IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 4, NO. 2, APRIL 2019

Hybrid Nonsmooth Barrier Functions With Applications to Provably Safe and Composable Collision Avoidance for Robotic Systems

Paul Glotfelter, Ian Buckley, and Magnus Egerstedt

Abstract—**Robots are entering an age of ubiquity, and to operate effectively, these systems must typically satisfy a series of constraints (e.g., collision avoidance, obeying speed limits, maintaining connectivity). In addition, modern applications hinge on the completion of particular tasks, such as driving to a certain location or monitoring a crop patch. The dichotomy between satisfying constraints and completing objectives creates a need for constraint-satisfaction frameworks that are composable with a pre-existing primary objective. Barrier functions have recently emerged as a practical and the composable method for constraint satisfaction, and prior results demonstrate a system of Boolean logic for nonsmooth barrier functions as well as a composable controller-synthesis framework; however, this prior work does not consider dynamically changing constraints (e.g., a robot sensing and avoiding an obstacle). Consequently, the main theoretical contribution of this letter extends nonsmooth barrier functions to time-varying barrier functions with jumps. In a practical instantiation of the theoretical main results, this letter revisits a classic problem by formulating a collision-avoidance framework and composing it with a nominal controller. Experimental results show the efficacy of this framework on a light detection and ranging (LIDAR)-equipped differential-drive robot in a real-time obstacle-avoidance scenario.**

Index Terms—**Optimization and optimal control, robot safety, autonomous vehicle navigation.**

I. Introduction

THE introduction of robots to new application domains, such as precision agriculture and autonomous transportation, creates a new set of associated challenges [1], [2]. In these domains, truly independent operation is predicated on the satisfaction of certain constraints such as avoiding collisions, occupying a designated area, or maintaining connectivity to a base station. Often, these constraints are safety critical, particularly in dynamic, human-occupied spaces, so provable constraint-satisfaction guarantees are valuable [3]. Moreover, most applications rely on the completion of an ever-changing primary objective, such as driving a passenger to various destinations or monitoring different crop patches, which induces varying constraints. For robots to thrive in these domains, they require constraint-satisfaction frameworks that can synthesize safe controllers for various objectives.

Barrier functions have recently reemerged as a provably correct constraint-satisfaction method for robotic systems and have been applied to a variety of scenarios, including autonomous transportation, remote-access testbeds, temporal-logic specifications, and quadrotor teams [3]–[9]. When utilized as constraints in an optimization program, barrier functions enable a composable controller-synthesis framework that can minimally modify a primary controller to satisfy the specified constraints. Moreover, some of this prior work has focused on Boolean composition of smooth barrier functions [6], [9], [10]; however, smooth barrier functions are limited because Boolean composition of set-based constraints typically involves set intersections and unions, which introduces nonsmoothness. Resolving this issue, recent results have extended barrier functions to Nonsmooth Barrier Functions (NBFs) and developed a complete system of Boolean logic through $\min$ and $\max$ operators [11]. Moreover, other work has addressed NBF-based controller synthesis [12].

Prior approaches to set invariance in robotics include [13]–[15]; the use of barrier functions distinguishes this letter from the existing literature. In addition to provable guarantees for forward set invariance, barrier functions offer a number of advantages, namely their implicit formulation for general dynamical systems and composability. The implicit nature of barrier functions relates to Lyapunov functions in that satisfying a particular inequality point-wise across the state space provides a global set-invariance result, meaning that barrier-function-based controller-synthesis frameworks do not necessarily require simulation of the system over an interval (e.g., as in model predictive control). The computational significance of this is demonstrated in [8], where such a framework is used to synthesize controllers for 80-dimensional ensemble systems of differential-drive robots at 100 Hz. Furthermore, barrier functions are composable, which means that arbitrary controllers, such as those generated by the methods of [14], [16], can be easily incorporated during barrier-function-based controller synthesis. For further advantages of barrier functions, see [3], [5]–[7], [11].

As for this letter's contribution, prior work on NBFs, particularly in [11], [12], does not address time-varying NBFs, inhibiting their application in robotics scenarios where unexpected events and dynamic environments may induce changing objectives and constraints (e.g., an autonomous vehicle avoiding a previously unsensed obstacle). Accordingly, the main contribution of this work formulates Hybrid NBFs (HNBFs): time-varying NBFs with jumps. This work also addresses a practically useful class of controlled HNBFs and presents a quadratic-program-based controller-synthesis method with the

Manuscript received September 10, 2018; accepted December 21, 2018. Date of publication January 24, 2019; date of current version February 15, 2019. This letter was recommended for publication by Associate Editor A. Dietrich and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported by the U.S. National Science Foundation under Grant 1531195.
(Corresponding author: Paul Glotfelter.)

The authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30318 USA (e-mail: paul.glotfelter; ihbuckl@gatech.edu; magnus@gatech.edu).
Digital Object Identifier 10.1109/LRA.2019.2895125

aforementioned advantages. HNBFs are superficially similar in their hybrid nature to the switching sequences presented in [17], [18] and much other work on hybrid systems. However, the analysis in this letter differs by focusing on set invariance rather than stability.

Because of their recent renewed interest, the modern formulation of barrier functions has not been widely applied to classic robotics problems. Accordingly, to demonstrate a practical application of HNBFs in robotics, this letter revisits an established problem: collision-avoidance for arbitrary primary objectives. Specifically, HNBFs are used to encode a collision-avoidance algorithm that composes with the primary controller of a light detection and ranging (LIDAR)-equipped differential-drive robot.

This letter is organized as follows. Sec. II contains background material, including tools from [11], [12], [19]–[22]. Using these tools, the main theoretical results in Sec. III extend NBFs to HNBFs and applies them to controlled systems. For use in the subsequent experiment, a general measurement model and a composable collision-avoidance framework for control-affine systems are formulated. To demonstrate the proposed method, Sec. IV instantiates this framework for a LIDAR-equipped differential-drive robot in local coordinates, and in Sec. V, the robot uses the framework to achieve collision-free navigation. Sec. VI concludes the letter.

II. Background Material

The proposed HNBFs require tools from nonsmooth analysis and the theory of discontinuous dynamical systems. As such, this section introduces background material including: notation, system of interest, discontinuous dynamical systems, nonsmooth analysis, Boolean logic for barrier functions, and controller synthesis for NBFs.

A. Notation

The symbol $\times$ stands for the Cartesian product. The notation $\mathbb{R}_{\geq a}$ represents the set of nonnegative real numbers greater or equal to a; $a.e.$ denotes almost everywhere in the sense of Lebesgue measure. The expression $B(x', \delta)$ denotes an open ball of radius δ centered on a point $x' \in \mathbb{R}^n$. The operation co represents the convex hull of a set. A function $\alpha : \mathbb{R} \to \mathbb{R}$ is extended class-$\mathcal{K}$ if α is continuous, strictly increasing, and $\alpha(0) = 0$.

B. Control-Affine Systems

Many robots can be modelled as control-affine systems (e.g., differential-drive robots, quadrotors) of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t), t), x(t_0) = x_0, t_0 \in \mathbb{R}, \quad (1)$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^m$ are locally Lipschitz continuous; $u : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ is measurable and locally bounded in both arguments. In this case, u is not necessarily continuous, so these assumptions model discontinuities in the control input, which inevitably occur during controller synthesis in the context of this work. For additional information and definitions (e.g., locally bounded), see [22, p. 44, p. 49].

For differential equations, such as (1), solutions may not exist. Toward analyzing discontinuous differential equations, applying Filippov's operator $K[f + gu] : \mathbb{R}^n \times \mathbb{R} \to 2^{\mathbb{R}^n}$ yields a particular system to which solutions exist. This operator maps discontinuous differential equations, as in (1), to a solution-bearing differential inclusion. At a point (x', t'),

$$K[f + gu](x', t') = \text{co}\Big\{ \lim_{i\to\infty} f(x_i) + g(x_i)u(x_i, t') : x_i \to x', x_i \notin N_f, N \Big\}, \quad (2)$$

where N_f is a particular zero-Lebesgue-measure set and N is an arbitrary zero-Lebesgue-measure set [20, Thm. 1].

For a general differential inclusion,

$$\dot{x}(t) \in F(x(t), t),\ x(t_0) = x_0, t_0 \in \mathbb{R}, \quad (3)$$

on a set-valued map $F : \mathbb{R}^n \times \mathbb{R} \to 2^{\mathbb{R}^n}$, a Carathéodory solution is an absolutely continuous function $x : [t_0, t_1] \to \mathbb{R}^n$ such that $\dot{x}(t) \in F(x(t), t)$ almost everywhere on $[t_0, t_1] \ni t$ and $x(t_0) = x_0$. Such solutions exist if F is compact-, convex-valued, nonempty, and locally bounded; for each fixed t', the function $x' \mapsto F(x', t')$ is upper semi-continuous; and for each fixed x', the function $t' \mapsto F(x', t')$ is measurable [22, p. 44, p. 49]. In terms of completeness of solutions, this work requires that solutions exist but does not require that they exist for all time.

Combining these results, Filippov's operator in (2) automatically satisfies these properties. That is, Carathéodory solutions to the differential inclusion

$$\dot{x}(t) \in K[f + gu](x(t), t), x(t_0) = x_0, t_0 \in \mathbb{R}, \quad (4)$$

always exist. See [22] for a comprehensive coverage of discontinuous dynamical systems.

This work provides results for the general differential inclusion in (3) as they apply to the dynamical system in (1) via (2). In practice, the explicit computation of Filippov's operator is not required; instead, theoretical validation eliminates the need to calculate (4).

C. Nonsmooth Barrier Functions and Boolean Logic

Barrier functions provably guarantee forward invariance of a set that captures domain-specific constraints (e.g., an autonomous vehicle staying in a lane or avoiding collisions). Denote $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ as the barrier function. The goal becomes to ensure forward invariance of the set

$$\mathcal{C} = \{(x', t') \in \mathbb{R}^n \times \mathbb{R}_{\geq t_0} : h(x', t') \geq 0\}.$$

It turns out that forward invariance can be guaranteed when

$$\dot{h}(x(t), t) \geq -\alpha(h(x(t), t), a.e.[t_0, t_1] \ni t, \quad (5)$$

for every Carathéodory solution, for some locally Lipschitz extended class-$\mathcal{K}$ function $\alpha : \mathbb{R} \to \mathbb{R}$ [11].

However, if the barrier function is not continuously differentiable but is locally Lipschitz, as in this letter, calculating $\dot{h}$ can no longer be done via the usual chain rule. Fortunately, the generalized gradient of [19] can be employed. At a point $(x', t') \in \mathbb{R}^n \times \mathbb{R}_{>t_0}$, the generalized gradient $\partial h : \mathbb{R}^n \times \mathbb{R}_{>t_0} \to 2^{\mathbb{R}^n \times \mathbb{R}}$ is defined as

$$\partial h(x', t') = \text{co}\Big\{ \lim_{i\to\infty} \nabla h(x_i, t_i) : (x_i, t_i) \to (x', t'), (x_i, t_i) \notin \Omega, \Omega_h \Big\}, \quad (6)$$

where Ω_h designates the zero-measure set on which h is non-differentiable, Ω is any zero-Lebesgue-measure set, and ∇h is the usual gradient [19, Thm. 2.5.1].

The generalized gradient in (6) is useful because it provides analysis of derivatives along Carathéodory solutions to a differential inclusion. Specifically, given a Carathéodory solution to (3), the time derivative of the function $t \mapsto h(x(t), t)$, satisfies

$$\dot{h}(x(t), t) \geq \min \partial h(x(t), t)^\top (F(x(t)) \times 1) \tag{7}$$

almost everywhere on $[t_0, t_1] \ni t$ [21], [23]. Importantly, the inequality in (7) circumvents the calculation of $\dot{h}(x(t), t)$, which is particularly useful for controller synthesis. Now, combining (5) with (7) yields the desired forward-invariance result [11]. Note that for sets $A, B \subset \mathbb{R}^n$, $A^\top B = \{a^\top b : a \in A, b \in B\}$, such as in (7).

NBFs inherently capture Boolean composition. That is, barrier functions composed with $\wedge$, $\vee$, and $\neg$ operators. For example, an autonomous vehicle must stay in the designated lane and not violate the speed limit. The results of [11], [12] show that max, min, and negation form a full system of Boolean logic with logical operators defined as

$$\begin{aligned} h_1 \wedge h_2 &:= \min\{h_1, h_2\} \\ h_1 \vee h_2 &:= \max\{h_1, h_2\} \\ \neg h_1 &:= -h_1. \end{aligned} \tag{8}$$

Note that the logical operations are point-wise applied. For example, $(h_1 \wedge h_2)(\cdot) = \min\{h_1(\cdot), h_2(\cdot)\}$. This system of logic can generate complex constraints for robotic systems from basic component functions. Moreover, the generalized gradient becomes straightforward to estimate, and the corresponding controller-synthesis framework only requires knowledge of the component functions.

The generalized gradient of such an NBF, at $(x', t') \in \mathbb{R}^n \times \mathbb{R}_{>t_0}$, satisfies the inclusion

$$\partial h(x', t') \subset \text{co} \bigcup_{i \in I(x', t')} \partial h_i(x', t'),$$

where $I(x', t') = \{i : h_i(x', t') = h(x', t')\}$ is the active-function map [19, Prop. 2.3.12]. That is, the convex hull of the generalized gradients of the active component functions (i.e., those currently attaining the max or min) encapsulates the generalized gradient of the NBF.

In [12], the NBF framework is extended to controller synthesis for Boolean compositions. In particular, the work in [12] develops an almost-active gradient for specific NBFs, as in (8), that yields a validating controller when included as a constraint to a QP. Moreover, Filippov's operator on the controller does not have to be explicitly computed, and, despite the required analysis, the controller-synthesis framework can, effectively, be automatically applied.

In this case, the continuous differentiability of the component functions becomes a key property. The almost-active set for a so-called smoothly composed NBF, as in (8), is defined, at $(x', t') \in \mathbb{R}^n \times \mathbb{R}_{>t_0}$, as

$$I_\epsilon(x', t') = \{i : |h_i(x', t') - h(x', t')| \leq \epsilon\}.$$

Similarly, the almost-active generalized gradient is

$$\partial_\epsilon h(x', t') = \text{co} \bigcup_{i \in I_\epsilon(x', t')} \partial h_i(x', t').$$

For such smoothly composed NBFs, [11] shows that including the almost-active generalized gradient as a constraint in an optimization program produces a potentially discontinuous but validating control input, assuming that the resulting controller is measurable and locally bounded [12, Thm. 3].

III. Main Results

This section contains the main results of this work: the formulation of HNBFs, Control HNBFs (CHNBFs), and piece-wise-constant CHNBFs, with the appendix containing proofs of the theorems. The piece-wise-constant CHNBFs are particularly useful for applications with varying constraints. To demonstrate their applicability, this section also formulates a general-purpose collision-avoidance and controller-synthesis framework for the system in (1) in terms of a piece-wise-constant CHNBF that represents dynamically appearing obstacles. Despite the breadth of analysis required to prove the theorems, this framework essentially depends on solving a QP with distance measurements as constraints, and it is simple to apply, effective, and computationally feasible.

A. Hybrid Nonsmooth Barrier Functions

To account for the time-varying nature of the problem as well as potential jumps, the formulation of HNBFs first requires a modification to the concept of forward invariance, which is captured by the following two definitions.

Definition 1: A sequence $\{\tau_k\}_{k=1}^{\infty}$ is a *switching sequence* for (3) if and only if it is strictly increasing, unbounded, and $\tau_1 = t_0$. With respect to $\{\tau_k\}_{k=1}^{\infty}$, let

$$K_{t_1} = \inf\left\{K \in \mathbb{N} : [t_0, t_1] \subset \bigcup_{k=1}^{K} [\tau_k, \tau_{k+1})\right. .$$

Definition 2: A set $D \subset \mathbb{R}^n \times \mathbb{R}$ is *hybrid forward invariant* with respect to (3) and a switching sequence $\{\tau_k\}_{k=1}^{\infty}$ for (3) if and only if for every Carathéodory solution starting from x_0 at t_0,

$$(x(\tau_k), \tau_k) \in D, \forall k \leq K_{t_1} \Rightarrow (x(t), t) \in D,$$

$\forall t \in [t_0, t_1]$.

Def. 2 captures the desired behavior of the system. Informally, if the system starts in the set and every jump remains within the set, then the system cannot leave the set. With regards to composable collision avoidance as presented in Sec. III-B, this assumption implies that no obstacles instantaneously appear too close to the robot. The following two definitions construct candidate HNBFs as well as the conditions under which they are valid.

Definition 3: A function $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ is a *candidate Hybrid Nonsmooth Barrier Function (HNBF)* for (3) if and only if there exists a switching sequence $\{\tau_k\}_{k=1}^{\infty}$ for (3) such that, for every $k \in \mathbb{N}$, h is locally Lipschitz continuous on $\mathbb{R}^n \times [\tau_k, \tau_{k+1})$.

Remark 1: In the context Def. 3, the generalized gradient of h is only defined on each open interval (τ_k, τ_{k+1}), as it requires that the function be defined on an open set; though, local Lipschitz continuity holds on $[\tau_k, \tau_{k+1})$.

Definition 4: A candidate HNBF $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ for (3) by the switching sequence $\{\tau_k\}_{k=1}^{\infty}$ is a *valid HNBF* for (3) if and only if the set

$$\mathcal{C} = \{(x', t') \in \mathbb{R}^n \times \mathbb{R}_{\geq t_0} : h(x', t') \geq 0\}$$

is hybrid forward invariant via (3), $\{\tau_k\}_{k=1}^{\infty}$.

Def. 4 departs from the formulation of NBFs in [11, Def. 4]. Instead, Def. 4 only requires positivity of the function $t \mapsto h(x(t), t)$ and does not require $\mathcal{C}$ to be nonempty, which has been recognized to be superfluous since the forward-invariance statement is an implication. The robustness properties of asymptotic stability to C are precluded by the potential jumps in the system, and recovery of this property is left to future work. In this work, the set $\mathcal{C}$ still remains attractive; however, the system will not be able to reach C asymptotically due to potential jumps. Regardless, the following results utilize the above definitions to guarantee hybrid forward invariance.

Theorem 1: Let $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ be a candidate HNBF for (3) under the switching sequence $\{\tau_k\}_{k=1}^{\infty}$. If for every $k \in \mathbb{N}$, there exists a locally Lipschitz extended class-$\mathcal{K}$ function $\alpha^k : \mathbb{R} \to \mathbb{R}$ such that

$$\min \partial h(x', t')^\top (F(x', t') \times 1) \geq -\alpha^k(h(x', t')),$$

$\forall x' \in \mathbb{R}^n, t' \in (\tau_k, \tau_{k+1})$, then h is a valid HNBF for (3).

Following from Thm. 1, the next results concern a class of piece-wise constant HNBFs for controlled systems, and the subsequent section utilizes them to encode a series of collision-avoidance constraints that stem from range and bearing measurements. Toward this end, the following definition and result capture the notion of a CHNBF. By definition, valid CHNBFs for (1) are valid HNBFs for (4).

Definition 5: A candidate HNBF $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ for (1) under the switching sequence $\{\tau_k\}_{k=1}^{\infty}$ is a valid *Control HNBF (CHNBF)* for (1) if and only if there exists a measurable and locally bounded function $u : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ such that for every $k \in \mathbb{N}$ there exists a locally Lipschitz extended class-$\mathcal{K}$ function $\alpha^k : \mathbb{R} \to \mathbb{R}$ satisfying

$$\min \partial h(x', t')^\top (K[f + gu](x', t') \times 1) \geq -\alpha^k(h(x', t')),$$

$\forall x' \in \mathbb{R}^n, t' \in [\tau_k, \tau_{k+1})$.

Theorem 2: Let $h^k : \mathbb{R}^n \to \mathbb{R}$, $k \in \mathbb{N}$, be a collection of valid CNBFs, in the sense of [12, Def. 4], for (1) under the control laws and extended class-$\mathcal{K}$ functions, $u^k : \mathbb{R}^n \to \mathbb{R}$, $\alpha^k : \mathbb{R} \to \mathbb{R}, k \in \mathbb{N}$; and let $\{\tau_k\}_{k=1}^{\infty}$ be a switching sequence for (1). Then, the function $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ defined as

$$h(x', t') = h^k(x'), \forall k \in \mathbb{N}, \forall t' \in [\tau_k, \tau_{k+1}), \forall x' \in \mathbb{R}^n$$

is a candidate HNBF. Moreover, h is a valid CHNBF for (1) with the input $u : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ defined as

$$u(x', t') = u^k(x'), \forall k \in \mathbb{N}, \forall t' \in [\tau_k, \tau_{k+1}), \forall x' \in \mathbb{R}^n.$$

Remark 2: In this theorem, the choice of the switching interval remains arbitrary. Moreover, the controller u in the proof is only defined on $\mathbb{R}^n \times \mathbb{R}_{\geq t_0}$; however, it can be extended to $\mathbb{R}^n \times \mathbb{R}$, without affecting measurability and local boundedness, by setting it to zero on the region $\mathbb{R}^n \times \mathbb{R}_{<t_0}$.

In Thm. 2, the collection of CNBFs is not time varying, as is allowed for by Thm. 1; this choice is made so that Thm. 2 resembles the upcoming collision-avoidance algorithm, and such an extension (i.e., to make each $h^k : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$, $u^k : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$) would be possible. Regardless, with these results, the next subsection revisits a classic problem, collision avoidance, where the collision avoidance algorithm is encoded via an HNBF as in Thm. 2.

B. Composable Collision Avoidance

Using the main results of Sec. III-A, this section formulates a CHNBF for composable collision avoidance. Here, a scenario is modelled in which a sensor (e.g., LIDAR, infrared) or a map provides piece-wise-constant relative measurements to points that a robot must avoid. The system is assumed to evolve according to control-affine dynamics as in (1), and x is assumed to be the position of the robot. Formally, the measurement model takes the form of a piece-wise-constant set-valued map $M : \mathbb{R}_{\geq t_0} \to 2^{\mathbb{R}^n}$ that returns a finite set of points over intervals indicated by a switching sequence $\{\tau_k\}_{k=1}^{\infty}$. That is, $M(t') = M(t''), \forall t', t'' \in [\tau_k, \tau_{k+1})$. As such, the shorthand $M(\tau_k) = M(t'), t' \in [\tau_k, \tau_{k+1})$, indicates the points over the interval $[\tau_k, \tau_{k+1})$. The set-valued map M takes finite values in $\mathbb{R}^n$ with cardinality satisfying $|M(\tau_k)| = N_k$. The index $i \in N_k$ denotes a corresponding measurement $m_i \in M(\tau_k)$.

The value N_k represents the number of points to be avoided over the interval $[\tau_k, \tau_{k+1})$. For each of these points on the interval, the barrier function

$$h_i^k(x') = \|x' - m_i\|^2 - d^2, \; m_i \in M(t_k), \; i \in N_k, \quad (9)$$

encapsulates the constraint that the robot should remain at least a distance d away from the point m_i on the k^{th} interval. Then, the NBF given by

$$h^k = \bigwedge_{i=1}^{N_k} h_i^k$$

represents the constraint that all of these sampled points must be simultaneously avoided over a specific interval.

For controller-synthesis purposes, the almost-active gradient must be calculated. For this case, the almost-active gradient on an interval k at a point x' is given by

$$\partial_\epsilon h^k(x') = \mathrm{co} \bigcup_{i \in I_\epsilon^k(x')} \nabla h_i^k(x'),$$

where $\nabla h_i^k(x') = 2(x' - m_i)$. Importantly, this procedure greatly prunes the number of constraints that must be included for controller synthesis, as only those measurements which are relatively close to the robot must be utilized.

Splicing the component functions together, the CHNBF $h : \mathbb{R}^n \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ defined as

$$h(x', t') = h^k(x'), \forall k \in \mathbb{N}, \forall t' \in [\tau_k, \tau_{k+1}), \forall x' \in \mathbb{R}^n, \quad (10)$$

is a candidate CHNBF by direct application of Thm. 2. To synthesize a controller for the CHNBF, it suffices to consider the controller for each set of measurements, as per Thm. 2.

Since each of the individual gradients at each point in time is smooth, the results of [12] apply, and utilizing the almost-active gradient as a constraint to an optimization program yields a collision-avoiding controller. Let a nominal controller $u_k^{\mathrm{nom}} : \mathbb{R}^n \to \mathbb{R}^m$ represent the primary objective, then the solution to

Algorithm 1: Collision-Free Controller Synthesis.

Input: Nominal controller: $u_k^{\text{nom}}(x(t))$
Current measurements: $M(\tau_k)$
Output: Collision-free controller: $u(x(t), t)$.
$h_i^k(x(t)) \leftarrow \|x(t) - m_i\|^2 - d^2, \forall m_i \in M(\tau_k)$
$h^k(x(t)) \leftarrow \min_i\{h_i^k(x(t))\}$
$I_\epsilon^k(x(t)) \leftarrow \emptyset$
for $i = 1 : N_k$ **do**
  if $|h_i^k(x(t)) - h^k(x(t))| \leq \epsilon$ **then**
    $I_\epsilon^k(x(t)) \cup \{i\}$
$u(x(t), t) \leftarrow \arg\min_{u \in \mathbb{R}^m} \|u - u_k^{\text{nom}}(x(t))\|^2$
s.t. $2(x(t) - m_i)^\top (f(x(t)) + g(x(t))u) \geq$
$-\alpha^k(h^k(x(t))) \forall i \in I_\epsilon^k(x(t))$

the QP

$$u^k(x') = \arg\min_{u \in \mathbb{R}^m} \|u - u_k^{\text{nom}}(x')\|^2$$
$$\text{s.t. } 2(x' - m_i)^\top (f(x') + g(x')u) \geq -\alpha^k(h^k(x')),$$
$$\forall i \in I_\epsilon^k(x') \tag{11}$$

is a validating controller on each interval, by the results in [12, Thm. 3], assuming that the controller exists and is measurable and locally bounded. Specifically, each NBF h^k is a valid CNBF under u^k and α^k. Accordingly, the CHNBF in (10) is a valid CHNBF for (1) by Thm. 2. Note that each α^k is, effectively, a design parameter. In practice, by [12, Thm. 3], the almost-active gradient does not have to be included as a constraint directly. Rather, just the gradients of the component functions must be included, eliminating the convex-hull operation. In general, the QP in (11) is not feasible for every possible system. Rather, if the solution exists and is measurable and locally bounded, then Thm. 2 holds.

Alg. 1 combines these results into a cohesive procedure for calculating collision-free controllers; a consequence of this formulation is that the synthesized controller tries to match the nominal controller but will preferably ensure collision avoidance. The QP may be solved through many software libraries such as the MATLAB optimization toolbox or the Python library CVXOPT. The complexity of Alg. 1 depends on determining the ϵ-close points to the minimum (linear complexity) and solving a QP. Typically, strongly convex QPs, a class to which this algorithm belongs, have a runtime that is cubic in the number of decision variables. Assuming a standard conversion of inequality constraints to equality constraints, the runtime should be on the order of $O((m + N_k)^3)$ for most solvers. Finally, note that while the barrier functions, h_i^k, are hand designed, the composition is automatically computed by Alg. 1.

IV. Composable Collision-Avoidance for Differential-Drive Robots

Previous sections detailed the general methods and theoretical constraint-satisfaction guarantees afforded by the CHNBF controller-synthesis framework summarized in Alg. 1. The purpose of this section is to formulate these methods for a particular choice of dynamics in order to achieve composable collision-avoidance for LIDAR-equipped differential-drive robots. Moreover, Alg. 1 is encoded in the local coordinates of the robot,

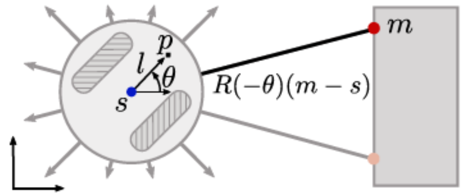

Fig. 1. The differential-drive robot, modeled by unicycle dynamics as in (12), is equipped with a LIDAR. The point p is a distance l from s, the center of the robot and the LIDAR, which returns the vector $R(-\theta)(m - s)$.

meaning that the robot does not need to estimate its global state; this formulation also makes the collision-avoidance algorithm independent from the particular objective at hand.

A. CHNBF Controller Synthesis in Local Coordinates

The CHNBF controller-synthesis method in Alg. 1[1] must be adapted to apply to LIDAR-equipped differential-drive robots, which are modelled by unicycle dynamics:

$$\dot{x} = v\cos(\theta) \qquad \dot{y} = v\sin(\theta) \qquad \dot{\theta} = \omega, \tag{12}$$

where $s = [x, y]^\top$ is the position; θ is the heading; and v, ω are the linear and angular velocity control inputs, respectively. The LIDAR is located at s.

Under unicycle dynamics, direct application of Alg. 1 results in limited control authority because ω does not appear in the time derivative of the position of the robot. Thus, Alg. 1 is instead formulated for a particular output of the state such that satisfying the collision-avoidance constraints for the output is sufficient to ensure collision-avoidance for the state; this output is chosen to be a point p orthogonal to the wheel axis of the robot as in Fig. 1, which is given by the following expression: $p = s + lR(\theta)e_1$, where $e_1 = [1, 0]^\top$ and $R(\theta)$ is the counterclockwise rotation matrix given by

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix};$$

note that $R(\theta) = R^\top(-\theta)$ and $R(-\theta) = R(\phi - \theta)R(-\phi)$.

Underpinning this choice of output is a near-identity diffeomorphism (NID), as presented in [24], which provides the following invertible mapping

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/l \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \dot{p} = LR(-\theta)\dot{p}.$$

Denote by r the radius of the robot. Now, observe that $\|p - m_i\| \leq \|s - m_i\| + l$. Then, picking $d \geq r + l$, ensures that $\|s - m_i\| \geq r$, when $\|p - m_i\| \geq d$. Considering p as the state variable, the barrier functions in (9) become $h_i^k(p) = \|p - m_i\|^2 - d^2$. Accordingly, it must be that

$$2(p - m_i)^\top \dot{p} \geq -\gamma(h_i^k(p))^3, \forall i \in I_\epsilon^k(p) \tag{13}$$

to satisfy Alg. 1, where $h(p) \mapsto h(p)^3$ is the selected extended class-$\mathcal{K}$ function and $\gamma > 0$. Applying identity transformations

[1] In the following sections, time dependence is suppressed for clarity.

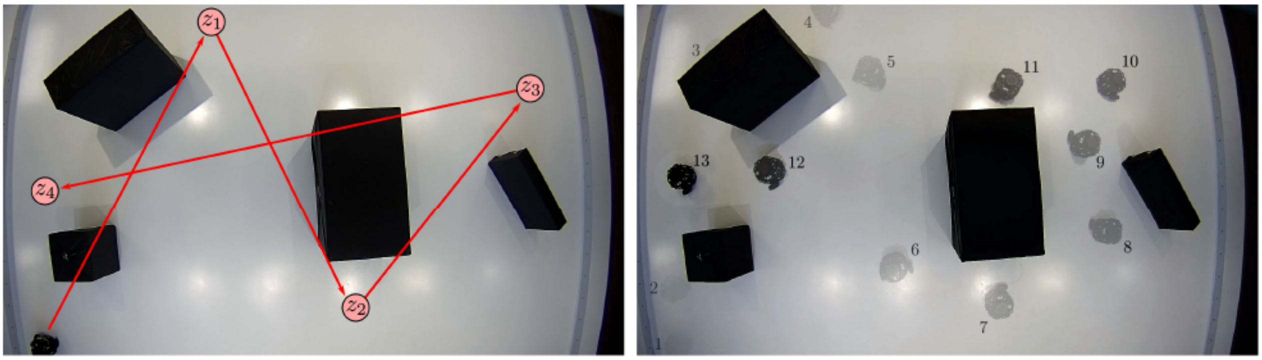

Fig. 2. (Left) The primary objective of the robot is to sequentially visit four points in the obstacle field. (Right) Using the controller-synthesis framework in Alg. 1 and Sec. IV, the robot navigates without collision; the numbers above the robot indicate its progress along the trajectory.

to (13) yields the following equations:

$$\begin{aligned} 2(p-m_i)^\top \dot{p} &= 2(p-m_i)^\top R(\theta)R(-\theta)\dot{p} \\ &= 2\left(R(-\theta)(s-m_i)+le_1\right)^\top L^{-1}u; \\ \left(\|p-m_i\|^2-d^2\right)^3 &= \left(\|R(-\theta)(s-m_i)+le_1\|^2-d^2\right)^3. \end{aligned}$$

Thus, the constraint in (13) can be written in local coordinates, and yielding the QP

$$\begin{aligned} u^k(p) &= \underset{u\in\mathbb{R}^2}{\arg\min}\, \|G(u-u_k^{\text{nom}})\|^2 \\ \text{s.t. } & 2\left(R(-\theta)(s-m_i)+le_1\right)^\top L^{-1}u \\ & \geq -\gamma\left(\|R(-\theta)(s-m_i)+le_1\|^2-d^2\right)^3, \forall i \in I_\epsilon^k(p), \end{aligned} \tag{14}$$

where G is a positive-definite weight matrix (in Sec. V, $G = L^{-1}$) and u_k^{nom} is the nominal control input to the unicycle in local coordinates. The QP in (14) selects a collision-free control input and can be solved onboard the robot without global information because $R(-\theta)(s-m_i)$ is in the coordinate frame of the robot. Also, note that $R(-\theta)(s-m_i)$ is not the static LIDAR reading at time τ_k. However, denoting the position and heading of the robot at the time τ_k by s_{τ_k} and θ_{τ_k}, $R(-\theta)(s-m_i)$ can be written:

$$\begin{aligned} R(-\theta)(s-m_i) &= R(-\theta)(s-s_{\tau_k}+s_{\tau_k}-m_i) \\ &= R(-\theta)(s-s_{\tau_k})+R(-\theta)(s_{\tau_k}-m_i) \\ &= \underbrace{R(\theta_{\tau_k}-\theta)}_{\text{local change in } \theta}\Big(\underbrace{R(-\theta_{\tau_k})(s-s_{\tau_k})}_{\text{local change in } s}-\underbrace{R(-\theta_{\tau_k})(m_i-s_{\tau_k})}_{\text{LIDAR measurement}}\Big). \end{aligned} \tag{15}$$

Thus, $R(-\theta)(s-m_i)$ can be obtained using LIDAR measurements and local change in state (e.g., from encoders).

In summary, Alg. 2 shows the local formulation of the CHNBF controller-synthesis framework for LIDAR-equipped differential-drive robots,[2] providing onboard composable, provably safe, and computationally straightforward collision avoidance. Because Alg. 2 is decoupled from the nominal controller, it can be easily integrated into a ROS node, providing plug-and-play collision avoidance for arbitrary objectives.

[2]For simplicity, Alg. 2 directly uses the LIDAR measurements under the assumption that the local changes in θ and s are negligible on the interval; the local state estimation in (15) can be used if this assumption is invalid.

Algorithm 2: Local Collision-Free Controller Synthesis for LIDAR-Equipped Differential-Drive Robots.

Input: Local nominal unicycle control input: u_k^{nom}
Current LIDAR measurements: laser_scans
Output: Collision-free unicycle control input: u^k.
for $i = 1 : N_k$ **do**
  $h_i^k \leftarrow \| - \text{laser\_scans}_i + le_1\|^2 - d^2$
  $\nabla h_i^k \leftarrow 2\left(-\text{laser\_scans}_i + le_1\right)$
$h^k \leftarrow \min_i\{h_i^k\}$
$I_\epsilon^k \leftarrow \emptyset$
for $i = 1 : N_k$ **do**
  if $|h_i^k - h^k| \leq \epsilon$ **then**
    $I_\epsilon^k \cup \{i\}$
$u^k \leftarrow \arg\min_{u\in\mathbb{R}^2} \|G(u-u_k^{\text{nom}})\|^2$
  s.t. $(\nabla h_i^k)^\top L^{-1}u \geq -\gamma(h_i^k)^3$, $\forall i \in I_\epsilon^k$

B. Parameter Selection

The values of ϵ and γ are chosen experimentally using the following rationale. Increases in ϵ result in a larger almost-active set, generating more constraints and increasing the runtime of the QP; the increase in the runtime is tempered by the practical benefits to collision avoidance from accounting for an increased number of nearby measured points. The choice of γ is informed by its effect on $\gamma h(p)^3$: if γ is small, then $\gamma h(p)^3$ becomes small around $h(p) = 0$, preventing the robot from making progress toward the measured points; for larger γ, the robot approaches measured points more freely.

V. Experimental Results

The theoretical results in Sec. III and the application of Alg. 2 to differential-drive robots in Sec. IV are validated using a TurtleBot3, a LIDAR-equipped differential-drive robot; all computations are performed onboard the embedded computer of the robot using local information because the methods proposed in Sec. III and Sec. IV are formulated to achieve collision avoidance independent of the primary objective. To demonstrate this point, a primary objective of sequentially visiting four points $Z = \{z_1, z_2, z_3, z_4\}$ is chosen, which is shown in Fig. 2. Here, the nominal controller representing this primary objective is a proportional controller that switches when the robot is within a threshold of a point, where switches concur with new measurements.

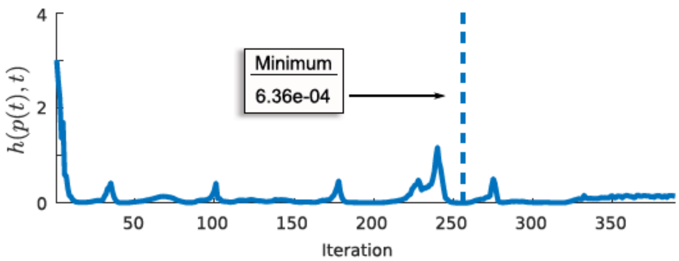

Fig. 3. $\min_t(h(p(t),t)) > 0$ indicates that the robot maintains a distance greater than r from all measured obstacles.

The local formulation of the CHNBF controller-synthesis method in Alg. 2 was implemented on the embedded computer of the TurtleBot3 as a ROS node in Python. The update rate of the LIDAR is 200ms; however, given the laser scanner measurements, Alg. 2 runs in approximately 10ms, suggesting that the possible performance of Alg. 2 is approximately 100Hz using the local estimation detailed in (15). In this implementation of Alg. 2, raw laser scanner measurements, admissible because of the controlled environmental conditions of the laboratory setting, were used. Notably, due to the form of Alg. 2, the navigation objective is independent of the ROS node, so any primary objective (i.e., nominal controller) could be integrated without modification of the ROS node.

In the experiment captured in Fig. 2, the TurtleBot3 synthesizes a controller that minimally modifies the nominal controller while satisfying the CHNBF constraints to avoid collisions ($\epsilon = 0.2$, $\gamma = 100$, $G = L^{-1}$, and $d = 0.2$ m). The leftmost image of Fig. 2 highlights the obvious collisions that would occur if the robot were to execute the nominal controller without accounting for obstacles. In the rightmost image of Fig. 2, the trajectory of the robot is sampled, showing that by executing the synthesized controller, the TurtleBot3 successfully navigates the obstacle field. To quantify this success, Fig. 3 shows that $\min_t(h(p(t),t)) \geq 0$ throughout the experiment, implying that the TurtleBot3 maintains a distance at least r from all measured obstacles.

VI. Conclusion

This work presented a hybrid extension of nonsmooth barrier functions and provided an associated controller-synthesis framework. These objects can encapsulate safety constraints for robotic systems and, due to their hybrid nature, represent dynamic changes. To demonstrate the practical nature of the theoretical results, a LIDAR-equipped mobile robot utilized the controller-synthesis framework to generate a collision-free controller in a navigation scenario.

Appendix

This appendix contains the proofs of Thms. 1 and 2.

A. *Proof of Theorem 1*

Let $x : [t_0, t_1] \to \mathbb{R}^n$ be a Carathéodory solution to (3) and assume that $(x(\tau_k), \tau_k) \in C, \forall k \leq K_{t_1}$. Let $t' \in [t_0, t_1]$. It remains to be shown that $(x(t'), t') \in C$. Since $\{\tau_k\}_{k=1}^{\infty}$ is a valid switching sequence, there exists a unique interval such that $t' \in [\tau_k, \tau_{k+1})$ for some k.

Consider the interval $[\tau_k, t']$. If $t' = \tau_k$, then $(x(t'), t') \in C$. Otherwise, $t' > \tau_k$. Since h is a candidate HNBF, it is locally Lipschitz on the interval $[\tau_k, t']$; thus, the function $[\tau_k, t'] \ni t \mapsto h(x(t), t)$ is absolutely continuous. Moreover, since h is a candidate HNBF, ∂h exists on (τ_k, t'), and by assumption (7),

$$\dot{h}(x(t),t) \geq \min \partial h(x(t),t)^\top (F(x(t),t) \times 1) \geq -\alpha^k(h(x(t),t))$$

almost everywhere on $(\tau_k, t') \ni t$, which is almost everywhere on $[\tau_k, t']$. Because $h(x(\tau_k), \tau_k) \geq 0$, $h(x(t), t) \geq 0, \forall t \in [\tau_k, t']$, by [11, Lem. 2]. Thus, $(x(t'), t') \in C$. ∎

B. *Proof of Theorem 2*

The proof proceeds by showing h is a candidate HNBF, writing ∂h in terms of each ∂h^k, showing that u is measurable and locally bounded, and writing $K[f + gu]$ in terms of each $K[f + gu^k]$. Then, the desired result follows from the application of Thm. 1. In each of these five cases, notation for the switching sequence is reused.

To show that h is a candidate HNBF, consider the switching interval given by assumption, and let $k \in \mathbb{N}$. By assumption, $h(x', t') = h^k(x'), \forall t' \in [\tau_k, \tau_{k+1})$. Consequently, h is also locally Lipschitz on $\mathbb{R}^n \times [\tau_k, \tau_{k+1})$, since $t' \mapsto h(x', t')$ remains constant on this interval. Thus, h is a candidate HNBF.

Now, consider $\partial h(x', t')$ on the set $\mathbb{R}^n \times (\tau_k, \tau_{k+1}) \ni (x', t')$. Let Ω_{h^k} be the zero-measure set in $\mathbb{R}^n$ where h^k is nondifferentiable. Then, $\Omega = \Omega_{h^k} \times \mathbb{R}_{\geq t_0}$ is zero-measure in $\mathbb{R}^n \times \mathbb{R}_{\geq t_0}$. By (6),

$$\partial h(x', t') = \text{co}\Big\{ \lim_{i \to \infty} \nabla h(x_i, t_i) : (x_i, t_i) \to (x', t'), (x_i, t_i) \notin \Omega_h, \Omega \Big\} = \text{co} A.$$

Take $z \in A$. By definition of h, $h(x_i, t_i) = h^k(x_i)$, so

$$z = \lim_{i \to \infty} \nabla h(x_i, t_i) = \lim_{i \to \infty} \nabla h^k(x_i) = \lim_{i \to \infty} \left[\nabla_x h^k(x_i)^\top, 0\right]^\top = \left[\lim_{i \to \infty} \nabla_x h^k(x_i)^\top, 0\right]^\top \in \partial h^k(x') \times 0$$

This follows from (6), because $\partial h^k(x')$ can be taken as

$$\partial h^k(x') = \text{co}\Big\{ \lim_{i \to \infty} \nabla_x h^k(x_i) : x_i \to x', x_i \notin \Omega_{h^k} \Big\}.$$

Thus, $\partial h(x', t') \subset \partial h^k(x') \times 0$, because $\partial h^k(x') \times 0$ is convex and $A \subset \partial h^k(x') \times 0$ implies that $\text{co} A \subset \partial h^k(x') \times 0$.

Now, consider the proposed control input u. To be a valid input, it must be measurable and locally bounded. To show that u is locally bounded, take $(x', t') \in \mathbb{R}^n \times \mathbb{R}_{\geq t_0}$. There exists a unique interval k such that $t' \in [\tau_k, \tau_{k+1})$. Since each u^k is locally bounded, there exists a $\delta_1 > 0$ such that $\|u^k(y)\| \leq M$, $\forall y \in B(x', \delta_1)$.

Because the switching sequence is strictly increasing, there exists a $\delta_2 > 0$ such that $[t', t' + \delta_2] \subset [\tau_k, \tau_{k+1})$. Accordingly, for every $y \in B(x', \delta_1)$, $s \in [t', t' + \delta_2] \|u(y, s)\| = \|u^k(y)\| \leq M$, showing that u is locally bounded.

It remains to be shown that u is measurable. Let O be an open set in $\mathbb{R}^n$. Then, it must be shown that $u^{-1}(O) = \{(x', t') \in \mathbb{R}^n \times \mathbb{R}_{\geq t_0} : u(x', t') \in O\}$ is measurable. This set is equivalent to

$$\bigcup_{k=1}^{\infty} \{(x', t') \in \mathbb{R}^n \times [\tau_k, \tau_{k+1}) : u(x', t') \in O\}.$$

Examining any $k \in \mathbb{N}$ yields that

$$\{(x', t') \in \mathbb{R}^n \times [\tau_k, \tau_{k+1}) : u(x', t') \in O\}$$
$$= (u^k)^{-1}(O) \times [\tau_k, \tau_{k+1}),$$

because $u^k(x') = u(x', t')$ on $\mathbb{R}^n \times [\tau_k, \tau_{k+1})$. Thus, if x' is such that $u^k(x') \in O$, then $u(x', t') \in O$ for every $t' \in [\tau_k, \tau_{k+1})$. Each u^k and every $[\tau_k, \tau_{k+1})$ is measurable, so the set $u^{-1}(O)$, is measurable.

Since u is measurable and locally bounded, consider $K[f + gu]$ on any interval $[\tau_k, \tau_{k+1})$. By definition, there exists a zero-measure set N_f^1 such that for any other zero-measure set N^1,

$$K[f + gu](x', t') =$$
$$\text{co}\Big\{ \lim_{i \to \infty} f(x_i) + g(x_i)u(x_i, t') : x_i \to x', x_i \notin N_f^1, N^1 \Big\},$$

for every $x' \in \mathbb{R}^n$, $t' \in [\tau_k, \tau_{k+1})$. Now, consider $K[f + gu^k]$. By definition, there exists a zero-measure set N_f^2 such that for any other zero-measure set N^2

$$K[f + gu^k](x') =$$
$$\text{co}\Big\{ \lim_{i \to \infty} f(x_i) + g(x_i)u^k(x_i) : x_i \to x', x_i \notin N_f^2, N^2 \Big\},$$

for any $x' \in \mathbb{R}^n$.

In each of these definitions, set $N^1 = N_f^2$ and $N^2 = N_f^1$. Consequently, since $u(x', t') = u^k(x')$ for every $x' \in \mathbb{R}^n$, $t' \in [\tau_k, \tau_{k+1})$,

$$K[f + gu](x', t') = K[f + gu^k](x'),$$

$\forall x' \in \mathbb{R}^n, t' \in [\tau_k, \tau_{k+1})$.

It remains to be shown that h is a valid HNBF for (4) under the control input u. Take any $k \in \mathbb{N}$; by the above results, examining any $x' \in \mathbb{R}^n$, $t' \in (\tau_k, \tau_{k+1})$ yields

$$\min \partial h(x', t')^\top \left(K[f + gu](x', t') \times 1\right)$$
$$= \min \partial h^k(x')^\top K[f + gu^k](x')$$
$$\geq -\alpha^k(h^k(x')) = -\alpha^k(h(x', t')),$$

so h is a valid HNBF for (4) by application of Thm. 1. ■

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[2] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: A review," *Precis. Agriculture*, vol. 13, no. 6, pp. 693–712, Dec. 2012.

[3] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6271–6278.

[4] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 477–492.

[5] X. Xu *et al.*, "Robustness of control barrier functions for safety critical control," *IFAC*, vol. 48, no. 27, pp. 54–61, 2015.

[6] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *Proc. 55th IEEE 55th Conf. Decis. Control*, 2016, pp. 2659–2664.

[7] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 3293–3298.

[8] D. Pickem *et al.*, "The Robotarium: A remotely accessible swarm robotics research testbed," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1699–1706.

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

[10] X. Xu, "Constrained control of input-output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.

[11] P. Glotfelter, J. Corts, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 310–315, Oct. 2017.

[12] P. Glotfelter, J. Corts, and M. Egerstedt, "Boolean composability of constraints and control synthesis for multi-robot systems via nonsmooth control barrier functions," in *Proc. IEEE Conf. Control Technol. Appl.*, Aug. 2018, pp. 897–902.

[13] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: A modular framework for fast and guaranteed safe motion planning," in *Proc. 56th IEEE Conf. Decis. Control*, Dec. 2017, pp. 1517–1522.

[14] M. Hoy *et al.*, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[15] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.

[16] I. Mitchell *et al.*, "Safety preserving control synthesis for sampled data systems," *Nonlinear Anal., Hybrid Syst.*, vol. 10, pp. 63–82, 2013.

[17] M. S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 475–482, Apr. 1998.

[18] J. P. Hespanha, "Uniform stability of switched linear systems: Extensions of LaSalle's invariance principle," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 470–482, Apr. 2004.

[19] F. Clarke, *Optimization and Nonsmooth Analysis*. Philadelphia, PA, USA: Soc. for Ind. and Appl. Math., 1990.

[20] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," in *Proc. 25th IEEE Conf. Decis. Control*, Dec. 1986, pp. 578–582.

[21] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1910–1914, Sep. 1994.

[22] J. Cortés, "Discontinuous dynamical systems," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.

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

[24] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential epsilon-tracking and epsilon-stabilization of first-order nonholonomic SE(2) vehicles," in *Proc. IEEE Proc. Amer. Control Conf.*, 2002, vol. 6, pp. 4690–4695.