define*

$$\delta := \min_{I \in \mathcal{S}_\cap,\ i \in I} h_i(y_I) > 0, \tag{7}$$

and let γ, ϵ with $\gamma\delta > \epsilon$. Then, $\mathrm{Proj}_n(\mathcal{C}^s) = \mathrm{Proj}_n(\mathcal{C})$. •

Proof. That $\mathcal{C}^s$ is a subset of $\mathcal{C}$ follows directly from the definition of these sets and the fact that $\mathcal{C}_i^s = \mathcal{C}_i$ for $i \in \{1, \dots, r\}$. This also implies that $\mathrm{Proj}_n(\mathcal{C}^s) \subset \mathrm{Proj}_n(\mathcal{C})$. Note that Assumption 1 ensures that $\delta > 0$. Since by assumption $\gamma\delta > \epsilon$, there exists $0 < \sigma < 1$ such that $\gamma\sigma\delta > \epsilon$. Given $(x_1, x_2) \in \mathcal{C} = \cup_{\ell \in \mathcal{L}} \cap_{i \in \mathcal{I}^\ell} \mathcal{C}_i$, there is an $\ell' \in \mathcal{L}$ such that $(x_1, x_2) \in \mathcal{C}_i$, for all $i \in \mathcal{I}^{\ell'}$, i.e., $h_i(x) = B_i(x) \geq 0$, for all $i \in \mathcal{I}^{\ell'}$. By Assumption 1, there exists $y_{\mathcal{I}^{\ell'}} \in \mathcal{C}$ satisfying $h_i(y_{\mathcal{I}^{\ell'}}) > 0$ for all $i \in \mathcal{I}^{\ell'}$. The choice $x' = (x_1, x_2')$, where $x_2' = -\gamma\sigma(x_1 - y^n_{\mathcal{I}^{\ell'}})$, where $y^n_{\mathcal{I}^{\ell'}}$ denotes the first n components of $y_{\mathcal{I}^{\ell'}}$, gives

$$\begin{aligned} B_{i+r}(x') &= a_i^\top x_2' + \gamma B_i(x) - \epsilon \\ &= \gamma(1-\sigma) B_i(x) + \gamma\sigma B_i(y_{\mathcal{I}^{\ell'}}) - \epsilon \\ &\geq \gamma(1-\sigma) B_i(x) + \gamma\sigma\delta - \epsilon \\ &> \gamma(1-\sigma) B_i(x) \geq 0. \end{aligned}$$

Therefore, $B_i(x') \geq 0$ for all $i \in \bar{\mathcal{I}}^{\ell'}$, implying that $x' \in \cap_{i \in \bar{\mathcal{I}}^{\ell'}} \mathcal{C}_i^s \subseteq \mathcal{C}^s$. Therefore, $\mathrm{Proj}_n(\mathcal{C}) \subset \mathrm{Proj}_n(\mathcal{C}^s)$. ☐

Remark III.2 (Maximizing Flexibility of Safe Set Design)**.** Lemma III.1 requires that the parameters γ and ϵ satisfy $\gamma\delta > \epsilon$. Since δ is dependent on the choice of points $\{y_I\}$, whose existence is assumed in Assumption 1, the choice of γ and ϵ is dependent on $\{y_I\}$. The greater δ, the more flexibility for choosing γ and ϵ. Note, however, that the specific choice of $\{y_I\}$ is not crucial for the results: any choice makes the condition $\gamma\delta > \epsilon$ satisfiable. •

To address requirement (iii) in Problem 1, we need to identify a control action at each state of $\mathcal{C}^s$ that makes $\mathcal{C}^s$ forward-invariant. We introduce the following condition.

Condition 1 (General Safety Condition)**.** For each $x \in \partial\mathcal{C}^s$, there exists $u_x \in \mathcal{U}$ such that

$$\dot{B}_i(x) = \nabla B_i(x)^\top (f(x) + G(x) u_x) > 0$$

for all $i \in \mathcal{I}_{\text{act}}(x) := \{i \in \{1, \dots, 2r\} \mid \exists \ell \in \mathcal{L}_i : \ B(x) = B^\ell(x) = B_i(x)\}$, with $B^\ell(x) = \min_{i \in \bar{\mathcal{I}}^\ell} B_i(x)$ and $\mathcal{L}_i := \{\ell \in \bar{\mathcal{L}} \mid i \in \bar{\mathcal{I}}^\ell\}$. •

Condition 1 requires that there exists a control input that steers the system to the interior of the safe set at all boundary points where the drift of the system is not guaranteed to do that on its own. Consider now the feedback controller u^* defined by the following quadratic program:

$$\begin{aligned} &(u^*(x), \alpha^*(x), M^*(x)) := \\ &\arg\min_{\alpha, M, u \in \mathcal{U}} u^\top Q(x) u + q(x)^\top u + q_\alpha \alpha^2 + q_M M^2 \\ &\text{s.t. } M \geq c_M, \ \alpha \geq c_\alpha \\ &\quad \nabla B_i(x)^\top (f(x) + G(x) u) + \alpha B_i(x) \\ &\quad + M\left(B(x) - B^\ell(x)\right) \geq 0, \ \forall \ell \in \bar{\mathcal{L}}, \ \forall i \in \bar{\mathcal{I}}^\ell. \end{aligned} \tag{8}$$

Here, $Q : \mathcal{X} \to \mathbb{R}^{m \times m}$ is a Lipschitz function which takes values in the set of positive-definite matrices, $q : \mathcal{X} \to \mathbb{R}^m$ is a Lipschitz function, and q_α, q_M, c_M, c_α are positive design constants. The following result summarizes the properties of u^* and, in particular, that it makes $\mathcal{C}^s$ control-invariant. under Condition 1.

Theorem III.3 (Safe Controller [24, Thm. 4.12])**.** *Let $\mathcal{C}^s$ be compact. Under Condition 1, there exists a neighborhood of $\mathcal{C}^s$ where program (8) is feasible and u^* is single-valued, continuous, and renders $\mathcal{C}^s$ control-invariant (i.e., any solution of (1) with $u = u^*$ starting in $\mathcal{C}^s$ stays in $\mathcal{C}^s$).* •

Our approach to establish (iii) in Problem 1 is then to verify the hypotheses of Theorem III.3. Keep in mind that the original set of safety requirements, $\mathcal{C}$, does not satisfy the hypotheses. This is why the construction of $\mathcal{C}^s$ is required so that the control techniques developed in [24] can be applied. The next result shows that $\mathcal{C}^s$ is compact if $\mathrm{Proj}_n(\mathcal{C})$ is compact, which is readily ensured by Assumption 1.

Proposition III.4 (Compactness of Safe Set)**.** *Under Assumption 1, $\mathcal{C}^s$ is compact.* •

Proof. We reason by contradiction. Suppose $\mathcal{C}^s = \cup_{\ell \in \bar{\mathcal{L}}} \cap_{i \in \bar{\mathcal{I}}^\ell} \mathcal{C}_i^s$ is not compact. Then, for some $\ell' \in \mathcal{L}$, the convex closed set $\cap_{i \in \bar{\mathcal{I}}^{\ell'}} \mathcal{C}_i^s$ is unbounded and, thus, contains a ray [27, Result 2.5.1]. That is there is a point $x_0 = (x_{0,1}, x_{0,2})$ and a direction $\zeta = (\zeta_1, \zeta_2) \neq \mathbf{0}$ such that $B_i(x_0 + t\zeta) \geq 0$ for all $t \geq 0$ and all $i \in \bar{\mathcal{I}}^{\ell'}$. We distinguish two cases: (a) $\zeta_1 \neq \mathbf{0}$ and (b) $\zeta_1 = \mathbf{0}$. In case (a), $B_i(x_0 + t\zeta) = h_i(x_0 + t\zeta) \geq 0$ for all $i \in \mathcal{I}^{\ell'}$, so $\mathrm{Proj}_n(\mathcal{C})$ is not compact, which contradicts Assumption 1. In case (b), $\zeta_1 = \mathbf{0} \neq \zeta_2$. So, $B_{i+r}(x_0 + t\zeta) = t a_i^\top \zeta_2 + a_i^\top x_{0,2} + \gamma(a_i^\top x_{0,1} + b_i) - \epsilon \geq 0$, for all $i \in \mathcal{I}^{\ell'}$ and all $t \geq 0$. Since all terms in the inequality are constants except for the first one, we deduce that $t a_i^\top \zeta_2 \geq 0$ for t large enough, which implies $a_i^\top \zeta_2 \geq 0$, for all $i \in \mathcal{I}^{\ell'}$. Therefore, $h_i(x_0 + t(\zeta_2, \mathbf{0})) = t a_i^\top \zeta_2 + a_i^\top x_{0,1} + b_i = t a_i^\top \zeta_2 + h_i(x_0) \geq 0$ for all $t \geq 0$ and all $i \in \mathcal{I}^{\ell'}$, which implies that $\mathrm{Proj}_n(\mathcal{C})$ is unbounded, again contradicting Assumption 1. ☐

Next, we focus on the satisfaction of Condition 1. The following result particularizes this condition to our context.

Lemma III.5 (General Invariance Condition)**.** *Condition 1 is satisfied if, for all $x \in \partial\mathcal{C}^s$ and $i' + r \in \mathcal{I}_{act}(x)$ for which $B_{i'+r}(x) = 0$, there exists $u_x \in \mathcal{U}$ such that*

$$a_{i'}^\top (\gamma x_2 + f_2(x) + G_2(x) u_x) > 0. \tag{9}$$

Proof. If $x \in \partial\mathcal{C}^s$, then $B(x) = 0$ and thus $B_i(x) = 0$ for $i \in \mathcal{I}_{\text{act}}(x)$. For any such $i \in \mathcal{I}_{\text{act}}(x)$, it follows that $B_{i'}(x) \geq 0$ for any $\ell \in \mathcal{L}_i$ and $i' \in \bar{\mathcal{I}}^\ell$. Let us now verify the inequality of Condition 1 for $i \in \mathcal{I}_{\text{act}}(x)$. We distinguish two cases: (a) $i \leq r$ or (b) $i > r$. In case (a), $B_i(x) = a_i^\top x_1 + b_i = 0$ and $i' = i + r \in \bar{\mathcal{I}}^\ell$. Thus,

$$0 \leq B_{i'}(x) = a_i^\top x_2 + \gamma B_i(x) - \epsilon = a_i^\top x_2 - \epsilon. \tag{10}$$

So $\dot{B}_i(x) = a_i^\top x_2 \geq \epsilon > 0$ by (10). In case (b), for $i' = i - r$, $\dot{B}_i(x) = \dot{B}_{i'+r}(x)$ equals the left-hand side of (9), which verifies Condition 1 by assumption. ☐

The safety condition, inequality (9), can fail to hold in two ways. The first is when there is no $u_x \in \mathbb{R}^m$ that satisfies it. The second is when such a $u_x \in \mathbb{R}^m$ exists, but it does not belong to the allowable input set $\mathcal{U}$. The distinction between the two possibilities is useful since they can be overcome by different tools. A practical solution to the second failure type might be to expand $\mathcal{U}$ (e.g., by employing a more powerful actuator). In the first failure type, however, the set $\mathcal{C}^s$ cannot be made safe given the dynamics, and a better design must be sought. We exploit this distinction in what follows.

The inequality (9) is sufficient to establish invariance of general second-order systems. It is however difficult to verify for under-actuated systems. Instead, fully actuated systems are relatively easier to deal with, as shown next.

Proposition III.6 (Control-Invariance with Full Actuation)**.** *Assume that, for all $x \in \partial\mathcal{C}^s$, $G_2(x)$ is right invertible and $\mathcal{U} = \mathbb{R}^m$. Then, under Assumption 1, choosing γ and ϵ such that $\gamma\delta > \epsilon$ ensures that Condition 1 is satisfied.* •

Proof. Given $x = (x_1, x_2) \in \partial\mathcal{C}^s$, $B(x) = 0$. Define $I_x := \{i \in \{1, \dots, r\} \mid i + r \in \mathcal{I}_{\text{act}}(x), B_{i+r}(x) = 0\}$. If I_x is empty then the premise of Lemma III.5 is satisfied by default. Otherwise, by the definitions of $\mathcal{I}_{\text{act}}(x)$ and $\bar{\mathcal{I}}^\ell$'s, $B_i(x) \geq 0$ for all $i \in I_x$. Therefore, by Assumption 1, there exists $y_{I_x} \in \mathcal{C}$ satisfying $h_i(y_{I_x}) > 0$ for all $i \in I_x$. Then, for all $i \in I_x$, $B_{i+r}(x) = a_i^\top x_2 + \gamma(a_i^\top x_1 + b_i) - \epsilon = 0$, which implies $a_i^\top(-x_2 - \gamma x_1 + \gamma y_{I_x}^n) = \gamma(a_i^\top y_{I_x}^n + b_i) - \epsilon \geq \gamma\delta - \epsilon > 0$, where $y_{I_x}^n$ denotes the first n components of y_{I_x}. Because of this inequality, one can choose $u_x = \rho G_2^\dagger(x) y_x$, where $G_2^\dagger$ denotes the right inverse of G_2 and $y_x = -x_2 - \gamma x_1 + \gamma y_{I_x}^n$, with ρ large enough, so that the inequality (9) is satisfied. The result then follows from Lemma III.5. □

The combination of Propositions III.4 and III.6 allows us to invoke Theorem III.3 to establish that u^*, as defined in (8), renders $\mathcal{C}^s \subseteq \mathcal{C}$ forward-invariant. This, together with Lemma III.1, means that $\mathcal{C}^s$ solves Problem 1 for the case of full actuation. The assumption of full actuation is not uncommon in the control of second-order systems, whether when studying safety [18], [19] or stability [28]. The following result shows that the result of Proposition III.6 still holds for sufficiently large compact input sets $\mathcal{U}$.

Corollary III.7 (Compact Input Set Suffices for Control-Invariance)**.** *Assume that, for all $x \in \partial\mathcal{C}^s$, $G_2(x)$ is right invertible. Then, under Assumption 1, choosing γ and ϵ such that $\gamma\delta > \epsilon$ ensures that Condition 1 is satisfied for some compact input set $\mathcal{U} \subsetneq \mathbb{R}^m$.* •

Proof. Since, by Proposition III.4, $\partial\mathcal{C}^s$ is compact and the left-hand side of (9) is continuous in x, $a_i^\top(\gamma x_2 + f_2(x))$ is bounded in $\partial\mathcal{C}^s$. But $a_i^\top G_2(x) G_2(x)^\dagger y_x = a_i^\top y_x \geq \gamma\delta - \epsilon > 0$. So there is a finite ρ that validates (9) for all x with $u_x = \rho G_2(x)^\dagger y_x$. Noting the boundedness of y_x and $G_2(x)^\dagger$ in $\partial\mathcal{C}^s$ completes the proof. □

IV. Determining Input Magnitude for Control Invariance of Euler-Lagrange Systems

In this section we consider the class of Euler-Lagrange systems [28] and study how large the input set should be to render $\mathcal{C}^s$ control-invariant.

Assumption 2 (Input Set Structure and Euler-Lagrange Systems Properties)**.** Let $\mathcal{U} \supseteq \{u \in \mathbb{R}^m \mid \|u\| \leq d\}$ for some $d > 0$. Further, assume the dynamics (1) is such that:

(a) the matrix function $G_2(x)$ is only dependent on x_1 and has right inverse $G_2^\dagger(x_1)$ defined for all $x \in \mathcal{C}$; and
(b) $f_2(x) = f_2^1(x_1) + f_2^2(x)$, with $\|f_2^2(x)\| \leq k_2\|x_2\|$. •

Assumption 2(a) can be interpreted as having the inertia matrix $G_2^\dagger(x)$ independent of the system's velocity. Assumption 2(b) splits the forcing on the system into potential forces, such as gravity, $f_2^1(x_1)$, and other forces, such as damping, $f_2^2(x)$. It also asks that the magnitude of the latter be at most proportional to the system's velocity $\|x_2\|$. Define

$$k_1 := \max_{x \in \mathcal{C}} \|f_2^1(x_1)\| \quad \text{and} \quad k_G := \max_{x \in \mathcal{C}} \|G_2^\dagger(x_1)\|. \qquad (11)$$

The existence of these constants is ensured by the continuity of f_2 and $G_2^\dagger$ and the compactness of $\text{Proj}_n(\mathcal{C})$. Our approach to establishing control-invariance under Assumption 2 has two steps. First, we show that the velocity magnitude $\|x_2\|$ in the safe set $\mathcal{C}^s$ can be arbitrarily bounded by the design parameter γ, cf. Lemma IV.1. Second, we show that $\mathcal{U}$ in Assumption 2 is sufficient for control-invariance when $\|x_2\|$ is forced to be small enough through suitable design of γ, cf. Theorem IV.2.

Lemma IV.1 (Bound on Velocity Magnitude in Safe Set)**.** *Under Assumption 1, there exists a constant c that depends only on $\{a_i, b_i\}_{i\in\mathcal{I}}$ defining $\{h_i\}_{i\in\mathcal{I}}$ such that, for all γ and ϵ satisfying $\gamma\delta > \epsilon$, $\|x_2\| \leq \gamma c$ for all $x = (x_1, x_2) \in \mathcal{C}^s$.* •

Proof. By Proposition III.4, $\mathcal{C}^s$ is bounded. Therefore, each of its (finitely many) components $\cap_{i\in\bar{\mathcal{I}}^\ell}\mathcal{C}_i^s$ is bounded too. Since each $\mathcal{C}_i^s$ is a half-space, $\cap_{i\in\bar{\mathcal{I}}^\ell}\mathcal{C}_i^s$ is a bounded polytope. Given $\ell \in \bar{\mathcal{L}}$, consider the n programs

$$\begin{aligned} x_j^* = \arg\max\, &|e_j^\top x_2| \\ \text{s.t. } &x \in \cap_{i\in\bar{\mathcal{I}}^\ell}\mathcal{C}_i^s, \end{aligned}$$

where $j \in \{1, \dots, n\}$. Let $j_\ell^* := \arg\max_{1\leq j\leq n}\{|e_j^\top x_{j,2}^*|\}$, where $x_{j,2}^*$ denotes the last n components of x_j^*, and $x^{*,\ell} := x_{j_\ell^*}^*$. Since $x^{*,\ell} \in \mathbb{R}^{2n}$ is a solution to a linear program over a polytope, it is a vertex of the polytope [29, Thm. 2.4]. By [30, Thm. 10.4], there are $2n$ indices $\mathcal{I}_v \subseteq \bar{\mathcal{I}}^\ell$ such that $B_i(x^{*,\ell}) = 0$, for all $i \in \mathcal{I}_v$. Direct evaluation yields

$$\begin{aligned} a_i^\top \begin{bmatrix} I & \mathbf{0}_{n\times n} \end{bmatrix} x^{*,\ell} &= -b_i, \text{ if } i \in \mathcal{I}_v \cap \mathcal{I}, \\ a_i^\top \begin{bmatrix} \gamma I & I \end{bmatrix} x^{*,\ell} &= -\gamma b_i + \epsilon, \text{ if } i + r \in \mathcal{I}_v. \end{aligned}$$

Stacking the above equations into one matrix equation gives

$$\begin{bmatrix} A_1 & \mathbf{0} \\ \gamma A_2 & A_2 \end{bmatrix} x^{*,\ell} = \begin{bmatrix} b_1 \\ \gamma b_2 + \epsilon\mathbf{1_n} \end{bmatrix}, \qquad (12)$$

with appropriate matrices A_1, A_2, b_1 and b_2. Since $x^{*,\ell}$ is a unique solution as it is a vertex, the coefficient matrix in (12) is invertible, and thus is of rank $2n$. The matrices A_1 and A_2 have n columns, so their ranks are at most n. Since the rank of the block $\begin{bmatrix} \gamma A_2 & A_2 \end{bmatrix}$ is equal to the rank of A_2, then it must be that $\text{rank}(A_1) = \text{rank}(A_2) = n$, and hence A_1 and A_2 are invertible. Solving for $x_2^{*,\ell}$, which is the vector comprising the last n components of $x^{*,\ell}$, gives

$x_2^{*,\ell} = \gamma(A_2^{-1}b_2 - A_1^{-1}b_1) + \epsilon A_2^{-1}\mathbf{1_n}$. By definition of $x^{*,\ell}$ and j_ℓ^* and the triangle inequality, for all $j \in \{1, \ldots, n\}$ and all $x = (x_1, x_2) \in \cap_{i \in \bar{\mathcal{I}}^\ell} \mathcal{C}_i^s$,

$$|e_j^\top x_2| \le |e_{j_\ell^*}^\top x_2^{*,\ell}| \le \gamma c_1' + \epsilon c_2' \le \gamma(c_1' + \delta c_2'),$$

where $c_1' = |e_{j_\ell^*}^\top (A_1^{-1}b_1 - A_2^{-1}b_2)|$ and $c_2' = \|A_2^{-1}\mathbf{1_n}\|$. This holds for each $\ell \in \bar{\mathcal{L}}$ with the appropriate A_1, A_2, b_1 and b_2, so every component of $x_2 \in \mathcal{C}^s = \cup_{j \in \bar{\mathcal{L}}} \cap_{i \in \bar{\mathcal{I}}^\ell} \mathcal{C}_i^s$ is bounded by $\gamma c'^*$, where c'^* is the greatest of the finite constants $c_1' + \delta c_2'$ for the different ℓ's. Hence, $\|x_2\| \le \gamma\sqrt{n}c'^*$. □

Interestingly, Lemma IV.1 can be leveraged to meet any safety specification on the magnitude of x_2 by taking a sufficiently small γ. The smaller γ is, the slower the system will move, possibly hindering other control objectives. Lemma III.1, however, ensures that no positions are lost no matter how small γ is chosen to be. We are now ready to show that Assumption 2 is enough to establish the existence of a design parameter γ that makes $\mathcal{C}^s$ control-invariant with bounded input set $\mathcal{U} \supseteq \{u \in \mathbb{R}^m \mid \|u\| \le d\}$.

Theorem IV.2 (Control-Invariance with Input Constraints). *Under Assumptions 1 and 2, if $d - k_G k_1 > 0$, cf. (11), then γ and ϵ can be chosen to ensure Condition 1 is satisfied.* •

Proof. Let γ be such that $\gamma(k_2+\gamma)k_G c < \frac{1}{2}(d - k_1 k_G)$, where c is given in Lemma IV.1. Select ϵ to satisfy $\gamma\delta > \epsilon$. For all $x \in \partial\mathcal{C}^s$ and $i \in \mathcal{I}_{\text{act}}(x)$ with $i > r$, recall from the proof of Proposition III.6 that $a_i^\top y_x > 0$, where y_x is as defined there. Let $\beta_x > 0$ satisfy $\beta_x\|y_x\| \le \frac{1}{2}(d - k_1 k_G)$ and choose $u_x = -G_2^\dagger(x)(f_2(x) + \gamma x_2) + \beta_x y_x$. By the triangle inequality and the definition of the matrix induced 2-norm,

$$\|u_x\| \le \|G_2^\dagger(x)\|(\|f_2^1(x_1)\| + \|f_2^2(x)\| + \gamma\|x_2\|) + \beta_x\|y_x\|.$$

By Assumption 2, Lemma IV.1, and (11), $\|u_x\| \le k_G k_1 + \gamma(k_2+\gamma)k_G c + \beta_x\|y_x\|$. By our choice of γ and β_x, $\|u_x\| \le k_G k_1 + \frac{1}{2}(d - k_G k_1) + \frac{1}{2}(d - k_G k_1) = d$, and thus $u_x \in \mathcal{U}$. Noting that $u_x \in \mathcal{U}$ validates (9) for all $i \in \mathcal{I}_{\text{act}}(x)$ with $i > r$ completes the proof. □

The condition $d - k_G k_1 > 0$ in Theorem IV.2 amounts to having enough control authority to counter the potential force. In the absence of such a force (i.e., when $k_1 = 0$, cf. Section V), this condition is valid by default allowing $\mathcal{C}^s$ to be made safe for any bounded $\mathcal{U}$. If, however, no such control is available, the system might be inherently unsafe.

V. Safe Control of Robotic Manipulator

In this section, similar to [18], we illustrate our results on a 2-degree-of-freedom planar elbow manipulator [31] but with linear nonconvex constraints instead of the hypercubic constraints in [18]. The system consists of two horizontally oriented links (no gravity) hinged to each other, with the first link hinged to a fixed frame. Torque manipulation is available at each joint. The system model is $\mathcal{M}\begin{bmatrix}\ddot\theta_1 & \ddot\theta_2\end{bmatrix}^\top = C\begin{bmatrix}\dot\theta_1 & \dot\theta_2\end{bmatrix}^\top + \begin{bmatrix}u_1 & c_{25}\cos(\theta_1+\theta_2) + u_2\end{bmatrix}^\top$ where

$$\mathcal{M} = \begin{bmatrix} c_{11} + c_{12}\cos(\theta_2) & c_{13} + c_{14}\cos(\theta_2) \\ c_{22} + c_{23}\cos(\theta_2) & c_{21}\end{bmatrix}, \qquad (13)$$

$$C = \begin{bmatrix} c_{15}\sin(\theta_2)\dot\theta_2 & c_{16}\sin(\theta_2)\dot\theta_2 \\ c_{24}\sin(\theta_2)\dot\theta_1 & 0\end{bmatrix}$$

with constant coefficients c_{ij} dependent on the links' lengths and masses, cf. [31]. We require the arm to avoid collision with two walls, as shown in Figure 1a.

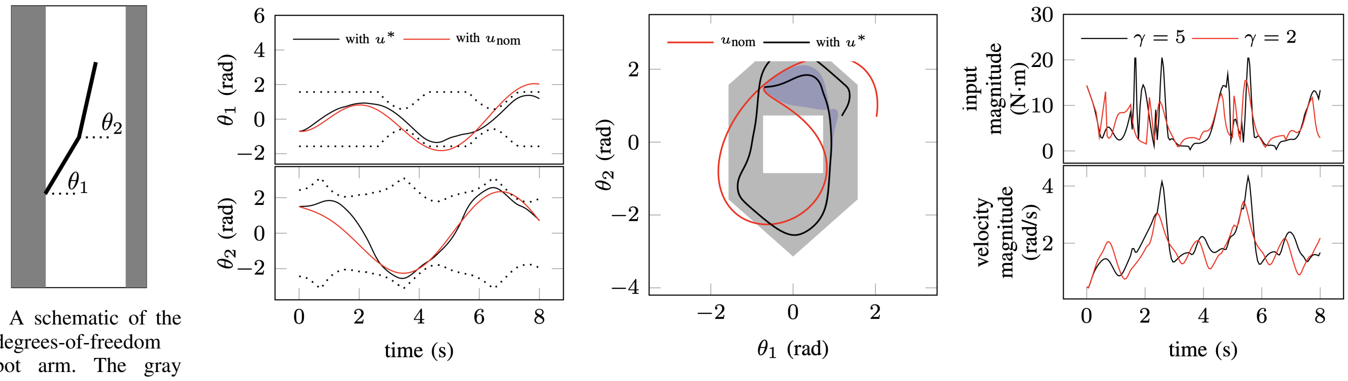
In the space of angles, the positional safety constraints due to the first wall correspond to the interior of the dotted hexagon in Figure 1c. The safety constraints of the parallel wall are not linear in the space of angles. So we overapproximate them by the exterior of the square at the center in Figure 1c. This makes the safety requirement set, $\mathcal{C}$, the intersection of the interior of the hexagon and the exterior of the square, which is nonconvex. For brevity, we omit the details of the specific halfspaces and the formulation of the set in the form of (3). We construct the control-invariant set $\mathcal{C}^s$ as described in Section III, with $\gamma = 4$ and $\epsilon = 0.5$. It is not difficult, albeit a bit lengthy, to show that the points y_I can be chosen such that our choice of γ and ϵ satisfy $\gamma\delta > \epsilon$ for δ defined as in (7). The controller u^* is computed according to (8), with the objective function $\|u - u_{\text{nom}}(t)\|^2 + M^2 + \alpha^2$. Here, u_{nom} is a nominal controller that tracks $r = (\pi\sin(t), \frac{\pi}{2}\sin(4t))$, cf. [32], $u_{\text{nom}}(t) = \mathcal{M}(\ddot r - \dot e - e) + C\begin{bmatrix}\dot\theta_1 & \dot\theta_2\end{bmatrix}^\top$ where $e = (\theta_1, \theta_2) - r$ and $\mathcal{M}$ and C are as defined in (13).

Figure 1b shows the time evolution of the joint angles under the nominal controller and under the safety-filtered controller u^*, along with the hexagon constraints evolution as viewed from the position of the safe trajectory. Figure 1c shows a phase portrait of the evolution of the angles under the nominal and the safe controllers, showing an average of 50 trajectories with different initial velocity. Those trajectories are all contained in the blue area. The safe controller renders the nonconvex $\mathcal{C}$ invariant, per Corollary III.7, while the nominal controller does not.

Note that our example's model satisfies Assumption 2 with $f_2^1(x) = \mathcal{M}^{-1}C\begin{bmatrix}\dot\theta_1 & \dot\theta_2\end{bmatrix}^\top$, $f_2^1(x) = \mathcal{M}^{-1}\begin{bmatrix}0 & c_{25}\cos(\theta_1+\theta_2)\end{bmatrix}^\top$ and $G_2(x_1) = M^{-1}$. Thus, by Theorem IV.2, any $\mathcal{U} = \{u \in \mathbb{R}^2 \mid \|u\| \le d\}$ such that $d > k_G k_1$ suffices for invariance, with a sufficiently small γ and a suitably chosen ϵ. This is reflected in Figure 1d, which shows the effect of the design parameter γ in constraining the control and velocity magnitudes. As shown there, lower values of γ allow for safety with lower control magnitudes, at the expense of reducing the velocity of the execution. Finally, we note that extending this example to higher degrees of freedom adds complexity to the polytopic representation of the constraints but, once available, the computation of the control design remains the same.

VI. Conclusions

Given a second-order system and positional safety specifications described by linear boundaries, we have identified conditions that allow the explicit construction of a verifiably safe set in the full state space. We have also designed an associated QP controller that ensures this set is safe. The identified conditions are always satisfied by fully actuated systems and, in the case of Euler-Lagrange systems, the controller design can incorporate velocity and input constraints. Future work will explore



(a) A schematic of the 2-degrees-of-freedom robot arm. The gray space is the wall that should be avoided during manipulation.

(b) Time evolution of the link angles. The dotted lines are the safety constraints as seen from the safe trajectory.

(c) Trajectories on the position plane, showing the invariance enforced by the controller u^* instead of u_{nom}.

(d) Time evolution of the average velocity and average control effort on both links for different values of γ.

Fig. 1: Simulation results for safe control of a 2-link robot arm.

second-order systems with other forms of underactuation, incorporate time-varying safety considerations, robustify the approach to handle system uncertainties, and apply a similar safe-set construction to non-linear specifications.

References

[1] M. Nagumo. Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan*, 24:551–559, 1942.

[2] H. Khalil. *Nonlinear Systems, 3rd ed.* Prentice Hall, Englewood Cliffs, NJ, 2002.

[3] P. Wieland and F. Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007.

[4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: theory and applications. In *European Control Conference*, pages 3420–3431, Naples, Italy, 2019.

[5] R. Konda, A. D. Ames, and S. Coogan. Characterizing safety: minimal control barrier functions from scalar comparison systems. *IEEE Control Systems Letters*, 5(2):523–528, 2021.

[6] M. Ghanbarpour, A. Isaly, R.G. Sanfelice, and W. E. Dixon. Optimal safety for constrained differential inclusions using nonsmooth control barrier functions. *IEEE Control Systems Letters*, 7:1303–1308, 2022.

[7] P. Glotfelter, J. Cortés, and M. Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE Control Systems Letters*, 1(2):310–315, 2017.

[8] L. Lindemann and D. V. Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, 3(1):96–101, 2019.

[9] K. Long, C. Qian, J. Cortés, and N. Atanasov. Learning barrier functions with memory for robust safe navigation. *IEEE Robotics and Automation Letters*, 6(3):4931–4938, 2021.

[10] C. Wang, Y. Meng, Y. Li, S. L. Smith, and J. Liu. Learning control barrier functions with high relative degree for safety-critical control. In *European Control Conference*, pages 1459–1464, Delft, Netherlands, 2021.

[11] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela. Synthesis of control barrier functions using a supervised machine learning approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 7139–7145, Las Vegas, NV, USA, 2020.

[12] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni. Learning control barrier functions from expert demonstrations. In *IEEE Conf. on Decision and Control*, pages 3717–3724, Jeju, South Korea, 2020.

[13] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert. Robust control barrier-value functions for safety-critical control. In *IEEE Conf. on Decision and Control*, pages 6814–6821, Austin, TX, USA, 2021.

[14] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames. Safe backstepping with control barrier functions. In *IEEE Conf. on Decision and Control*, pages 5775–5782, Cancún, Mexico, 2022.

[15] A. Clark. Verification and synthesis of control barrier functions. In *IEEE Conf. on Decision and Control*, pages 6105–6112, Austin, TX, USA, 2021.

[16] M. Schneeberger, F. Dörfler, and S. Mastellone. SOS construction of compatible control Lyapunov and barrier functions. *IFAC-PapersOnLine*, 56(2):10428–10434, 2023. 22nd IFAC World Congress.

[17] J. Zheng, T. Dai, J. Miller, and M. Sznaier. Robust data-driven safe control using density functions. *IEEE Control Systems Letters*, 7:2611–2616, 2023.

[18] W. S. Cortez and D. V. Dimarogonas. Safe-by-design control for Euler–Lagrange systems. *Automatica*, 146:110620, 2022.

[19] A. W. Farras and T. Hatanaka. Safe control with control barrier function for Euler-Lagrange systems facing position constraint. In *SICE International Symposium on Control Systems*, pages 28–32, Tokyo, Japan, 2021.

[20] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[21] B. J. Morris, M. J. Powell, and A. D. Ames. Continuity and smoothness properties of nonlinear optimization-based feedback controllers. In *IEEE Conf. on Decision and Control*, pages 151–158, Osaka, Japan, Dec 2015.

[22] P. Mestres, A. Allibhoy, and J. Cortés. Regularity properties of optimization-based controllers. *European Journal of Control*, 81:101098, 2025.

[23] M. Alyaseen, N. Atanasov, and J. Cortés. Continuity and boundedness of minimum-norm CBF-safe controllers. *IEEE Transactions on Automatic Control*, 70(6):4148–4154, 2025.

[24] M. Alyaseen, N. Atanasov, and J. Cortés. Safety-critical control of discontinuous systems with nonsmooth safe sets. *Automatica*, 2024. Submitted.

[25] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 8129–8136, 2021.

[26] W. Xiao and C. Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2022.

[27] B. Grünbaum, V. Klee, M. A. Perles, and G. C. Shephard. *Convex Polytopes*, volume 16. Springer, 2 edition, 2003.

[28] R. Ortega, A. Loría, P. J. Nicklasson, and H. Sira-Ramírez. *Passivity-based Control of Euler-Lagrange Systems*. Springer, New York, 1998.

[29] D. Goldfarb and M. J. Todd. Chapter ii linear programming. In G.L. Nemhauser, A.H.G. Rinooy Kan, and M.J. Todd, editors, *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 73–170. Elsevier, 1989.

[30] A. Brondsted. *An Introduction to Convex Polytopes*, volume 90. Springer Science & Business Media, 2012.

[31] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, New York, 2020.

[32] W. S. Cortez and D. V. Dimarogonas. Correct-by-design control barrier functions for Euler-Lagrange systems with input constraints. In *American Control Conference*, pages 950–955, Denver, CO, USA, 2020.