Safety-Critical Control of Discontinuous Systems with Nonsmooth Safe Sets ⋆

Mohammed Alyaseen Nikolay Atanasov Jorge Cortes

University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA

Abstract

This paper studies the design of controllers for discontinuous dynamics that ensure the safety of non-smooth sets. The safe set is represented by arbitrarily nested unions and intersections of 0-superlevel sets of differentiable functions. We show that the satisfaction of the point-wise active safety constraints only does not necessarily imply safety. This rules out the standard techniques developed for safety of continuous dynamics. This motivates the introduction of the notion of transition functions, which allow us to incorporate even the inactive safety constraints without falling into unnecessary conservatism. These functions allow system trajectories to leave a component of the nonsmooth safe set to transition to a different one. The resulting controller is then defined as the solution to a convex optimization problem, which we show is feasible and continuous wherever the system dynamics is continuous. We illustrate the effectiveness of the proposed design approach in a multi-agent coverage control problem.

Key words: safety; discontinuous systems; nonsmooth control barrier functions; optimization-based control design

1 Introduction

Safety-critical control is a fundamental problem in numerous domains including autonomous driving, power and transportation systems, robotics, and even mitigation of epidemic spreading. This paper tackles the synthesis of easy-to-compute controllers for the objective of safety. Specifically, we address the problem of safety-critical control to render a desired nonsmooth set safe under the trajectories of discontinuous dynamics. Discontinuous dynamical systems are convenient for modeling a wide range of behaviors, including motion subject to Coulomb friction, systems with abrupt changes in forces, and dynamics of multi-agent systems under connectivity maintenance or collision avoidance constraints. Likewise, the flexibility enabled by nonsmooth safe sets allows us to encode safety requirements that cannot be expressed merely by a differentiable function, such as keeping the system evolution in or out of a set with nonsmooth boundary, or in some nonsmooth set and out of another simultaneously. Allowing nonsmoothness is also motivated by the fact that dealing with multiple constraints, some of which are active in some regions while different ones are active in others, can be represented as an overall nonsmooth safety requirement. The unique challenge tackled here is constructing an explicit, provably feasible, sufficiently regular, optimization-based safe controller for discontinuous systems with general nonsmooth sets defined by nested unions and intersections of smooth component sets.

Literature review: Safety-critical control is often formulated mathematically as a control design problem to achieve forward invariance of a desired set of the state space. The notion of control barrier function (CBF) has gained popularity as a tool for invariance control due to its versatility [2, 3, 23]. Broadly speaking, a CBF is a function for which every system state in its 0-superlevel set admits an input value that does not allow the dynamics to leave the superlevel set instantaneously. For convenience, we refer to this point-wise condition as the CBF condition. Achieving control invariance via CBFs is a twofold undertaking. Firstly, one must find a function for which the CBF condition holds. Secondly, one must design a controller with adequate regularity properties that keeps the the superlevel set of the function invariant. These two problems are in general distinct. Even if a differentiable CBF is available for continuous dynamics, there might not exist a continuous controller rendering its superlevel set invariant, cf. [1].

In the classical setting, a CBF is a differentiable function whose 0-superlevel set is made control-invariant by employing a Lipschitz controller that satisfies the CBF condition [3]. A widely used technique for constructing such a controller is quadratic programming (QP), constrained by the CBF condition, which is linear in the input for control-affine dynamics [2]. The differentiability requirement of the CBF stems from the use of its gradient in the CBF condition, while the Lipschitz requirement on the dynamics and controller ensures existence and uniqueness of solutions for the closed-loop system [13]. This excludes discon-

⋆ This paper was not presented at any IFAC meeting. Corresponding author Mohammed Alyaseen. All authors are with the Contextual Robotics Institute, University of California San Diego. M. Alyaseen is also affiliated with Kuwait University as a holder of a scholarship. This work was partially supported by AFOSR Award FA9550-23-1-0740.

Email addresses: malyasee@ucsd.edu (Mohammed Alyaseen), natanasov@ucsd.edu (Nikolay Atanasov), cortes@ucsd.edu (Jorge Cortes).

tinuous dynamics and non-smooth safe sets, which can be limiting in many applications. However, these two structural assumptions are not indispensable, and indeed many works have explored the extension of CBF techniques beyond them. In one line of work, safety conditions are developed for autonomous hybrid systems and non-smooth barrier functions [15]. These conditions extend to systems modeled by differential inclusions with control inputs as in [8, 12]. The work [8] provides general conditions for the existence of a continuous safe controller but does not provide an explicit one. Instead, [12] gives an explicit optimization-based continuous safe controller for non-smooth safe sets defined as an intersection of superlevel sets of differentiable functions. The work [9] gives sufficient conditions for safety of differential inclusions and general non-smooth barrier functions, and designs an optimization-based controller for sets defined by the intersection of multiple superlevel sets. This controller is safe assuming the optimization is feasible but no conditions are provided for the latter. These methods have been successfully applied for multi-agent network connectivity [16]. Other works [20] are not limited to continuous controllers and instead develop Lebesgue-measurable controllers for non-smooth CBFs whose superlevel sets are given as an intersection of multiple differentiable functions. The works mentioned hitherto give explicit safe controllers only for non-smooth sets defined as intersections, that is, conjunctive safety constraints. When the safe set is given as a union of component sets, the safety constraint on the controller changes depending on the component set in which the current state lies, leading to an optimization with disjunctive constraints. Such problems are equivalent to mixed-integer optimization programs [10] and thus are more challenging to deal with. Disjunctive safety constraints appear in another line of work, that of enforcing signal temporal logic (STL) specifications as barrier functions. The work [14] models conjunctive signal temporal tasks for Lipschitz dynamics as non-smooth barrier functions and provides a generally discontinuous optimization-based controller to enforce such constraints. The works [22, 24] do the same for logical specifications including disjunction. Although the STL-induced CBF is more general than mere time-independent CBFs, the controller developed in those works is generally discontinuous, even with the assumed Lipschitz dynamics, and is provably un-safe for discontinuous systems, as we establish below.

Contributions: We consider the problem of constructing an explicit, provably feasible, optimization-based safe controller for discontinuous systems with non-smooth sets defined by nested unions and intersections of smooth sets. We start by considering the *active-component* QP controller, an optimization-based controller satisfying only the pointwise active safety constraints. We show by a counterexample that this controller is generally un-safe for discontinuous systems and nonsmooth safe sets, thereby excluding the possibility of using the techniques developed for safe control of continuous dynamics in our setting. We leverage our analysis of the reasons behind this failure to construct a new QP controller that overcomes them. We do this by considering all safety constraints, even those that are inactive at the point at which the control is computed. Thus we introduce the *all-components* QP controller. The inclusion of all safety constraints allows the inactive constraints to influence the control action. Although this method adds more constraints than needed for safety, we show that it is not limiting for a wide range safe sets. This is due to a relaxation of the inactive constraints introduced by the novel concept of *transition function*, thereby avoiding conservatism. The inclusion of all safety constraints ensures safety while enabling the continuity of the optimization-based controller. Specifically, we show that our proposed controller is continuous wherever the dynamics are continuous even at points of nonsmoothness of the safe set. We prove the feasibility of the all-components QP controller for any safe set given by nested intersections and unions of smooth sets. Notice that since we allow representing our safe sets as unions, the safety conditions include disjunctive constraints. However, instead of solving the problem using disjunctive optimization, we develop the all-components QP controller which we show is feasible whenever the disjunctive program is. This QP representation of our controller is what deems it continuous even at the points where the disjunctive safety constraints discontinuously change. Our last contribution shows how the design parameters of our all-components QP controller can be chosen adaptively dependent on the state. This results in the introduction of the *all-components adaptive* QP controller. Finally, we demonstrate the efficacy of our proposed controller by applying it to control the motion of multi-agents with mixed safety specifications.

Notation: For $x \in \mathbb{R}^n$, we define $B_\epsilon(x) \coloneqq \{y \in \mathbb{R}^n \mid \|x - y\| < \epsilon\}$. For a set $\mathcal{S} \subseteq \mathbb{R}^n$, $B_\epsilon(\mathcal{S}) \coloneqq \bigcup_{x\in\mathcal{S}} B_\epsilon(x)$. Given a function $s : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}$, $s \in C$ means s is continuous and $s \in C^n$ means s has a continuous n^{th} derivative. A function $\alpha : (-a, b) \to \mathbb{R}$ is a class-κ function if it is strictly increasing and $\alpha(0) = 0$. The cardinality, boundary, interior, closure, and convex hull of a set $\mathcal{S}$ are denoted by $|\mathcal{S}|$, $\partial\mathcal{S}$, $\text{int}(\mathcal{S})$, $\text{cl}(\mathcal{S})$, and $\text{co}(\mathcal{S})$, respectively. We write $F : \mathcal{S} \rightrightarrows \mathcal{S}'$ to denote that F maps elements of $\mathcal{S}$ to subsets of $\mathcal{S}'$. A neighborhood of a $\mathcal{S}$ is a set containing an open set that contains $\mathcal{S}$. Table 1 provides a summary of key notation used throughout the technical exposition.

Table 1
Summary of key notation.

$\mathcal{I}(x)$	Indices of active barriers at x
$\mathcal{J}(x)$	Indices of active dynamics at x
$\tilde{\mathcal{I}}(x)$	arbitrary superset of $\mathcal{I}(x)$, cf. Assumption 1
$\mathcal{L}$, $\mathcal{I}^\ell$	Indices for safe set normal form, cf. Lemma 4.8

2 Problem Statement

Our goal is to construct computationally-inexpensive controllers that render solutions of possibly discontinuous dynamics forward invariant with respect to possibly non-smooth sets. We consider a the control-affine dynamics:

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

with $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u \in \mathcal{U}$, where $\mathcal{U}$ is a convex set in $\mathbb{R}^m$. We assume that $f : \mathcal{X} \to \mathbb{R}^n$ and $G : \mathcal{X} \to \mathbb{R}^{n\times m}$ are piecewise continuous [6], that is, there is a collection of open, disjoint sets $\mathcal{X}_1, \ldots, \mathcal{X}_\omega \subseteq \mathcal{X}$ with the closure of their union covering $\mathcal{X}$, and such that (f, G) is continuous on each $\mathcal{X}_j$. For simplicity, we assume that for each $j \in \{1, \ldots, \omega\}$, there is a continuous map $(f_j, G_j) : \mathcal{X} \to \mathbb{R}^n \times \mathbb{R}^{n\times m}$ such that $(f(x), G(x)) = (f_j(x), G_j(x))$ for all $x \in \mathcal{X}_j$. We denote the set of active dynamics by $\mathcal{J}(x) \coloneqq \{j \in \{1, \ldots, \omega\} \mid x \in \text{cl}(\mathcal{X}_j)\}$. Accordingly, $\mathcal{J}(x)$ is a singleton almost everywhere.

Our assumptions on the system dynamics are not as restrictive as assuming the Lipschitzness of the dynamics, and not as permissive as differential inclusions [6]. Unlike Lipschitz dynamics, our model can capture systems with Coulomb friction for instance. Our model, however, is more structured than a differential inclusion. This additional structure allows for convenient control techniques, as subsequent work will show. We next describe the types of sets that we want to keep forward invariant under the dynamics. Let $h_i : \mathcal{X} \to \mathbb{R}$, $i \in \{1, \dots, r\}$ be C^1 on $\mathcal{X}$ with associated 0-superlevel sets $\mathcal{C}_i \coloneqq \{x \in \mathbb{R}^n \mid h_i(x) \geq 0\} \subseteq \mathcal{X}$. Let

$$h : \mathcal{X} \to \mathbb{R} \tag{2}$$

be a piecewise differentiable Lipschitz function. That is, h is Lipschitz and there are r open, disjoint sets $\mathcal{X}^{h_1}, \dots, \mathcal{X}^{h_r}$ with the closure of their union covering $\mathcal{X}$ and $h(x) = h_i(x)$ for all $x \in \mathcal{X}^{h_i}$. We refer to the functions $\{h_i\}_{i=1}^r$ as the components of h, and denote the set of active components by $\mathcal{I}(x) \coloneqq \{i \in \{1, \dots, r\} \mid x \in \mathrm{cl}(\mathcal{X}^{h_i})\}$. Accordingly, $\mathcal{I}(x)$ is a singleton almost everywhere. We denote by $\mathcal{C} \coloneqq \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$ the 0-superlevel set of h.

Our aim is to construct a controller $u = k(x)$ for the dynamics (1) that renders $\mathcal{C}$ forward invariant under the closed-loop solutions of (1). We refer to such a controller as *safe*. To formalize this objective, it is necessary to specify the notion of solution to such discontinuous dynamics with a possibly discontinuous controller. To do so, we adopt the widely-used notion of Filippov solution [7].

Given a differential equation $\dot{x} = X(x)$, where $X : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}^n$, the Filippov set-valued map for $x \in \mathcal{X}$ is

$$F[X](x) \coloneqq \bigcap_{\delta > 0} \bigcap_{\eta(\mathcal{S})=0} \bar{\mathrm{co}}\{X(B_\delta(x) \setminus \mathcal{S})\}, \tag{3}$$

where $\bar{\mathrm{co}}$ denotes the convex closure and $\eta(\mathcal{S})$ the Lebesgue measure of $\mathcal{S}$. A *Filippov solution of* $\dot{x} = X(x)$ is an absolutely continuous map $\phi : [t_0, t_1] \to \mathcal{X}$ that satisfies $\dot{\phi}(t) \in F[X](\phi(t))$ for almost all $t \in [t_0, t_1]$. For the piecewise-continuous control-affine dynamics with a piecewise continuous controller $k : \mathcal{X} \to \mathcal{U}$, the Filippov set-valued map simplifies to [6]:

$$F[f + Gk](x) = \\ \mathrm{co}(\{\lim_{\mu \to \infty} (f(x_\mu) + G(x_\mu)k(x_\mu)) \mid x_\mu \to x, x_\mu \notin S\}), \tag{4}$$

where S is the set in which $f + Gk$ is discontinuous and $\{x_\mu\}$ is just a sequence with subscript $\mu \in \{1, 2, \dots\}$. We are now ready to define what we mean by a safe controller.

Definition 2.1 (Safe Feedback Controller and Set Invariance). A feedback controller $k : \mathcal{X} \to \mathcal{U}$ for the general non-autonomous dynamics $\dot{x} = \bar{f}(x, u)$ is *safe with respect to* $\mathcal{C} \subseteq \mathcal{X}$ if

(i) there exists at least one Filippov solution ϕ to $\dot{x} = \bar{f}(x, k(x))$ starting from any point in $\mathcal{C}$ and,
(ii) all Filippov solutions ϕ starting at $\mathcal{C}$ remain in $\mathcal{C}$ for all $t > 0$. •

The set $\mathcal{C}$ is *(forward) invariant under* k or just *controlled-invariant* if such a k exists. We use the term *safeness* when attributed to the control and the term *safety* when attributed to the set. We state next precisely the problem tackled in this work.

Problem 1 Find a controller for (1) that

(i) is continuous wherever the dynamics are continuous,
(ii) is safe with respect to $\mathcal{C}$, and
(iii) solves a feasible convex optimization problem. •

By feasible optimization, we mean that the constraints of the program can always be satisfied by some control values. The requirement that the controller solves a convex optimization problem is motivated by the prevalence of QP controllers for the simpler case of Lipschitz dynamics with smooth safe sets, cf. [2, 3]. The light computational effort required to solve such programs makes them useful for fast online control computation.

We aim here to enforce safety for the more general case of discontinuous dynamics and non-smooth sets without adding computational complexity to the control synthesis. Also, as seen from the first requirement, we only require our controller to be continuous when the dynamics are. In other words, we do not demand from the controller regularity properties that the dynamics does not have. This, we believe, is a reasonable middle ground between demanding continuity of the controller for discontinuous dynamics [12] and allowing discontinuity of the controller with continuous dynamics [20, 22, 24].

3 Review of Sufficient Conditions for Safe Control

We review here the state-of-the-art conditions on any control that renders a nonsmooth control-affine dynamics of the form (1) safe with respect to a given set. We employ the notion of generalized gradient [5, Thm. 2.5.1] of a Lipschitz function h, $\partial h(x) = \mathrm{co}(\{\lim_{i \to \infty} \nabla h(x_i) \mid x_i \to x, x_i \notin \Omega_h\})$, where Ω_h is the zero-measure set on which h is non-differentiable. For h that satisfies our description in Section 2, we have

$$\partial h(x) = \mathrm{co}(\{\nabla h_i(x) \mid i \in \mathcal{I}(x)\}). \tag{5}$$

We also use the notion of generalized Lie derivative [6] of a Lipschitz function h with respect to a set-valued map $F : \mathcal{S} \rightrightarrows \mathbb{R}^n$,

$$\tilde{\mathcal{L}}_F h(x) \coloneqq \{a \in \mathbb{R} \mid \text{there exists } v \in F(x), \text{ such that } \\ a = v^\top \zeta, \text{ for all } \forall \zeta \in \partial h(x)\}.$$

The following result, adapted from [9, Thm. 3], gives a general condition for safeness of a feedback controller.

Theorem 3.1 (Sufficient Condition for Safe Control [9, Thm. 3]). *Given $\epsilon > 0$, let $k : B_\epsilon(\mathcal{C}) \to \mathcal{U}$ be a feedback controller for the non-autonomous dynamics $\dot{x} = \bar{f}(x, u)$ and let F be the Filippov set-valued map associated to the closed-loop dynamics. Assume that $\dot{x} = \bar{f}(x, k(x))$ has a Filippov solution starting from every point in $\mathcal{C}$. If there exists a class-κ function α such that*

$$\inf \tilde{\mathcal{L}}_{F[\bar{f}(\cdot, k(\cdot))]} h(x) \geq -\alpha(h(x)), \tag{6}$$

in a neighborhood of $\partial\mathcal{C}$, then k is safe with respect to $\mathcal{C}$. More precisely, if (6) is satisfied in a neighborhood of $\bar{x} \in \partial\mathcal{C}$, then no Filippov solution of $\dot{x} = \bar{f}(x, k(x))$ can leave $\mathcal{C}$ from $\bar{x}$.

PROOF. Let $\phi : [t_0, t_1] \to B_\epsilon(\mathcal{C})$ be a Filippov solution of $\dot{x} = \bar{f}(x, k(x))$ and $\phi(t_0) \in \mathcal{C}$. Suppose by contradiction that ϕ leaves $\mathcal{C}$. Thus, without loss of generality, $\phi(t_1) \notin \mathcal{C}$, that is, $h(\phi(t_1)) < 0$. By the absolute continuity of h and ϕ and the facts that $h(\phi(t_0)) \geq 0$ and $h(\phi(t_1)) < 0$, there exists $\bar{t} \in [t_0, t_1]$ for which $h(\phi(\bar{t})) = 0$ and $h(\phi(t)) < 0$ for all $t \in (\bar{t}, t_1]$. To derive a contradiction, we use the fact that in a neighborhood of $\phi(\bar{t})$, we have $\inf \tilde{\mathcal{L}}_F h(x) \geq -\alpha(h(x))$ and assume without loss of generality that $\phi(t_1)$ is in that neighborhood. Due to absolute continuity, the time derivative $\frac{d}{dt}h(\phi(t))$ exists almost everywhere in $[t_0, t_1]$ and $\frac{d}{dt}h(\phi(t)) \in \tilde{\mathcal{L}}_F h(\phi(t))$ for almost every $t \in [t_0, t_1]$ [4, Lem. 1]. Thus, $\frac{d}{dt}h(\phi(t)) \geq -\alpha(h(t))$ almost everywhere in the interval $[\bar{t}, t_1]$. This gives

$$0 > h(\phi(t_1)) - h(\phi(\bar{t})) = \int_{\bar{t}}^{t_1} \frac{d}{dt} h(\phi(t)) dt \geq -\int_{\bar{t}}^{t_1} \alpha(h(\phi(t))) dt.$$

Recalling that $\alpha(h(\phi(t))) < 0$ on $(\bar{t}, t_1)$, as α is a class-κ function, the last inequality leads to a contradiction. •

Theorem 3.1 is a slight variation of [9, Thm. 3]: the difference is that condition (6) is required to hold on a neighborhood of $\partial\mathcal{C}$, whereas in [9, Thm. 3] it is required to hold on a neighborhood of $\mathcal{C}$. This is a minor advantage since an appropriate α can always be chosen to validate (6) if it is valid on a neighborhood of $\partial\mathcal{C}$ for a compact $\mathcal{C}$. The formulation here touches on whether a trajectory can leave the set from a specific single point, motivated by our ensuing discussion. Condition (6) is less strict than the requirement in [8] that $\zeta^\top v \geq \alpha(h(x))$ for all $\zeta \in F[\bar{f}(x, k(x))]$ and all $v \in \partial h(x)$. Note also that the safety condition (6) does not require continuity of k. From these two perspectives, Theorem 3.1 is a generalization of the safety conditions in [8].

4 Safe Control of Piece-wise Continuous Control-Affine Dynamics

Having described a sufficient condition for a feedback controller to be safe, here we tackle the problem of actually synthesizing the controller. To do so, our approach takes inspiration from studying the safety limitations of controllers developed for continuous settings. We start by considering a controller that we term *active-component QP controller*. It is a direct extension of the controller used in [14, 22, 24] to verify safety with non-smooth safe sets but with continuous dynamics. We study the extent to which this controller can yield safety with discontinuities in the dynamics. The naming of the active-component QP controller emphasizes the fact that its safety constraints at any state x are only concerned with the components of h active at that particular x. We show that this controller actually fails to enforce safety at points of non-differentiability of h. This motivates our ensuing control design to overcome these limitations, termed the *all-components QP controller*, and establish its safeness, feasibility, and regularity properties.

4.1 Active-Component QP Controller

The widely-used QP controller [2] for a differentiable CBF h solves a quadratic program, with the constraint being the classical CBF condition

$$\nabla h(x)^\top (f(x) + G(x)u) + \alpha(h(x)) \geq 0, \tag{7}$$

where α is a class-κ function. The constraint is guaranteed to be feasible because of the definition of CBF. The active-component QP controller is the analogue of this classical QP controller, and is given by the following QP:

$$\begin{aligned} u_{\text{act}}(x) \coloneqq \arg\min_{u \in \mathcal{U}} \; & u^\top Q(x) u + b(x)^\top u \\ \text{s.t. } & \nabla h_i(x)^\top (f(x) + G(x)u) + \alpha(h_i(x)) \geq 0, \; \forall i \in \mathcal{I}(x) \end{aligned} \tag{8}$$

where $Q : \mathcal{X} \to \mathbb{R}^{m \times m}$ and $b : \mathcal{X} \to \mathbb{R}^m$ are Lipschitz, $Q(x)$ is positive definite on $\mathcal{X}$, and α is a class-κ function. This controller can be seen as minimally constrained, in that it only asks that the safety constraint be satisfied for the active component of h, that is, h_i with $i \in \mathcal{I}(x)$, and for the dynamics at x only, hence the name *active-component QP controller*. A similar controller is introduced in works [14, 22, 24] involving STL specifications for settings with continuous dynamics.

Similarly to what is done for the classical QP controller, we need to make an assumption on the function h to ensure the constraints in the active-component QP controller are feasible.

Assumption 1 (Non-smooth Version of CBF Condition). Given the system dynamics (1) with piecewise continuous structure and a piecewise differentiable function h, let $\tilde{\mathcal{I}} : \mathcal{X} \rightrightarrows \{1, \ldots, r\}$ be a set-valued map such that $\mathcal{I}(x) \subseteq \tilde{\mathcal{I}}(x)$ for all $x \in \mathcal{X}$. We assume that for all $x \in \partial\mathcal{C}$, there exists $u_x \in \mathcal{U}$ such that

$$\nabla h_i(x)^\top (f_j(x) + G_j(x)u_x) > 0, \forall i \in \tilde{\mathcal{I}}(x), \forall j \in \mathcal{J}(x). \bullet$$

Assumption 1 simply requires that there are control values that steer the system from the boundary of $\mathcal{C}$ to its interior. It can be shown that the satisfaction of Assumption 1 with $\tilde{\mathcal{I}}(x) = \mathcal{I}(x)$ is all that is needed for the existence of a safe controller [8, Thm. 1]. However, for our purpose here, which is deriving an explicit QP controller, we will need the satisfaction of Assumption 1 with $\tilde{\mathcal{I}}(x)$ that is slightly larger than $\mathcal{I}(x)$ at a few states – this will be made explicit in our feasibility analysis in Theorem 4.9. The classical CBF definition [3] is less strict than Assumption 1 in that it only asks for the existence of a control value that steers the dynamics to the interior of the safe set or tangentially to it. The stricter requirement here is intended to remove difficulties related to regularity and boundedness of the optimization-based controller [1] that are allowed for by the classical CBF definition. Note also that the constraint in (8) is less restrictive than the condition in Assumption 1. The latter requires the safety condition to be satisfied for all $j \in \mathcal{J}(x)$, while the former requires it only for the dynamics at the state at which the controller is evaluated. As we show below, the reason for assuming more than the mere feasibility of the constraint of (8) is to attain regularity properties for the controller u_{act}.

We now study the extent to which the proposed active-component QP controller ensures safety in discontinuous settings. The abrupt changes in the functions defining the constraints in (8) generally introduce discontinuities in u_{act}. Despite this, the following result characterizes its regularity properties under Assumption 1.

Theorem 4.1 (Regularity Properties of Active-Components QP controller). *If Assumption 1 holds*

with $\tilde{\mathcal{I}}(x) = \mathcal{I}(x)$, then in a neighborhood $\mathcal{N}_{\partial\mathcal{C}}$ of $\partial\mathcal{C}$, the active-components controller $u_{\rm act}$ defined by (8) with any class-κ function α is

(i) feasible,
(ii) single-valued,
(iii) continuous almost everywhere,
(iv) bounded in a bounded neighborhood of every point in $\mathcal{N}_{\partial\mathcal{C}}$, and
(v) yields a Filippov solution to (1) with $u = u_{\rm act}(x)$ from every point in $\mathcal{N}_{\partial\mathcal{C}}$.

PROOF. We first choose the neighborhood $\mathcal{N}_{\partial\mathcal{C}}$. For all $x \in \partial\mathcal{C}$ and all $i \in \mathcal{I}(x)$ and $j \in \mathcal{J}(x)$, $\nabla h_i(y)^\top (f_j(y) + G_j(y)u_x)$ is continuous in y in a neighborhood of x. By Assumption 1, keeping in mind the continuity of α and that $\alpha(h_i(x)) = 0$ for all $i \in \mathcal{I}(x)$ at $x \in \partial\mathcal{C}$, there is a neighborhood $\mathcal{N}_x^{i,j}$ of x such that $\nabla h_i(y)(f_j(y)+G_j(y)u_x)+\alpha(h_i(y)) > 0$ for every $y \in \mathcal{N}_x^{i,j}$. We define then $\mathcal{N}_{\partial\mathcal{C}} = \cup_{x\in\partial\mathcal{C}}(\mathcal{M}_x \cap (\cap_{i,j}\mathcal{N}_x^{i,j})) \cap \mathcal{X}$, where $\mathcal{M}_x$ is given by Lemma A.1(a). We now prove the statements of the result:

(i) Noting that $(f, G) = (f_j, G_j)$ for some $j \in \mathcal{J}(x)$, the constraint in (8) is always satisfied by some u_x for any point x in $\mathcal{N}_{\partial\mathcal{C}}$, proving feasibility.

(ii) For every $x \in \mathcal{N}_{\partial\mathcal{C}}$, the program (8) has a strictly convex objective and a nonempty closed constraint set. Thus the function that equals the objective where the constraints are satisfied and equals ∞ otherwise satisfies the premises of [18, Thm. 1.9]. Thus, the set of minimizers is not empty. By [18, Thm. 2.6] the minimizer is unique.

(iii) Since $\mathcal{I}(x)$ and $\mathcal{J}(x)$ are singletons almost everywhere, for almost every $x \in \mathcal{N}_{\partial\mathcal{C}}$, there is a neighborhood $\mathcal{N}_x$ of x with $\mathcal{N}_x \subseteq \mathcal{N}_{\partial\mathcal{C}}$ such that, for all y in $\mathcal{N}_x$, $(h(y), \nabla h(y)) = (h_i(y), \nabla h_i(y))$ (where i is the only element in $\mathcal{I}(x)$) and $(f(y), G(y)) = (f_j(y), G_j(y))$ (where j is the only element in $\mathcal{J}(x)$). Thus, $\nabla h(x)^\top (f(x) + G(x)u) + \alpha(h(x))$ is continuous in both x and u and convex in u for almost every x. Keeping in mind the convexity of $\mathcal{U}$ and the existence of u_x that strictly satisfies the constraint, the set-valued map $\mathcal{N}_{\partial\mathcal{C}} \ni x \mapsto \{u \mid u \text{ satisfies (8) at } x\} \subset \mathcal{U}$, is continuous in a neighborhood of x by [11, Thms. 10 & 12]. Thus for almost every $x \in \mathcal{N}_{\partial\mathcal{C}}$, $u_{\rm act}$ is continuous at x by [11, Cor. 8.1].

(iv) By definition of $\mathcal{N}_{\partial\mathcal{C}}$, for all $\bar{x} \in \mathcal{N}_{\partial\mathcal{C}}$ there exists $x \in \partial\mathcal{C}$ such that $\nabla h_i(\bar{x})^\top (f_j(\bar{x}) + G_j(\bar{x})u_x) + \alpha(h_i(\bar{x})) > 0$ for all $i \in \mathcal{I}(\bar{x})$ and $j \in \mathcal{J}(\bar{x})$. By continuity of ∇h_i, f_i, and G_i there is a neighborhood $\mathcal{N}$ of $\bar{x}$ such that $\nabla h_i(y)^\top (f_j(y) + G_j(y)u_x) + \alpha(h_i(y)) > 0$ for all $y \in \mathcal{N}$ for all $i \in \mathcal{I}(\bar{x})$ and $j \in \mathcal{J}(\bar{x})$. By Lemma A.1(a), the neighborhood $\mathcal{N}$ can be chosen such that the strict inequality holds for all $i \in \mathcal{I}(y)$ and $j \in \mathcal{J}(y)$. Hence, $u_{\bar{x}}$ is feasible in program (8) for all $y \in \mathcal{N}$, which implies

$$u_{\rm act}(y)^\top Q(y) u_{\rm act}(y) + b(y)^\top u_{\rm act}(y) \le u_{\bar{x}}^\top Q(y) u_{\bar{x}} + b(y)^\top u_{\bar{x}}.$$

Thus, invoking the Lipschitzness of Q and b, and the positive definiteness of $Q(\cdot)$, we deduce that $\|u_{\rm act}\|$ is bounded on a neighborhood of $\bar{x}$.

(v) By the assumed piecewise continuous structure of the dynamics (1) and the fact that $u_{\rm act}$ is continuous almost everywhere and bounded on a neighborhood of every point, this statement is immediate from [6, Prop. 1]. •

Next, we study the safeness properties of the active component QP controller $u_{\rm act}$. The following result shows that trajectories of (1) under $u_{\rm act}$ can be kept from leaving the set from the points at which h is differentiable.

Theorem 4.2 (Safeness of Active-Components QP Controller at Points of Smoothness). *Under the assumptions of Theorem 4.1, let $k : \mathcal{C} \cup \mathcal{N}_{\partial\mathcal{C}} \to \mathcal{U}$ be a controller that gives a solution from any initial condition in $\mathcal{C}$ and is such that $k(x) = u_{\rm act}(x)$ for all $x \in \mathcal{N}_{\partial\mathcal{C}}$. Let $\phi : [t_0, t_1] \to \mathcal{N}_{\partial\mathcal{C}}$ be a Filippov solution of (1) under $u = k(x)$ with $\phi(t_0) \in \mathcal{C}$. If $\phi(t) \notin \mathcal{C}$, for some $t > t_0$ then there exists $\bar{t} \in [t_0, t)$ such that $h(\phi(\bar{t})) = 0$ and $\mathcal{I}(\phi(\bar{t}))$ is not a singleton.*

PROOF. Since $h(\phi(t_0)) \ge 0$ and $h(\phi(t)) < 0$ there exists $\bar{t} \in [t_0, t)$ for which $h(\phi(\bar{t})) = 0$ and $h(\phi(t)) < 0$ for all $t \in (\bar{t}, t)$. We reason by contradiction and suppose that $\mathcal{I}(\phi(\bar{t}))$ is the singleton $\{i\}$. Therefore, h is differentiable on a neighborhood of $\phi(\bar{t})$ according to our assumptions on the structure of h. The gradient of h at $\phi(\bar{t})$ is $\nabla h_i(\phi(\bar{t}))$. We now prove that $\inf \tilde{\mathcal{L}}_{F[f+Gk]} h(x) \ge -\alpha(h(x))$ in a neighborhood of $\phi(\bar{t})$, which by Theorem 3.1 implies that ϕ does not leave $\mathcal{C}$ from $\phi(\bar{t})$ which contradicts the supposition. For any $a \in \tilde{\mathcal{L}}_{F[f+Gk]} h(x)$ where x is sufficiently close to $\phi(\bar{t})$ there is $\zeta \in F[f + Gk](x)$ such that $a = \nabla h_i(x)^\top \zeta$. Since $\zeta \in F[f + Gk](x)$, by (4), there is a finite number say p, of sequences $\{\{x_\mu^1\}, \dots, \{x_\mu^p\}\}$ such that $x_\mu^j \to x$ as $\mu \to \infty$ and $\zeta = \sum_{j=1}^p \sigma_j \lim_{\mu\to\infty}(f(x_\mu^j) + G(x_\mu^j)k(x_\mu^j))$, where σ_j's are the constants of the convex combination. So,

$$\begin{aligned}
\nabla h(x)^\top \zeta &= \nabla h_i(x)^\top \sum_{j=1}^p \sigma_j \lim_{\mu\to\infty} (f(x_\mu^j) + G(x_\mu^j)k(x_\mu^j)) \\
&= \sum_{j=1}^p \sigma_j \lim_{\mu\to\infty} \nabla h_i(x_\mu^j)^\top (f(x_\mu^j) + G(x_\mu^j) u_{\rm act}(x_\mu^j)) \\
&\ge -\sum_{j=1}^p \sigma_j \lim_{\mu\to\infty} \alpha(h_i(x_\mu^j)) = -\alpha(h(x)).
\end{aligned}$$

The equality is due to the continuity of ∇h_i; and the inequality to the constraint in (8) which $u_{\rm act}$ satisfies. Thus in a neighborhood of $\phi(\bar{t})$, $\inf \tilde{\mathcal{L}}_{F[f+Gk]} h(x) \ge -\alpha(h(x))$.•

A consequence of the previous result is that the active-component QP controller is good enough when h is differentiable even if the dynamics is discontinuous.

Corollary 4.3 (Active-Components QP Controller Is Safe for Smooth Set). *Under the assumptions of Theorem 4.1, let $k : \mathcal{C} \cup \mathcal{N}_{\partial\mathcal{C}} \to \mathcal{U}$ be a feedback controller defined as in Theorem 4.2. If $\mathcal{I}(x)$ is a singleton for all $x \in \partial\mathcal{C}$, then k is safe with respect to $\mathcal{C}$.*

Note that Theorem 4.2 leaves open the possibility of the closed-loop system under $u_{\rm act}$ being unsafe at points of nonsmoothness of h. The following example shows that this is indeed possible.

Example 4.4 (Un-safe Active-Components QP Controller). On $\mathbb{R}^2$, let $f(x) = (1, 0)$, $G_1(x) = (-2, 1)$,

$G_2(x) = (-2, -1)$ and consider the discontinuous control-affine dynamics defined by

$$(f(x), G(x)) = \begin{cases} (f(x), G_1(x)), & x_2 < 0, \\ (f(x), G_2(x)), & x_2 > 0. \end{cases}$$

Consider the function $h(x) = \min\{h_1(x), h_2(x)\}$, where $h_1(x) = x_2 - x_1 + 1$ and $h_2(x) = -x_2 - x_1 + 1$. Note that the hypotheses of Theorem 4.1 hold. Taking $Q(x) = 1$, $b = 0$, and $\alpha(r) = r$ in (8) gives the following explicit expression of u_{act} restricted to the region of interest in the state space, $\mathcal{Y} = \{x \mid -x_1 < x_2 < x_1\}$.

$$u_{\text{act}|\mathcal{Y}}(x) = \begin{cases} -\frac{1}{3}(x_2 - x_1), & x_2 < 0, \\ \frac{1}{3}(x_2 + x_1), & x_2 > 0. \end{cases}$$

The closed-loop dynamics is continuous around $x = (1, 0)$ except on the set with $x_2 = 0$, and is given in $\mathcal{Y}$ by

$$\dot{x}_{\mathcal{Y}} = \begin{cases} (1, 0) + (2, -1)\frac{1}{3}(x_2 - x_1), & x_2 < 0, \\ (1, 0) + (-2, -1)\frac{1}{3}(x_2 + x_1), & x_2 > 0. \end{cases}$$

By (4), in a neighborhood of $(1, 0)$, the Filippov set-valued map $F[f + Gu_{\text{act}}](x)$ is the set of all convex combinations of the two cases in this expression. One such vector for points in the neighborhood of $(1, 0)$ with $x_2 = 0$ is

$$\zeta = \frac{1}{2}\left(f(x) + \frac{1}{3}G_1(x)x_1 + f(x) + \frac{1}{3}G_2(x)x_1\right) = \begin{bmatrix} 1 - \frac{2}{3}x_1 \\ 0 \end{bmatrix}$$

However, $\dot{x}_1 = 1 - \frac{2}{3}x_1$, starting from $x(0) = (1, 0)$, has the solution $x_1(t) = 3/2 - 1/2e^{-2t/3}$, which leaves the 0-superlevel set of h. Therefore, u_{act} is not safe with respect to the 0-superlevel set of h. Figure 1 illustrates this. Even though the set of non-smoothness is of measure zero, the problem it brings up cannot be discarded from consideration in the practical implementation of the active component QP controller. This is because the set of non-smoothness may have a backward reachable set (i.e., the set of initial conditions from which trajectories reach it) of positive measure. This is indeed the case here, as one can easily see that, in a neighborhood of $(1, 0)$ in $\mathcal{C}$, the dynamics $\dot{x}_{\mathcal{Y}}$ always point to the right, which inevitably leads the system to reach the boundary in finite time. From there, the system slides to the point $(1, 0)$, where it loses safety. •

Example 4.4 shows that the obvious counterpart of the popular QP controller is not safe with respect to a non-smooth set. The reason for this lack of safety is that the controller enforces a safety constraint only with respect to the active h_i at a specific state and it does not enforce any constraints with respect to other $h_{i'}$'s no matter how near the states of their activity are. This might generate limiting dynamics that are unsafe with respect to them.

Remark 4.5 (Methods of Safety Enforcement at Non-Smooth Points). If a controller is to be safe, it should impose safety constraints with respect to the functions $h_{i'}$'s which are close enough to being active. This can be done in multiple ways. One way, adopted in [21], is to enforce the constraint of (8) for all $i \in \tilde{\mathcal{I}}(x)$, where $\tilde{\mathcal{I}}(x)$ is chosen to include the indices of all $h_{i'}$'s which are close enough to being active. Another way, used in [8], is to enforce a constraint on the Filippov set-valued map rather than on individual active dynamics and enforce the continuity of the controller. As shown in [8], under those two conditions, which are guaranteed to be satisfiable under Assumption 1, safety is attained. Both of those methods require online calculations other than those required by a mere QP: in the former, to find out $h_{i'}$'s which are close enough to being active, and in the latter to calculate the generalized Lie derivative for a differential inclusion or a similar quantity. Moreover, the first method does not produce a continuous controller with continuous dynamics and the synthesis of a continuous controller using the second method is not presented explicitly in [8]. In the following section, we propose a safe controller that overcomes these difficulties. •

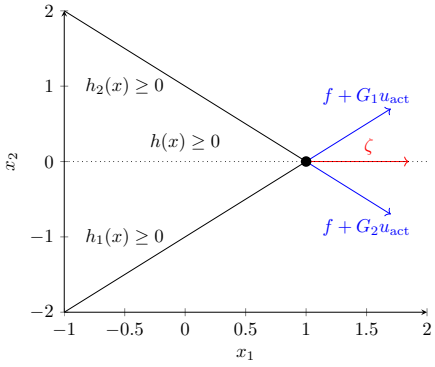


Fig. 1. Illustration for Example 4.4 of the closed-loop dynamics under the controller u_{act} at the corner point of the safe set. The controller prevents trajectories from violating safety constraints at points of smoothness of h, cf. Theorem 4.2, but does not prevent violating safety from points of non-smoothness.

4.2 All-Components QP Controller

In this section, we propose a safe controller that does not require more online calculation than a standard QP and is continuous when the dynamics is continuous. We further prove its existence for all safe sets described as nested unions and intersections of the superlevel sets of the components h_i's. The controller is obtained from a program of the following form

$$u_{\text{all}}(x) \coloneqq \underset{u \in \mathcal{U}}{\arg\min}\, u^\top Q(x)u + b(x)^\top u \tag{9}$$
$$\text{s.t. } \nabla h_i(x)^\top (f(x) + G(x)u) + \alpha(h_i(x)) + \beta_i(H(x)) \geq 0, \quad \forall i \in \{1, \ldots, r\},$$

with the same conditions on Q, b, and α as those in (8), where $H(x) \coloneqq (h_1(x), \ldots, h_r(x))$ and $\beta_i : \mathcal{X} \to \mathbb{R}$ any continuous function with the property that $\beta_i(H(x)) = 0$ when $i \in \mathcal{I}(x)$. We provide an exact form of β_i later when establishing the feasibility of (9). Notice that, unlike (8), the constraints in (9) are enforced for all i, and thus the terminology *all-components QP controller*. This is the central safety-related difference between the active component and the all-component QP controllers. It is this difference that will allow us to prove a safety result stronger than Theorem 4.2. The idea of enforcing safety constraints on all components h_i appears also in [9]. There, however, the structure of the safe set is limited to an intersection of the superlevel sets of the components h_i's and the feasibil-

ity of the program is not studied. Note also that, different from (8) and the controller in [9], the functions β_i appear in the constraints of (9). We refer to these functions as *transition functions* and will choose them later in a manner that facilitates the satisfaction of the constraints for the inactive h_i's. Our first result establishes regularity properties of the controller u_{all} if the program that defines it is feasible.

Theorem 4.6 (Single Valued-ness and Continuity of All-Components QP Controller). *Assume there exists a neighborhood $\mathcal{N}_{\partial\mathcal{C}}$ of $\partial\mathcal{C}$ where (9) is feasible for some α and β_i's. Then, u_{all} is single-valued. Furthermore, if f and G are continuous at $x \in \mathcal{N}_{\partial\mathcal{C}}$ where (9) is strictly feasible, then u_{all} is continuous at x.*

PROOF. Proving single valued-ness follows a similar argument as in the proof of Theorem 4.1(ii). As for continuity, keeping in mind the convexity of $\mathcal{U}$, the strict feasibility of (9), and the continuity of ∇h_i, f, G, α, and β, the set-valued map defined by the constraints of (9) around x is continuous by [11, Thms. 10 & 12]. Thus, u_{all} is continuous at x by [11, Cor. 8.1]. •

Next, we show that u_{all} is safe even at non-smooth points if the program that defines it is feasible.

Theorem 4.7 (All-Components QP Controller Is Safe). *Assume there exists a neighborhood $\mathcal{N}_{\partial\mathcal{C}}$ of $\partial\mathcal{C}$ where (9) is feasible for some α and β_i's. Then, any controller $k : \mathcal{C} \cup \mathcal{N}_{\partial\mathcal{C}} \to \mathcal{U}$ for which there exists a Filippov solution to the closed-loop dynamics from any point in $\mathcal{C}$ and that is equal to u_{all} on $\mathcal{N}_{\partial\mathcal{C}}$ is safe with respect to $\mathcal{C}$.*

PROOF. By (4), if $\zeta \in F[f + Gu_{\text{all}}](x)$, $x \in \mathcal{N}_{\partial\mathcal{C}}$, then there exist p sequences $\{x_\mu^1\}, \dots, \{x_\mu^p\}$ converging to x with

$$\zeta = \sum_{j=1}^{p} \sigma_j \lim_{\mu\to\infty} (f(x_\mu^j) + G(x_\mu^j)u_{\text{all}}(x_\mu^j)),$$

where σ_j's are convex combination constants. Thus, using the continuous differentiability of h_i, $i \in \{1, \dots, r\}$,

$$\nabla h_i(x)^\top \zeta = \sum_{j=1}^{p} \sigma_j \lim_{\mu\to\infty} \nabla h_i(x_\mu^j)^\top (f(x_\mu^j) + G(x_\mu^j)u_{\text{all}}(x_\mu^j)).$$

Using (9) and taking the limit as $\mu \to \infty$, we obtain

$$\begin{aligned}
&\nabla h_i(x_\mu^j)^\top (f(x_\mu^j)+G(x_\mu^j)u_{\text{all}}(x_\mu^j)) \geq -\alpha(h_i(x_\mu^j))-\beta_i(H(x_\mu^j)) \\
\implies &\lim_{\mu\to\infty} \nabla h_i(x_\mu^j)^\top (f(x_\mu^j) + G(x_\mu^j)u_{\text{all}}(x_\mu^j)) \\
&\qquad \geq -\lim_{\mu\to\infty} (\alpha(h_i(x_\mu^j)) + \beta_i(H(x_\mu^j))) \\
&\qquad = -\alpha(h_i(x)) - \beta_i(H(x)).
\end{aligned}$$

Therefore, $\nabla h_i(x)^\top \zeta \geq -\alpha(h_i(x)) - \beta_i(H(x))$ for all $i \in \{1, \dots, r\}$. By (5), for any $v \in \partial h(x)$, $v = \sum_{i\in\mathcal{I}(x)} \bar{\sigma}_i \nabla h_i(x)$, where $\bar{\sigma}_i$'s are convex combination constants. Thus $v^\top \zeta \geq -\sum_{i\in\mathcal{I}(x)} \bar{\sigma}_i(\alpha(h_i(x)) + \beta_i(H(x)))$. Recall that $\beta_i(H(x)) = 0$ when $i \in \mathcal{I}(x)$ and that if $i, i' \in \mathcal{I}(x)$ then $h(x) = h_i(x) = h_{i'}(x)$. Thus, $\sum_{i\in\mathcal{I}(x)} \bar{\sigma}_i(\alpha(h_i(x)) + \beta_i(H(x))) = \alpha(h(x))$, so we have $v^\top \zeta \geq -\alpha(h(x))$. The application of Theorem 3.1 now completes the proof. •

We now turn our attention to establishing the feasibility of the program defining u_{all}. Specifically, we show that, under Assumption 1, there exists a choice of α and β_i's that makes the program in (9) feasible under widely applicable conditions. Those conditions are summarized in the following assumption.

Assumption 2 (Structure of Safe Set). The set $\partial\mathcal{C}$ is compact and $\mathcal{C}$ is given by finitely many (arbitrary) nested unions and intersections of superlevel sets $\mathcal{C}_i := \{x \in \mathbb{R}^n \mid h_i(x) \geq 0\}$ of the continuously differentiable functions $\{h_i\}_{i=1}^r$. We also assume that h_i's are sufficiently different, i.e., for any $i \neq i'$, $h_i(x) \neq h_{i'}(x)$ for almost all $x \in \mathcal{X}$. •

The next result provides a normal form for any safe set that satisfies Assumption 2.

Lemma 4.8 (Normal Form of Safe Set). *Under Assumption 2, $\mathcal{C}$ can be expressed as a union of intersections of collections of $\mathcal{C}_i$'s,*

$$\mathcal{C} = \bigcup_{\ell\in\mathcal{L}} \bigcap_{i\in\mathcal{I}^\ell} \mathcal{C}_i, \tag{10}$$

with $\mathcal{L}$ finite and $\mathcal{I}^\ell \subseteq \{1, \dots, r\}$ for all $\ell \in \mathcal{L}$.

PROOF. Let P_i denote the statement that $x \in \mathcal{C}_i$ and P denote that $x \in \mathcal{C}$. Let '$\vee$' and '$\wedge$' signify logical disjunction and conjunction, respectively. Then the statement that $x \in (\mathcal{C}_{i_1} \cup \mathcal{C}_{i_2})$ corresponds to $P_{i_1} \vee P_{i_2}$, while the statement that $x \in (\mathcal{C}_{i_1} \cap \mathcal{C}_{i_2})$ corresponds to $P_{i_1} \wedge P_{i_2}$. Thus, by definition, P is equivalent to a statement formed by nested applications of conjunction and disjunction on the elementary propositions P_i. From [17, Part C, Central Thm.], the successive distribution of the conjunction over disjunction gives a statement that is equivalent to P and is of the form of disjunction of conjunctions of collections of P_i's. This means that P is equivalent to $\bigvee_{\ell\in\mathcal{L}} \bigwedge_{i\in\mathcal{I}^\ell} P_i$, where $|\mathcal{L}|$ is the number of conjunction collections, and each $\mathcal{I}^\ell$ is a subset of $\{1, \dots, r\}$ containing all i in the j^{th} collection. •

The previous result is not only an existence result but its proof also provides guidance on how to explicitly represent a set $\mathcal{C}$ that satisfies Assumption 2 as a union of intersections. As a consequence of Lemma 4.8, we have that the set $\mathcal{C}$ corresponds to the 0-superlevel set of the function

$$h(x) = \max_{\ell\in\mathcal{L}} \min_{i\in\mathcal{I}^\ell} h_i(x), \tag{11}$$

where h_i is the function that has $\mathcal{C}_i$ as its 0-superlevel set. We establish the feasibility of the program defining u_{all}.

Theorem 4.9 (Feasibility of All-Components QP Controller). *Under Assumption 2, let $h^\ell(x) := \min_{i\in\mathcal{I}^\ell} h_i(x)$ and define $\mathcal{L}_i = \{\ell \in \mathcal{L} \mid i \in \mathcal{I}^\ell\}$. Let Assumption 1 be satisfied with $\tilde{\mathcal{I}}(x) := \{i \in \{1, \dots, r\} \mid \exists \ell \in \mathcal{L}_i,\ h(x) = h^\ell(x) = h_i(x)\}$. Then, for any compact set $\mathcal{D} \subseteq \mathcal{C}$, there exist constants α, $M \in \mathbb{R}$ such that (9) is feasible in a neighborhood of $\mathcal{D}$ with*

$$\alpha(\rho) = \alpha\rho \text{ and } \beta_i(H(x)) := M\big(h(x) - \max_{\ell\in\mathcal{L}_i} h^\ell(x)\big). \tag{12}$$

PROOF. We start by noting that, with the above definition, $\mathcal{I}(x) \subseteq \tilde{\mathcal{I}}(x)$ for all $x \in \mathcal{X}$, cf. Lemma A.2. Let $\mathcal{D} \subseteq \mathcal{C}$ be an arbitrary compact set. For $i \in \tilde{\mathcal{I}}(x)$, we have $h(x) = \max_{\ell\in\mathcal{L}} h^\ell(x) = \max_{\ell\in\mathcal{L}_i} h^\ell(x)$. Thus $\beta_i(H(x)) = 0$ for all $i \in \mathcal{I}(x) \subseteq \tilde{\mathcal{I}}(x)$, ensuring the property of β_i required in program (9). For convenience, we define $\bar{\beta}_\ell(H(x)) = M(h(x) - h^\ell(x))$ for $l \in \mathcal{L}$. Next, we show that there exist α and M such that, for all $x \in \mathcal{D}$, there is $u_x \in \mathcal{U}$ with $\nabla h_i(x)^\top (f(x) + G(x)u_x) + \alpha(h_i(x)) + \bar{\beta}_\ell(H(x)) > 0$, for all $\ell \in \mathcal{L}$ and all $i \in \mathcal{I}^\ell$. This is sufficient for proving the statement since there exists $\ell \in \mathcal{L}_i$ such that $\beta_i(H(x)) = \bar{\beta}_\ell(H(x))$. Define first the indices set-valued map $L(x) \coloneqq \{\ell \in \mathcal{L} \mid h(x) = h^\ell(x)\}$. Our proof has three steps.

Step 1: Here, we prove there exists $\alpha \in \mathbb{R}_{>0}$ such that

$$\forall x\in\mathcal{D},\ \exists u_x\in\mathcal{U},\ \nabla h_i(x)^\top(f_j(x)+G_j(x)u_x)+\alpha h_i(x) > 0,$$

for all $\ell \in L(x)$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x)$. This is sufficient for the satisfaction of all constraints on h_i with $i \in \mathcal{I}^\ell$ and $\ell \in L(x)$. It is even stronger than mere feasibility of the constraint of h_i at x since we ask for the constraint to be satisfied for all $j \in \mathcal{J}(x)$ (we use this in Step 3 of the proof). We reason by contradiction. Suppose that this is not the case. Then, for all $n \in \mathbb{N}$, there exists $x_n \in \mathcal{D}$ such that for all $u_x \in \mathcal{U}$, there are $\ell \in L(x_n)$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x_n)$ with

$$\nabla h_i(x_n)^\top (f_j(x_n) + G_j(x_n)u_x) + nh_i(x_n) \leq 0. \tag{13}$$

Due to the compactness of $\mathcal{D}$, the sequence $\{x_n\}$ has a convergent subsequence [19, Thm. 3.6], say to $\bar{x} \in \mathcal{D}$. Thus, without loss of generality, we choose $\{x_n\}$ in the first place such that it is convergent to $\bar{x}$. Let $I_{\bar{x}} = \{i \in \{1,\dots,r\} \mid \exists \ell \in L(\bar{x}) \text{ s.t. } i \in \mathcal{I}^\ell \text{ and } h_i(\bar{x}) = 0\}$. Note that $I_{\bar{x}} \subseteq \tilde{\mathcal{I}}(\bar{x})$, since, for $\ell \in L(\bar{x})$, $h(\bar{x}) = h^\ell(\bar{x}) = \min_{i\in\mathcal{I}^\ell} h_i(x) \geq 0$, which gives $h(\bar{x}) = h^\ell(\bar{x}) = h_i(\bar{x}) = 0$ for $i \in I_{\bar{x}}$. Thus, by Assumption 1, there is $u_{\bar{x}}$ that validates

$$\nabla h_i(\bar{x})^\top (f_j(\bar{x}) + G_j(\bar{x})u_{\bar{x}}) > 0, \tag{14}$$

for all $i \in I_{\bar{x}}$ and $j \in \mathcal{J}(\bar{x})$. If $I_{\bar{x}}$ is empty, take $u_{\bar{x}}$ as any finite value in $\mathcal{U}$. By (13), with the choice $u_x = u_{\bar{x}}$, for every n there is $\ell_n \in L(x_n) \subseteq \mathcal{L}$, $i_n \in \mathcal{I}^{\ell_n}$ and $j_n \in \mathcal{J}(x_n)$ that validate (13). Since ℓ_n, i_n, and j_n are in finite sets they converge without loss of generality [19, Thm. 3.6], say to $\bar{\ell}, \bar{i}$, and $\bar{j}$ respectively. Hence, for n large enough, we have $\bar{\ell} \in L(x_n)$, $\bar{i} \in \mathcal{I}^{\bar{\ell}}$, $\bar{j} \in \mathcal{J}(x_n)$ and

$$\nabla h_{\bar{i}}(x_n)^\top (f_{\bar{j}}(x_n) + G_{\bar{j}}(x_n)u_{\bar{x}}) + nh_{\bar{i}}(x_n) \leq 0. \tag{15}$$

Because $\bar{\ell} \in L(x_n)$, $h^{\bar{\ell}}(x_n) = \min_{i\in\mathcal{I}^{\bar{\ell}}} h_i(x_n) = h(x_n) \geq 0$. Therefore, we deduce $h_{\bar{i}}(x_n) \geq h^{\bar{\ell}}(x_n) \geq 0$. The continuity of $\nabla h_{\bar{i}}(\cdot)^\top (f_{\bar{j}}(\cdot) + G_{\bar{j}}(\cdot)u_{\bar{x}})$ in a neighborhood of $\bar{x}$ implies that, for sufficiently large n, this expression evaluated at x_n is bounded. This, together with (15) and the fact that $h_{\bar{i}}(x_n) \geq 0$ implies that $h_{\bar{i}}(\bar{x}) = 0$. By Lemma A.1(b), $\bar{\ell} \in L(x_n) \subseteq L(\bar{x})$ and $\bar{j} \in \mathcal{J}(x_n) \subseteq \mathcal{J}(\bar{x})$ and thus $\bar{i} \in I_{\bar{x}}$. Thus (14) with $i = \bar{i}$ and $j = \bar{j}$ contradicts (15).

Step 2: Let α satisfy the statement of Step 1. Here, we prove that there exists $M > 0$ such that, for all $x \in \mathcal{D}$, there is u_x validating $\nabla h_i(x)^\top (f_j(x) + G_j(x)u_x) + \alpha(h_i(x)) + \bar{\beta}_\ell(H(x)) > 0$ for all $\ell \in \mathcal{L}$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x)$. We reason again by contradiction and assume this is not the case. Then, for all n, there exists $x_n \in \mathcal{D}$ such that, for all $u \in \mathcal{U}$, there are $\ell \in \mathcal{L}$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x_n)$ with

$$\nabla h_i(x_n)^\top (f_j(x_n) + G_j(x_n)u) + \alpha(h_i(x_n)) + n(h(x_n) - h^\ell(x_n)) \leq 0. \tag{16}$$

Again, without loss of generality, the sequence x_n converges [19, Thm. 3.6], say to $\bar{x} \in \mathcal{D}$. By Step 1 and the fact that $h(x) \geq h^\ell(x)$ for all $\bar{x} \in \mathcal{D}$, there exists $u_{\bar{x}}$ such that, for all $\ell \in L(\bar{x})$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(\bar{x})$,

$$\nabla h_i(\bar{x})^\top (f_j(\bar{x})+G_j(\bar{x})u_{\bar{x}})+\alpha(h_i(\bar{x}))+n(h(\bar{x})-h^\ell(\bar{x})) > 0.$$

Due to continuity, Lemma A.1(b), and the fact that $h(x) \geq h^\ell(x)$ for all $x \in \mathcal{D}$ and $\ell \in \mathcal{L}$, there is a neighborhood $B_\epsilon(\bar{x})$ such that

$$\nabla h_i(y)^\top (f_j(y)+G_j(y)u_{\bar{x}})+\alpha(h_i(y))+n(h(y)-h^\ell(y)) > 0,$$

for all $n > 0$, $y \in B_\epsilon(\bar{x})$, $\ell \in L(\bar{x})$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(y)$. Thus, for large enough n, there exists $u_{\bar{x}}$ that falsifies (16) for all $\ell \in L(\bar{x})$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x_n)$. For $\ell \notin L(\bar{x})$, $n(h(x_n) - h^\ell(x_n)) \to \infty$. But continuity ensures boundedness of $\nabla h_i(y)^\top (f_j(y)+G_j(y)u_{\bar{x}})+\alpha(h_i(y))$ for $y \in B_\epsilon(\bar{x})$. Therefore, for large enough n, (16) is false for all $\ell \notin L(\bar{x})$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x_n)$. Therefore, we have shown that (16) is falsified for large enough n for all $\ell \in \mathcal{L}$, $i \in \mathcal{I}^\ell$, and $j \in \mathcal{J}(x_n)$, completing the proof of Step 2.

Step 3: In this step, we prove that constraint satisfaction established in Step 2 with α and M is valid on a neighborhood of $\mathcal{D}$. For all points $x \in \partial\mathcal{D}$, Step 2 establishes the existence of u_x that validates the strict inequality constraints of the program (9) defining u_{all} for all $i \in \cup_{\ell\in\mathcal{L}}\mathcal{I}^\ell$ and $j \in \mathcal{J}(x)$. By continuity and Lemma A.1, there exists a neighborhood of x in which u_x validates the constraints.•

Essentially, the proof of Theorem 4.9 consists of showing how each of the three terms, $\nabla h_i^\top Gu$, $\alpha(h_i)$, and $\beta_i(H)$, in the constraints of (9) contribute to ensuring the feasibility of the optimization program. Intuitively, the first term ensures feasibility around the boundary points of $\mathcal{C}$ that are determined by the active h_i, as Assumption 1 ensures. The second term ensures feasibility at points of activity of h_i that are further away from the boundary (and thus, where $h_i > 0$). The last term ensures feasibility in the areas of inactivity of h_i, where $\beta > 0$.

Remark 4.10 (Role of Transition Functions). To understand the role of the transition function β_1, consider the simple case where the safe set is given by a superlevel set $\mathcal{C}_1$ of one differentiable function h_1. The standard CBF constraint (7) for h_1 ensures that $h_1(x(t)) \geq 0$, and hence the forward invariance of $\mathcal{C}_1$. This constraint is altered in our proposed design (9) as

$$\nabla h_1(x)^\top (f(x) + G(x)u) + \alpha(h_1(x)) + \beta_1(x) \geq 0.$$

Note that, under this constraint, system trajectories will not leave $\mathcal{C}_1$ from points $x \in \partial\mathcal{C}_1$ where $\beta_1(x) \leq 0$, cf. Theorem 3.1, whereas such a guarantee does not exist from

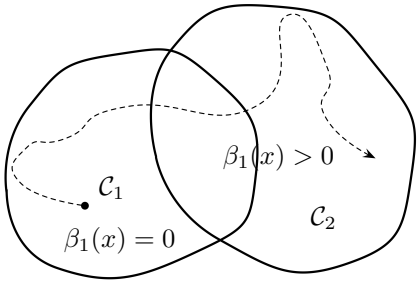


Fig. 2. Illustration of the role of the transition function. The safe set is $\mathcal{C}_1 \cup \mathcal{C}_2$. The continuous function β_1 is positive at the boundary points of $\mathcal{C}_1$ only if they are in $\text{int}(\mathcal{C}_2)$, and zero otherwise in $\mathcal{C}_1$. The constraints in the all-components QP controller u_{all} defined in (9) allow trajectories to leave one set only to the other, while remaining in the safe set.

points $x \in \partial\mathcal{C}_1$ where $\beta_1(x) > 0$. A nonnegative transition function therefore makes it easier to satisfy the constraint while possibly inducing the violation of the forward invariance of $\mathcal{C}_1$ from boundary points where the function is strictly positive. This is of course not desired if the goal is to keep $\mathcal{C}_1$ safe. However, if our safety requirement is keeping a union $\mathcal{C}_1 \cup \mathcal{C}_2$ safe and β_1 is designed to allow the trajectories to leave $\mathcal{C}_1$ from points of the boundary which are in $\mathcal{C}_2$, cf. Figure 2, then the constraint with β_1 provides the flexibility to "transition" from $\mathcal{C}_1$ to $\mathcal{C}_2$. This idea is extended in our design (9) to deal with multiple component sets, where the transition functions are chosen to be positive where it is desirable to allow the trajectory to leave one of those component sets to get to another. •

Remark 4.11 (Properties of the Feasibility Set). From the proof of Theorem 4.9, one can see that if α satisfies the statement of Step 1, then any $\bar{\alpha} > \alpha$ does too. Similarly, if M satisfies Step 2 for some α, then for any $\bar{\alpha} > \alpha$, there exists $\bar{M} > M$ satisfying Step 2. Thus, for any pair (α, M) satisfying Theorem 4.9, there is a pair of greater numbers $(\bar{\alpha}, \bar{M})$ that do too. We leverage this observation below in our approach to find appropriate parameters α and M that ensure feasibility. A larger value of α allows for faster approaches to the boundary of the safe set [3], while a larger M enhances the ability to transition from one component of the safe set to another (cf. Remark 4.10). •

4.3 All-Components Adaptive QP Controller

Theorem 4.9 is not a constructive result, in the sense that it does not provide an explicit way to construct the functions α and β_i's that ensure the feasibility of the program defining u_{all}. We here remedy this by introducing a controller where α and M (which in turn defines the functions β_i's) are taken as optimization variables themselves. Namely, using the fact that $h(\cdot) = \max_{\ell\in\mathcal{L}} \min_{i\in\mathcal{I}^\ell} h_i(\cdot)$ (cf. Lemma 4.8), we propose the following design:

$$
\begin{aligned}
(u_{\text{adp}}(x),&\alpha_{\text{adp}}(x), M_{\text{adp}}(x)) \coloneqq \\
&\underset{\alpha, M, u\in\mathcal{U}}{\arg\min}\ u^\top Q(x)u + b(x)^\top u + q_\alpha \alpha^2 + q_M M^2 \\
&\text{s.t. } M \geq c_M,\ \alpha \geq c_\alpha \\
&\quad \nabla h_i(x)^\top (f(x) + G(x)u) + \alpha h_i(x) \\
&\quad + M(h(x) - h^\ell(x)) \geq 0,\ \forall \ell \in \mathcal{L},\ i \in \mathcal{I}^\ell,
\end{aligned}
\tag{17}
$$

where $c_M, c_\alpha, q_\alpha, q_M$ are positive constants and the rest of the conditions on the objective function are the same as in (8). The positive constants q_α and q_M establish the positive definiteness of the objective function, which ensures the uniqueness of the solver of (17). They also penalize large α and M, and thus establish the existence of an upper bound for α_{adp} and M_{adp} in any compact set. This is needed for establishing control invariance. The positivity of c_α keeps the quantity $\alpha_{\text{adp}}(x)h_i(x)$ greater than the class-κ function $\alpha(h(x)) = c_\alpha h_i(x)$, a fact necessary for establishing control invariance. The positivity of c_M ensures that $M_{\text{adp}}(x)(h(x) - h^\ell(x))$ is positive when $h(x) \neq h^\ell(x)$. It thus promotes violating the safety constraint of the inactive components of h, inducing transitioning between different components of the safe set. Since the values of α and M are chosen adaptively in (17) as a function of the state, we refer to this as the *all-components adaptive QP controller*.

Theorem 4.12 (Properties of All-Components Adaptive QP Controller). *Under the assumptions of Theorem 4.9, in any compact $\mathcal{D} \subseteq \mathcal{C}$, the adaptive all-components QP controller u_{adp} defined by* (17) *is*

(i) feasible,
(ii) single-valued,
(iii) continuous wherever the system dynamics is continuous, and
(iv) safe, i.e., no trajectory can leave $\mathcal{C}$ from a point in $\mathcal{D}$.

PROOF. Statement (i) follows from Theorem 4.9 and Remark 4.11. Statement (ii) can be established with the same argument as in Theorem 4.1(ii) and statement (iii) with the same argument as in Theorem 4.6. Regarding statement (iv), let $\bar{M}$ and $\bar{\alpha}$ be upper bounds of $M_{\text{adp}}(x)$ and $\alpha_{\text{adp}}(x)$ on $\mathcal{D}$. Those upper bounds exist since by Theorem 4.9, finite α and M exist that validate the constraints for all $x \in \mathcal{D}$ at once. Now define the class-κ function $\tilde{\alpha}(r) = \bar{\alpha} r$ for $r \geq 0$ and $\tilde{\alpha}(r) = c_\alpha r$ for $r < 0$. Then, u_{adp} satisfies the constraints of (9) with $\alpha(r) = \tilde{\alpha}(r)$ and $\beta_i(H(x)) = \bar{M}\big(h(x) - \max_{\ell\in\mathcal{L}_i} h^\ell(x)\big)$, with $\mathcal{L}_i$ defined in Theorem 4.9, because, for each $x \in \mathcal{D}$,

$$
\begin{aligned}
\tilde{\alpha}(h(x)) + \bar{M}\big(h(x) - \max_{\ell\in\mathcal{L}_i} h^\ell(x)\big) \geq \\
\alpha_{\text{adp}}(x)h(x) + M_{\text{adp}}(x)\big(h(x) - \max_{\ell\in\mathcal{L}_i} h^\ell(x)\big).
\end{aligned}
$$

With the argument as in Step 3 in the proof of Theorem 4.9, we deduce that the constraints are satisfied with $\tilde{\alpha}$ and $\bar{M}$ on a neighborhood of $\mathcal{D}$. Hence, the same reasoning employed to show Theorem 4.7 ensures u_{adp} is safe. •

The all-components QP controller u_{all} and its adaptive version u_{adp} provide solutions to Problem 1. Given discontinuous dynamics and safety constraints defined by multiple components, we provide controllers that are safe, computable online, and continuous wherever the dynamics is continuous, under standard safety assumptions. We illustrate the versatility of the proposed designs in a multi-agent reconfiguration problem in Section 5 below.

5 Application to Multi-Agent Coverage

In this section, we demonstrate the efficacy of our control design by applying it to a multi-agent coverage control problem. We consider 5 robot agents with positions $x_i = (x_{i,1}, x_{i,2})$, $i \in \{1, \ldots, 5\}$. Define the full state as

$x = (x_1, \ldots, x_5)$. The coverage control objective consists of having the agents spread as far a way from each other as possible within the obstacle-free space. The obstacle-free space, denoted $\mathcal{C}_f$, is the nonconvex non-gray space shown in Figure 3. This set is a union of two rectangles, each of which is an intersection of half-spaces. The dynamics of each agent follow the rule "move away from the nearest neighbor". Formally, we set $\dot{x}_i = \frac{x_i - \mathcal{N}(x_i)}{\|x_i - \mathcal{N}(x_i)\|} + u_i$, where $\mathcal{N}(x_i) = \arg\min_{\{x_j, j \neq i\}} \|x_i - x_j\|$ is the position of the closest agent to x_i. Since its identity might change along the evolution, the resulting dynamical system is discontinuous [6]. This dynamics naturally prevent the agents from colliding, so our safety requirements are merely staying within the obstacle-free space. Figure 3 shows the safe agents' trajectories under u_{adp} with choices $c_\alpha = 1$, $c_M = 10$, and $q_\alpha = q_M = 0.1$. The rightmost plots further show the evolution of the terms of αh^j's and β_j's in the constraints of (17). Note that, when h^1 becomes negative, and thus is unable to support the satisfaction of its respective constraint, β_1 carries the load of validating it.

6 Conclusions

We designed controllers that render discontinuous dynamics forward invariant with respect to non-smooth sets. The safe set is represented by arbitrarily nested unions and intersections of 0-superlevel sets, termed components, of differentiable functions, resulting in a nonsmooth CBF. We showed that the satisfaction of the safety condition for the active components of the non-smooth CBF does not render the discontinuous dynamics safe. We remediated this problem by enforcing a novel version of the safety constraints for all the components of the non-smooth CBF that incorporates transition functions. These functions allow system trajectories to leave a component of the safe set to transition to a different one. The resulting all-components QP controller is feasible, safe, and continuous wherever the system dynamics is continuous. To enhance its implementability, we proposed an extension termed all-components adaptive QP controller which determines important design parameters in an adaptive fashion. Our results suggest the possibility of combining multiple safe set design methods to achieve control objectives that might not be achievable with one method alone. Future work will identify conditions to ensure that the assumption about the existence of safe control actions holds for general system classes, extend our controller design method to deal with time-varying safety requirements, and exploit the flexibility of transitioning between components of the safe set to allow richer specifications combining safety, motion planning, and control.

References

[1] M. Alyaseen, N. Atanasov, and J. Cortés. Continuity and boundedness of minimum-norm CBF-safe controllers. *IEEE Transactions on Automatic Control*, 70(6), 2025. To appear.

[2] 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.

[3] 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.

[4] A. Bacciotti and F. Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 4:361–376, 1999.

[5] F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, 1990.

[6] J. Cortés. Discontinuous dynamical systems - a tutorial on solutions, nonsmooth analysis, and stability. *IEEE Control Systems*, 28(3):36–73, 2008.

[7] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides: Control Systems*, volume 18. Springer Science & Business Media, 2013.

[8] 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.

[9] 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.

[10] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252, 2002.

[11] W. W. Hogan. Point-to-set maps in mathematical programming. *SIAM Review*, 15(3):591–603, 1973.

[12] A. Isaly, M. Ghanbarpour, R. G. Sanfelice, and W. E. Dixon. On the feasibility and continuity of feedback controllers defined by multiple control barrier functions. *IEEE Transactions on Automatic Control*, 2024. To appear.

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

[14] L. Lindemann and D. V. Dimarogonas. Barrier function based collaborative control of multiple robots under signal temporal logic tasks. *IEEE Transactions on Control of Network Systems*, 7(4):1916–1928, 2020.

[15] M. Maghenem and R. G. Sanfelice. Sufficient conditions for forward invariance and contractivity in hybrid inclusions using barrier functions. *Automatica*, 124:109328, 2021.

[16] P. Ong, B. Capelli, L. Sabattini, and J. Cortés. Nonsmooth control barrier function design of continuous constraints for network connectivity maintenance. *Automatica*, 156: 111209, 2023.

[17] E. L. Post. Introduction to a general theory of elementary propositions. *American Journal of Mathematics*, 43(3): 163–185, 1921.

[18] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*, volume 317 of *Comprehensive Studies in Mathematics*. Springer, New York, 1998.

[19] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

[20] J. Usevitch, K. Garg, and D. Panagou. Strong invariance using control barrier functions: a Clarke tangent cone approach. In *IEEE Conf. on Decision and Control*, pages 2044–2049, Jeju Island, Republic of Korea, December 2020.

[21] M. Vahs and J. Tumova. Non-smooth control barrier functions for stochastic dynamical systems. *arXiv preprint arXiv:2309.06494*, 2023.

[22] A. Wiltz and D. V. Dimarogonas. Handling disjunctions in signal temporal logic based control through nonsmooth barrier functions. In *IEEE Conf. on Decision and Control*, pages 3237–3242, Cancún, Mexico, 2022.

[23] W. Xiao, C. G. Cassandras, and C. Belta. *Safe Autonomy with Control Barrier Functions: Theory and Applications*. Synthesis Lectures on Computer Science. Springer, New York, 2023.

[24] A. Zehfroosh and H. G. Tanner. Non-smooth control barrier navigation functions for STL motion planning. *Frontiers in Robotics and AI*, 9:782783, 2022.

A Supplementary Results on Set-Valued Maps

Here we prove the upper semi-continuity of the activity set-valued maps $\mathcal{I}$ and $\mathcal{J}$ defined in Section 2.

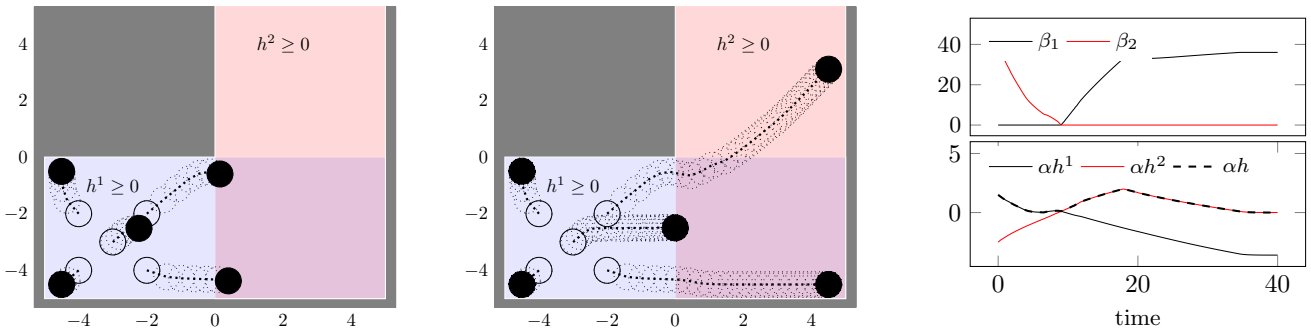


Fig. 3. Illustration of the adaptive all-components controller u_{adp} acting in a multi-agent coverage problem. The two left plots show different snapshots of the agents evolution. The union of light red and blue areas is the safe area, each of which is associated with a respective h^j. The rightmost plots show the evolution of the terms αh^j and β_j appearing in the constraints of u_{adp} for the upper-right agent. The agents ability to safely switch from the safe space determined by h^1 to that determined by h_2 is due to the transition functions, β_j's. As the figure shows, β_j helps in the satisfaction of h^j where it is not active for safety (cf. Remark 4.10).

Lemma A.1 (Upper Semi-continuity of Activity Set-Valued Maps). *Consider the structure of the dynamics* (1) *and the function* h *in* (2).

(a) For all $x \in \mathcal{X}$*, there is a neighborhood* $\mathcal{M}_x$ *such that* $\mathcal{I}(y) \subseteq \mathcal{I}(x)$ *and* $\mathcal{J}(y) \subseteq \mathcal{J}(x)$*, for all* $y \in \mathcal{M}_x$*.*

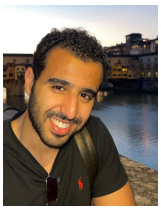
(b) Given the normal form of h *in* (11)*, let* $L(x) \coloneqq \{\ell \in \mathcal{L} \mid h(x) = h^\ell(x)\}$*. Then, the neighborhood* $\mathcal{M}_x$ *can be taken such that* $L(y) \subset L(x)$ *for all* $y \in \mathcal{M}_x$*.*

PROOF. (a) Let $x \in \mathcal{X}$. If $i' \notin \mathcal{I}(x)$, then $x \notin \text{cl}\big(\mathcal{X}^{h_{i'}}\big)$ by definition. Thus, there exists a neighborhood $B_\epsilon(x)$ such that $y \notin \text{cl}\big(\mathcal{X}^{h_{i'}}\big)$ for all $y \in B_\epsilon(x)$. An analogous argument shows that if $j' \notin \mathcal{J}(x)$, then there exists a neighborhood $B_{\epsilon'}(x)$ such that $y \notin \text{cl}(\mathcal{X}_j)$ for all $y \in B_{\epsilon'}(x)$. Taking $\mathcal{M}_x$ as the finite intersection of these neighborhoods for all $i' \notin \mathcal{I}(x)$ and $j' \notin \mathcal{J}(x)$ proves the statement.

(b) If $\ell \notin L(x)$, then $h^\ell(x) \neq h(x)$. By continuity of h and h^ℓ, there exists a neighborhood where $h(y) \neq h^\ell(y)$ for every point y in that neighborhood. Taking the intersection of this finite number of neighborhoods for all $\ell \notin L(x)$ proves the statement. •

Lemma A.2 (Upper Approximation of Activity Set). *Under Assumption 2, defining the set-valued map* $\tilde{\mathcal{I}}(x)$ *as in Theorem 4.9 yields* $\mathcal{I}(x) \subseteq \tilde{\mathcal{I}}(x)$ *for all* $x \in \mathcal{X}$*.*

PROOF. We reason by contradiction. Assume there exists $x \in \mathcal{X}$ and $i' \in \{1, \dots, r\}$ such that $i' \in \mathcal{I}(x)$ and $i' \notin \tilde{\mathcal{I}}(x)$. Thus, $h_{i'}(x) = h(x)$ and for all $\ell' \in \mathcal{L}_{i'}$, $h^{\ell'}(x) \neq h(x)$. By continuity, there exists a neighborhood $B_\epsilon(x)$ where $h^{\ell'}(y) \neq h(y)$ for all $y \in B_\epsilon(x)$ and all $\ell' \in \mathcal{L}_{i'}$. Since $x \in \text{cl}\big(\mathcal{X}^{h_{i'}}\big)$, $B_\epsilon(x) \cap \mathcal{X}^{h_{i'}}$ is not empty and open (since it is an intersection of two open sets), and thus has positive measure. Since for each $y \in B_\epsilon(x) \cap \mathcal{X}^{h_{i'}}$, $h(y) = h^\ell(y)$ for some $\ell \notin \mathcal{L}_{i'}$, there is $i \neq i'$ such that $h(y) = h_i(y)$. At the same time, by definition of $\mathcal{X}^{h_{i'}}$, $h(y) = h_{i'}(y)$ in $B_\epsilon(x) \cap \mathcal{X}^{h_{i'}}$. Consequently, and given that the set $\{1, \dots, r\}$ is finite, there exists $i \neq i'$ such that $h_i(y) = h_{i'}(y)$ on a set of positive measure. This contradicts Assumption 2.



Mohammed Alyaseen received his Bachelor's in Mechanical Engineering from Kuwait University. He then received his master's degree in Mechanical Engineering as well from the University of California San Diego (UCSD), respectively. He is now a PhD candidate in UCSD under the supervision of Jorge Cortés. His research interests include safety critical control, non-smooth analysis, and optimization based control design.



Nikolay Atanasov is an Associate Professor of Electrical and Computer Engineering at the University of California San Diego. He obtained a B.S. degree in Electrical Engineering from Trinity College, Hartford, CT, in 2008 and M.S. and Ph.D. degrees in Electrical and Systems Engineering from the University of Pennsylvania, Philadelphia, PA, in 2012 and 2015, resp. His research focuses on robotics, control theory, and machine learning with emphasis on active perception problems for autonomous mobile robots. His work has been recognized by the Joseph and Rosaline Wolf award for the best Ph.D. dissertation in Electrical and Systems Engineering at the University of Pennsylvania in 2015, the Best Conference Paper Award at ICRA in 2017, the NSF CAREER Award in 2021, and the IEEE RAS Early Academic Career Award in Robotics and Automation in 2023.



Jorge Cortés received the Licenciatura degree in mathematics from Universidad de Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from Universidad Carlos III de Madrid, Spain, in 2001. He held postdoctoral positions with the University of Twente, The Netherlands, and the University of Illinois at Urbana-Champaign, USA. He is a Professor in the Department of Mechanical and Aerospace Engineering, UC San Diego, USA. He is a Fellow of IEEE, SIAM, and IFAC. His research interests include distributed control and optimization, network science, nonsmooth analysis, reasoning and decision making under uncertainty, network neuroscience, and multi-agent coordination in robotic, power, and transportation networks.