Handling Disjunctions in Signal Temporal Logic Based Control Through Nonsmooth Barrier Functions

Adrian Wiltz and Dimos V. Dimarogonas

Abstract—For a class of spatio-temporal tasks defined by a fragment of Signal Temporal Logic (STL), we construct a nonsmooth time-varying control barrier function (CBF) and develop a controller based on a set of simple optimization problems. Each of the optimization problems invokes constraints that allow to exploit the piece-wise smoothness of the CBF for optimization additionally to the common gradient constraint in the context of CBFs. In this way, the conservativeness of the control approach is reduced in those points where the CBF is nonsmooth. Thereby, nonsmooth CBFs become applicable to time-varying control tasks. Moreover, we overcome the problem of vanishing gradients for the considered class of constraints which allows us to consider more complex tasks including disjunctions compared to approaches based on smooth CBFs. As a well-established and systematic method to encode spatio-temporal constraints, we define the class of tasks under consideration as an STL-fragment. The results are demonstrated in a relevant simulation example.

I. INTRODUCTION

In applications, one often encounters spatio-temporal constraints which impose both state- and time-constraints on a system. Logic expressions can be used to express such constraints. For example, one can form out of elementary rules Robot 1 must move within 5 seconds to region A (R1), or Robot 2 must move within 5 seconds to region B (R2), and Robot 1 and Robot 2 must keep a distance of at most d to each other (R3) the overall rule (R1 ∨ R2) ∧ R3 where ∧, ∨ denote logic AND and OR, respectively. Temporal logics like STL (Signal Temporal Logic) [14] allow the specification of such spatio-temporal constraints and increase the expressiveness of boolean logic by the temporal aspect. In the sequel, we call a composition of various spatio-temporal constraints by logic operators a task. Although STL originates from the field of formal verification in computer science, it is becoming increasingly popular as a well-established and systematic method to formulate spatio-temporal tasks in the field of control. Therefore, we also define the class of tasks under consideration as an STL-fragment in this paper. Most available control approaches for spatio-temporal tasks as [3], [6], [13] are based upon automata theory, which is often computationally expensive due to state discretization. Thereby, potential field based methods can be a computationally efficient alternative for some classes of spatio-temporal constraints [10], [11].

Potential field based methods have a long tradition in control theory and have been successfully applied to tasks as collision- [4] and obstacle-avoidance [16] as well as spatio-temporal tasks [10], [11]. In the latter case, smooth time-varying Control Barrier Functions (CBF) are employed. CBFs, introduced in [17] and [19], are a control concept for ensuring the invariance of sets and proved to be a suitable tool for guaranteeing the satisfaction of state constraints on control problems [1]. By now, a broad range of results on CBFs can be found in the literature. Whereas first approaches on CBFs [19], [2] consider systems with relative degree one, [15], [21] also consider systems with higher relative degree.

With view to constraints specified via logic expressions in the context of CBFs, especially two approaches must be named: For state-constraints specified via boolean logic, [9] employs nonsmooth CBFs. On the other hand, [10] constructs a smooth time-varying CBF that ensures the satisfaction of specified spatio-temporal tasks defined via an STL-fragment. However, since [10] uses a smoothed approximation of maximum and minimum operators, there exist points where the gradient of the CBF vanishes. This may be problematic when considering disjunctions (logic OR) in the context of time-varying CBFs.

In this paper, we resolve this problem by using a non-smooth CBF approach and can therefore take also disjunctions into account when considering spatio-temporal tasks. In contrast to [9], it is too conservative to require that a control action results in an ascend on multiple “active” CBFs at the same time. Therefore, although inspired by [7], [8], we do not base our control approach on the Filippov-operator and differential inclusions as [9], and employ a somewhat different approach in those points where the CBF is nonsmooth. In fact, we can circumvent the usage of differential inclusions by basing our controller on a set of optimization problems that exploit the piecewise smoothness of the CBF and we can show that the solutions to the closed-loop system are Carathéodory solutions. Thereby, we make the nonsmooth approach less conservative and thus applicable to time-varying CBFs.

The sequel is structured as follows: Section II introduces the considered dynamics, nonsmooth time-varying CBFs, and reviews STL; Section III constructs a nonsmooth CBF candidate for the STL-fragment under consideration, presents the control approach and proves set invariance; Section IV presents a relevant simulation example and demonstrates applicability of the proposed control scheme; Section V summarizes the conclusions of this paper. Proofs to the derived theoretic results can be found in the appendix of [20].
Notation: Sets are denoted by calligraphic letters. Let $\mathcal{A} \subseteq \mathbb{R}^n$, $\mathcal{B} \subseteq \mathbb{R}^m$, and let $d(\cdot,\cdot)$ define a metric on $\mathcal{A}$. The $\varepsilon$-neighborhood of $x \in \mathcal{A}$ is $B_{\varepsilon}(x) := \{y \in \mathcal{A} | d(y,x) < \varepsilon\}$. Int $\mathcal{A}$ the interior, $\partial \mathcal{A}$ the boundary of $\mathcal{A}$; the Lebesgue measure of $\mathcal{A}' \subseteq \mathcal{A}$ is $\mu(\mathcal{A}')$, and if a property of a function $f : \mathcal{A} \rightarrow \mathcal{B}$ holds everywhere on $\mathcal{A} \setminus \mathcal{A}'$ with $\mu(\mathcal{A}') = 0$, we say that it holds almost everywhere (a.e.). Let $\mathcal{I} \subseteq \mathbb{N}$ be a finite index set with cardinality $|\mathcal{I}|$ and $(a_i)_{i \in \mathcal{I}} := \{a_i | i \in \mathcal{I}\}$. Let $f_i : \mathcal{A} \rightarrow \mathcal{B}$. The maximum and minimum operators are denoted by $\max_{i \in \mathcal{I}} f_i$ and $\min_{i \in \mathcal{I}} f_i$, respectively, and we define $\max_{i \in \mathcal{I}} f_i := 0$, $\min_{i \in \mathcal{I}} f_i := 0$ for $\mathcal{I} = \emptyset$. A function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class $K$ function if it is continuous, strictly increasing and $\alpha(0) = 0$.

II. Preliminaries

At first, we introduce the system dynamics under consideration, redefine the concept of control barrier functions (CBF) in order to suit the control problem, and review STL-formulas as a formalism for defining complex spatio-temporal constraints on a control problem.

A. System Dynamics

We consider the input-affine system

$$\dot{x} = f(x) + g(x)u, \quad x(t_0) = x_0$$

on the closed time-interval $T = [t_1,t_2] \subseteq \mathbb{R}$, $t_0 \in T$, where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f,g$ are continuous functions with respective dimensions. Besides, we say that a time-varying set $\mathcal{C}(t) \subseteq \mathcal{X}$ is forward time-invariant for system (1), if $x(t) \in \mathcal{C}(t) \forall t \geq t_0$ for $x(t_0) = x_0 \in \mathcal{C}(t_0)$.

B. Non-Smooth Time-Varying Control Barrier Functions

Let $b(t,x)$ be a real-valued function $b : T \times \mathcal{X} \rightarrow \mathbb{R}$, and we define $T \subseteq T$ as the time for which $b(t,x) \equiv 0$ for all $t > T$. We assume that $b(t,x)$ is continuous and piecewise continuously differentiable in $x$, almost everywhere continuously differentiable in $t$, and

$$\lim_{t \rightarrow t^+} b(t,x) < \lim_{t \rightarrow t^+} b(t,x)$$

holds for discontinuities at $t < T$. Due to their role in what follows, we call a function $b(t,x)$ with the aforementioned properties a barrier function (BF).

Definition 1 (Safe Set). Let $T$ be the time-interval where (1) is defined. The set-valued function $\mathcal{C}(t) := \{x \in \mathcal{X} | b(t,x) \geq 0\}$ is called a time-varying safe set where $b$ is a barrier function. If $x(t) \in \mathcal{C}(t)$ for all times $t \in T$, we call $x$ feasible.

Note that $\mathcal{C}(t) \subseteq \mathcal{X}$ for $t > T$, i.e., the condition $b(t,x) \geq 0$ is trivially satisfied as $b(t,x) \equiv 0$. Next, we derive from the continuity properties of $b(t,x)$ the following continuity properties of the set-valued function $\mathcal{C}(t)$.

Lemma 1. The time-varying superlevel sets $\mathcal{C}(t) := \{x \in \mathcal{X} | b(t,x) \geq c\}$, $c \in \mathbb{R}$, of $b(t,x)$ are continuous a.e. with respect to $t$, and at a discontinuity at time $t$ it holds

$$\lim_{t \rightarrow t^+} \mathcal{C}(t) \subseteq \lim_{t \rightarrow t^+} \mathcal{C}(t).$$

For details on the continuity of set-valued functions, we refer to [18, Ch. 5B]. By (3) it is ensured that for a discontinuity at time $t$ it holds $x \in \lim_{t \rightarrow t^+} \mathcal{C}(t) \Rightarrow x \in \lim_{t \rightarrow t^+} \mathcal{C}(t)$, i.e., a feasible state stays feasible. Finally, we define control barrier functions for the nonsmooth time-varying case as follows.

Definition 2 (Control Barrier Function (CBF)). A barrier function $b(t,x)$ is a control barrier function for system (1) if there exists a class $K$ function $\alpha$ such that

$$\sup \{b(t,x) + \delta \} \geq - \alpha(b(t,x)) \quad \forall x \in \mathcal{C}(t) \text{ and all } t \in T \text{ where } b \text{ is continuous}.$$
In the sequel, we consider the STL-fragment
\[ \psi := T[p]\psi_1 \lor \psi_2 \psi_1 \land \psi_2 \] (8a)
\[ \phi := \phi_1 \lor \phi_2 \phi_1 \land \phi_2 \psi[G(a,b)\psi]\psi_1 \psi[G(a,b)\psi_2 \psi_1 \psi_2 \] (8b)
where \(a, b \in T\) and \(a \leq b\), which is the fragment considered in [10] extended by disjunctions. Below we denote STL-formulas satisfying grammar (8a) or (8b) by \(\psi\) or \(\phi\), respectively. As we see later, the problem of vanishing gradients in the presence of disjunctions as encountered in [10], [11] can be resolved with a non-smooth approach.

Assumption 1. We assume that \(h : \mathcal{X} \rightarrow \mathbb{R}\) is a continuously differentiable and concave function. Let \(\mathcal{H}\) be the set of all maximum points of \(h(x)\), i.e., \(\mathcal{H} := \{x_0 \in \mathcal{X} | \exists \varepsilon > 0 : |x_0 - x| < \varepsilon \Rightarrow h(x) < h(x_0)\}\). We additionally assume that \(L_g h(x) \neq 0 \forall x \in \mathcal{X} \setminus \mathcal{H}\) and \(L_g h(x) \neq 0\) if \(L_f h(x) \neq 0\) for \(x \in \mathcal{H}\) (first-order condition on \(h\)).

III. MAIN RESULTS

In the sequel, we present the construction of a BF which parallels [9], [10] in parts, and show that it satisfies the properties assumed in Section II-B. Thereafter, we outline the proposed control approach and prove the invariance of safe sets for the closed-loop system.

A. Construction of BFs

Consider an STL-formula \(\phi_0\) that satisfies grammar (8) and comprises predicates \(\{p_i\}_{i \in I^e}\) where \(I^e \subset \mathbb{N}\) is an index set; the corresponding predicate functions are \(\{h_i(x)\}_{i \in I^e}\).

In this section, our goal is to construct a BF \(b_0(t, x)\) for the STL-formula \(\phi_0\) such that \(b_0(t, x(t)) \geq 0 \forall t \in T\) implies \(x \models \phi_0\); then we say that \(b_0\) implements the STL-formula \(\phi_0\).

In a first step, we construct the BFs \(\{b_i\}_{i \in I^e}\) for each of the predicates \(\{p_i\}_{i \in I^e}\) which we call elementary barrier functions:

R0: For \(\psi_i = p_i\), the corresponding BFs are \(b_i(t, x) := h_i(x)\).

Using the set of elementary BFs as a starting point, we can recursively construct a BF \(b_0\) implementing \(\phi_0\).

Therefore, we introduce rules for the construction of BFs \(b_i\) which implement STL-formulas \(\psi_i\) and \(\phi_i\) satisfying grammar (8a) and (8b), respectively, as a composition of already constructed BFs \(\{b_j\}_{j \in B_i}\) where \(B_i \subset \mathbb{N}\) is a finite index set. The construction rules are given as follows:

R1: If \(\psi_i = \bigwedge_{\psi_j^t \in B_i} \psi_j^t\), choose \(b_i(t, x) = \min_{\psi_j^t \in B_i} b_j(t, x)\).

R2: If \(\psi_i = \bigvee_{\psi_j^t \in B_i} \psi_j^t\), \(b_i(t, x) = \max_{\psi_j^t \in B_i} b_j(t, x)\).

R3: For \(\phi_i = F(a,b)\psi_i\), we have \(B_i = \{\psi_i\}\) and choose \(b_i(t, x) = (b_i(t, x) + \gamma_i(t))\sigma^{-1}(t - \beta_i)\) where \(\sigma^{-1}\) is the inverse unit step as defined in the notation section, \(\gamma_i : T \rightarrow \mathbb{R}\) is a continuously differentiable function such that \(\exists \psi_i \in [a, b] : \gamma_i(t) = 0\), and time \(\beta_i := \min\{t' | \gamma_i(t') \leq 0\}\).

R4: For \(\phi_i = G(a,b)\psi_i\), we have \(B_i = \{\psi_i\}\) and choose \(b_i(t, x) = (b_i(t, x) + \gamma_i(t))\sigma^{-1}(t - \beta_i)\) where \(\sigma^{-1}\) is the inverse unit step, \(\gamma_i : T \rightarrow \mathbb{R}\) is a continuously differentiable, \(\gamma_i(t) \leq 0\) for all \(t \in [a, b]\), and time \(\beta_i = \min\{t' | \gamma_i(t') \leq 0\}\).

R5: If \(\phi_i = \bigwedge_{\phi_j^t \in B_i} \phi_j^t\), \(b_i(t, x) = \max_{\phi_j^t \in B_i} b_j(t, x)\) where \(B_i(t) := \{\psi_i \in B_i | \beta_i \geq \beta_j\}\) which means that \(b_i\) is deactivated at time \(\beta_i\), i.e., \(b_i\) does not depend on \(b_j\) for \(t > \beta_j\) anymore. In addition, set time \(\beta_i := \max_{\phi_j^t \in B_i} \beta_j\).

Fig. 1. Illustration of the recursive construction of \(b_0\).

R6: If \(\phi_i = \bigvee_{\phi_j^t \in B_i} \phi_j^t\), choose \(b_i(t, x) = \max_{\phi_j^t \in B_i} b_j(t, x)\sigma^{-1}(t - \beta_i)\) with \(B_i(t) = \{\psi_i \in B_i | \sigma^{-1}(t, x, t) \geq 0 \forall t \in [t_1, t_2]\}\) and time \(\beta_i := \min_{\phi_j^t \in B_i} \beta_j\).

R7: For \(\phi_i = \psi_i U(a,b)\psi_i\), note that \((x, t) = \psi_i U(a,b)U\psi_i' \iff \exists \psi_i' \in [t + a, t + b] s.t. (x, t) = \bigwedge_{\psi_i} \psi_i' \land F_{\psi_i'}\psi_i'\) and construction rules R2, R3 and R5 can be applied.

For BFs \(b_i\) constructed in R0, we define \(B_i = \emptyset\); for \(b_i\) constructed in R0, R1, R2, R3 and R4, we set \(\gamma_i(t) = 0\) and \(B_i(t) = B_i\); and for \(b_i\) constructed in R0, R1 and R2, we set \(\beta_i = \infty\). We call a scalar \(\beta_i\) deactivation time of \(b_i\). Deactivation times reduce the conservativeness of \(b_0\), cf. [10].

The set containing the indices of all BFs \(b_i\) constructed in intermediate steps of the construction of \(b_0\) is denoted by \(I\) and must be distinguished from \(\mathcal{I}\). Besides, we require that \(B_i \cap B_j = \emptyset\) for all \(i, j \in I\), i.e., every BF might be used for the construction of at most one other BF and hence \(b_0\) assumes a tree structure as illustrated by Figure 1.

We illustrate the application of the construction rules R0-R7 with an example.

Example 1. Consider \(\phi_0 = G[0,1][h_{11}(x) \geq 0] \land F[0,1][h_{211}(x) \geq 0] \land h_{121}(x) \geq 0\); here, we have \(I = \{11, 211, 212\}\). According to R0, define \(\psi_{11} := p_{11}, \psi_{211} := p_{211}\), and \(\psi_{212} := p_{212}\) with \(h_{11}, h_{211}, h_{212}\) as the respective predicate functions of predicates \(p_{11}, p_{211}, p_{212}\). Then, we start the recursive construction by choosing the elementary BFs as \(b_{11}(t, x) := h_{11}(x), b_{211}(t, x) := h_{211}(x), b_{212}(t, x) := h_{212}(x)\). Moreover, we define \(\phi_1 := G[0,1][\psi_{11}], \phi_2 := \psi_{211} \lor \psi_{212}, \phi_2 := F[0,1][\psi_{212}]\) and construct \(b_{11}, b_{212}\) by applying R4, R2, R3, respectively.

Now, we show that the elementary barrier functions from Section II-B and thus constitutes a BF.

Lemma 2. The function \(b_0(t, x)\) is a BF.

Next, we prove that the satisfaction of the time-dependent state-constraint \(b_0(t, x(t)) \geq 0\) for all \(t \in T\) implies the satisfaction of the STL-formula \(\phi_0\). In the next theorem, let \(\psi\) implement an STL-formula \(\psi\) satisfying grammar (8a) and \(\phi_0\) implement \(\phi\) satisfying grammar (8b).

Theorem 3. If \(\psi_0(t, x(t)) \geq 0\) for all \(t \in T\), then \(x \models \phi\).

In the sequel, we require the following assumption in addition to the fact that \(b_0\) is a BF.

Assumption 2. Let \(x\) be a maximum point of \(b_0\) at a given
time \( t \), i.e., \( b_0(t, x) \geq b_0(t, x') \) \( \forall x' \in B_\varepsilon(x) \) for some \( \varepsilon > 0 \).
We assume that there exists a constant \( b_{\text{min}} \in \mathbb{R} \) such that 
\( b_0(t, x) > b_{\text{min}} > 0 \) for any maximum point \( x \) of \( b_0 \) at any given time \( t > T \).

**Remark 2.** Assumption 2 excludes STL-formulas that require predictions in order to ensure forward invariance. Such control tasks are beyond the scope of this paper. In particular, Assumption 2 implies that there exist connected sets \( C_i(t) \) such that \( C(t) = \bigcup_i C_i(t) \) where \( \text{Int}(C_i(t)) \neq \emptyset \) for all \( t \in T \).

In the next section, we present a control scheme based on the constructed BF \( b_0 \) and show that it is a CBF.

**B. Controller Design**

In related CBF literature [1], [9], [10], an optimization problem is solved that ensures the satisfaction of a CBF gradient condition. However, directly solving
\[
 u^* = \arg \min_u u^T Q u \tag{9a}
\]
subject to
\[
 d_{\varepsilon} b_0(t + \delta, x + \delta(f(x) + g(x)u)) \bigg|_{\delta = 0} \geq -\alpha(b_0(t, x)) ,
\]
where (9b) ensures the satisfaction of the CBF gradient condition (4), is numerically difficult as \( b_0 \) is nonsmooth.
Therefore, we divide (9) into multiple basic optimization problems with a simplified gradient condition which can be numerically easily solved.

At first, we define some index sets that help us to describe the tree structure of the BFs constructed in Section III-A. Recall that we denote the index set of all BFs as \( \mathcal{I} \) and the index set of elementary BFs as \( \mathcal{I}^e \). For any \( i, k \in \mathcal{I}^e \), we define the index set \( Q^k_i(t) := \begin{cases} \{k\} & \text{if } i = k, \\ \emptyset & \text{otherwise} \end{cases} \) and for any \( k \in \mathcal{I}^e \) and \( i \in \mathcal{I} \preceq \mathcal{I}^e \) we define
\[
 Q^k_i(t) := \bigcup_{i' \in B_i(t)} Q^k_i(t) \cup \{i\} \quad \text{if } \exists i' \in B_i(t) : Q^k_i(t) \neq \emptyset
\]
otherwise.

Note that there exists \( \text{at most} \) one \( i' \in B_i(t) \) such that \( Q^k_i(t) \neq \emptyset \) because \( B_i \cap B_{i_1} = \emptyset \ \forall i_1, i_2 \in \mathcal{I} \) with \( i_1 \neq i_2 \) which is due to the tree structure of all BFs. Therefore, index set \( Q^k_i(t) \) can be interpreted as a branch in the tree structure that connects the BFs with indices and \( k \). Moreover, \( Q^k_i(t) \)

The active elementary BF index set for \( i \in \mathcal{I} \) is defined as
\[
 T^a_i(t, x) := \{k \in \mathcal{I}^e | \ Q^k_i(t) \neq \emptyset \ \wedge \\
 b_i(t, x) = h_k(x) + \sum_{i' \in Q^k_i(t)} \gamma_{i'}(t) \}
\]

**Example 2.** Consider \( \phi_0 \) in Example 1. As all expressions are considered for the same time \( t \) and state \( x \), we omit state and time arguments. According to (10), \( Q^{211}_0 = \{0, 2, 21, 211\} \) specifies the indices of the branch connecting BFs with indices 0 and 211. Correspondingly, \( Q^{212}_0 = \{0, 2, 21, 212\}, Q^{211}_2 = \{21\}, Q^{212}_2 = \{212\} \). Let \( T_0 = \{2\}, T_2 = \{21\}, T_2 = \{21, 212\} \) be active index sets. Then, as it can be seen from Figure 1, we have \( T_0 = T_0^{a} = T_0^{s} = \{21\}, T_2^{a} = \{212\} \) and \( T_2^{s} = \{21, 212\} \) according to (12). From these active elementary index sets, we can determine \( i_{kl} = 211, j_{kl} = 212, \) \( q_k = 21 \) for \( k = 211 \) and \( l = 212 \). Loosely speaking, \( b_{kl} \) denotes the BF from which the branches leading to \( b_k \) and \( b_l \) emanate, and \( i_{kl}, j_{kl} \) are chosen such that \( Q^k_{i_{kl}} \) and \( Q^l_{j_{kl}} \) do not contain common indices.

Next, we define subsets \( S_k(t) \subseteq \mathcal{X} \) on the state space as
\[
 S_k(t) := \{x | k \in T_0^{a}(t, x) \}
\]
which can be equivalently written as \( S_k(t) = \{x | b_0(t, x) = b_0_{kl}(t)\} = \{h_k(x) + \sum_{i \in Q^k_{i_{kl}}(t)} \gamma_{i}(t)\} \). Since for all times \( t \leq T \) and all states \( x \) in \( \mathcal{X} \) there exists at least one active elementary BF \( k \in T_0^{a}(t, x) \), it holds that \( \mathcal{X} = \bigcup_{k \in T_0^{a}} S_k(t) \).
The sets \( S_k(t) \) enjoy the favorable property that \( b_0(t, x) \) is continuously differentiable with respect to \( x \) in the interior \( \text{Int}(S_k(t)) \) and possibly non-smooth only on \( \partial S_k(t) \cap \partial S_i(t) \) for some \( l \in T^e \). Furthermore, it holds for some \( k, l \in T^e \) that \( b_0^l(t, x) = b_0^l(t, x) \) for all \( x \in S_0(t) \cap S_l(t) \), and in particular for \( x \in \partial S_0(t) \cap \partial S_l(t) \).

Given \( x(t) \in \partial S_0(t) \cap \partial S_l(t) \) for some \( t \in T \), we can therefore formulate the condition \( \partial S_0(t) \cap \partial S_l(t) \) and \( \partial S_0(t) \cap \partial S_l(t) \) and formulate an optimization problem for each \( k \in T_0^{a} \) with a gradient condition which is simplified in comparison to (9b).

For determining the boundary between \( S_k(t) \) and \( S_l(t) \), we define
\[
 s_{kl}(t, x) := \begin{cases}
 b_{l_{kl}}^k(t, x) - b_{l_{kl}}^k(t, x) & \text{if } q_k = \min_{i \in B_{l_{kl}}} q_i \\
 b_{l_{kl}}^k(t, x) - b_{l_{kl}}^k(t, x) & \text{if } q_k = \max_{i \in B_{l_{kl}}} q_i 
\end{cases}
\]
where \( b_0^k(t, x) := h_k(x) + \sum_{i \in Q^k_{i_{kl}}(t)} \gamma_{i}(t) \) (for \( i \in \mathcal{I}, k \in T_0^{a}(t, x) \)). Then for a given time \( t \), \( s_{kl}(t, x) = 0 \) determines those \( x \) on the boundary between \( S_k(t) \) and \( S_l(t) \), i.e., \( x \in \partial S_k \cap S_l \), \( x \in S_k \cap \partial S_l \), or \( x \in \partial S_l \cap \partial S_l \) if \( \frac{\partial s_{kl}(t, x)}{\partial x} \neq 0 \). Especially the set \( \partial S_k \cap \partial S_l \) is of interest as it can be shown that only then \( b_0 \) is non-differentiable in \( x \). The directional derivative of \( s_{kl} \) along the trajectory of (1) is given as
\[
 s_{kl}'(t, x, u) := \frac{\partial s_{kl}(t, x)}{\partial x} \left[ \frac{1}{f(x) + g(x)u} \right].
\]
For an illustration of (15), consider Figure 2. If \( x(t) \in \partial S_1(t) \cap \partial S_2(t) \), \( s_{12}(t, x, u) \geq 0 \) only admits inputs \( u \) such that \( x = f(x) + g(x)u \) points into \( S_1(t) \), thus \( x(t) \in S_1(t) \).
for all $t \in [t, t + \delta]$ and a $\delta > 0$.

Using this insight, we define optimization problems with a simplified gradient constraint as

$$u^*_k(t) = \text{argmin}_{u} u^T Q u$$

s.t. $\frac{\partial b_k}{\partial x}(t,x)(f(x) + g(x)u) + \frac{\partial b_k}{\partial t}(x,t) \geq -\alpha(b_0(t,x))$ (16a)

$$s'_{kl}(t,x,u) \geq 0 \quad \forall l \in T^a_0(x,t), l \neq k.$$ (16b)

for $k \in T^a_0(x,t)$ where $Q \in \mathbb{R}^{m \times m}$ is a positive-definite matrix and $\alpha$ a class $\mathcal{K}$ function. If (16) has no feasible solution, we set $u^*_k = \infty$. Finally,

$$u^*(t) = \text{argmin}_{u} u^*_k(t)^T Q u^*_k(t)$$ (17)

is applied as control input to (1). Note that in (15) and (16b) the time derivatives of $s_k$ and $b_0^*$ only exist on $\mathcal{T} \setminus \{\beta_i\} \in Q^*_0$. At times $t \in \{\beta_i\} \in Q^*_0$, $s_k$ and $b_0^*$ might be discontinuous and we consider the left sided derivative instead, i.e., $d_-s_k$ and $d_-b_0^*$, respectively. As it can be seen from the proof in Theorem 8, this choice is arbitrary and does not impact the invariance result. Next, we show that there always exists a feasible solution to (17) if the class $\mathcal{K}$ function $\alpha$ satisfies the following condition.

**Assumption 3.** It holds $\alpha(b_{\text{min}}) > -\frac{\partial \alpha}{\partial t}(t) \forall t \in \mathcal{T}, \forall i \in \mathcal{I}$.

Note that since $\gamma_i$ is continuously differentiable and defined on a closed interval, $\frac{\partial \gamma_i}{\partial x}$ is bounded. As the class $\mathcal{K}$ function $\alpha$ can be freely chosen, there always exists a class $\mathcal{K}$ function $\alpha$ such that Assumption 3 is fulfilled.

**Lemma 4.** Let Assumption 3 hold. For all $(t,x) \in \mathcal{T} \times \mathcal{C}(t)$, there exist $k \in T^a_0(x,t)$ such that (16) has a feasible and finite solution.

As it can be seen from the proof, in Assumption 3 the class $\mathcal{K}$ function $\alpha$ is chosen such that no increase in $b_0$ is required in (16b) by varying the system's state $x$ if $b_0(t,x) \geq b_{\text{min}}$. This is especially important when $x$ is already a maximum point of $b_0$ for a given time $t$ and a variation of $x$ does not lead to an increase on $b_0$.

Loosely speaking, $\alpha$ determines when $b_0(t,x)$ is sufficiently close to zero such that the controller reacts in order ensure the invariance of the safe set $\mathcal{C}(t)$, whereas functions $\gamma_i$ determine how "quickly" state $x$ has to change. Thereby, $\gamma_i$ is decisive for the magnitude of the control input $u$.

In the remainder of this section, we prove the forward invariance of $\mathcal{C}(t)$. Therefore, consider a solution $\varphi : \mathcal{T} \rightarrow \mathcal{X}$ to the closed-loop system (1) when control input (17) is applied. At first, we investigate the continuity properties of the control input trajectory $u : \mathcal{T} \rightarrow \mathbb{R}^m$ of the closed-loop system and show that $\varphi$ is a Carathéodory solution. However, this is not obvious as $u(t,x)$ is discontinuous both in $t$ and $x$.

**Lemma 5.** The control input trajectory $u^* : \mathcal{T} \rightarrow \mathbb{R}^m$ of the closed-loop system (1) with controller (16)-(17) is continuous a.e.

**Corollary 6.** A solution $\varphi : \mathcal{T} \rightarrow \mathcal{X}$ to the closed-loop system (1) with controller (16)-(17) is a Carathéodory solution. Moreover, it holds for the right sided derivative that $d_+\varphi(t) = f(\varphi(t)) + g(\varphi(t))u(t)$ for $t \notin \{\beta_i\} \in \mathcal{T}$.

**Remark 3.** Solution $\varphi$ is not necessarily unique. If multiple $u^*_k$ minimize the objective in (17), then any of the inputs could be applied. Depending on which input is chosen, different solutions $\varphi$ are obtained.

Next, we compare the controllers (9) and (16)-(17). This result facilitates the proof of forward invariance of the safe set in Theorem 8, and allows us to compare the proposed non-smooth CBF approach to other CBF-based controllers.

**Proposition 7.** The optimization problem (9) is equivalent to (16)-(17) for all $t \in \mathcal{T} \setminus \{\beta_i\} \in \mathcal{T}$.

In contrast to (9), the constraints in (16) can be more easily evaluated since $b_0^*$ is differentiable contrary to $b_0$. Based on the previous results, we can prove forward invariance of $\mathcal{C}(t)$.

**Theorem 8.** The control law (16)-(17) renders $\mathcal{C}(t)$ forward invariant for all $t \in \mathcal{T}$, and $b_0$ is a CBF.

**IV. SIMULATIONS**

Similarly to [10], we consider a multiagent system comprising three omnidirectional robots which are modeled as in [12] and use a collision avoidance mechanism as in [10]. The state of agent $i$ is given as $x_i \equiv [p_i^T, \rho_i]^T$ where $p_i = [x_{i,1}, x_{i,2}]$ denotes its position and $\rho_i$ its orientation; the state of all agents together is given as $x = [x_1^T, x_2^T, x_3^T]^T$. The dynamics of agent $i$ are

$$\dot{x}_i = f_i(x) + \begin{bmatrix} \cos(\rho_i) - \sin(\rho_i) \\ \sin(\rho_i) \cos(\rho_i) \end{bmatrix} 0 \begin{bmatrix} B_i^T \end{bmatrix}^{-1} R_i u_i$$

where $f_i(x) = \begin{bmatrix} f_{i,1}(x), f_{i,2}(x), 0 \end{bmatrix}^T$ with $f_{i,k}(x) = \sum_{j=1-j,k}^{3} \frac{\pi - \rho_{i,j,k}}{2.4} + 0.00001$ where $\frac{\pi - \rho_{i,j,k}}{2.4} + 0.00001 > 0$, $B_i = \begin{bmatrix} 0 & \cos(\pi/6) - \cos(\pi/6) \\ 1 & \sin(\pi/6) \sin(\pi/6) \end{bmatrix}$, and $R_i = 0.2$ as the radius of the agent, $R_i = 0.02$ is the wheel radius, and $u_i$ is the angular velocity of the wheels and serves as control input. As required, the system is input-affine and $f_i, g_i$ are continuous. Besides, we admit sufficiently large inputs to the system.

The task for the three agents comprises three parts: (1) approaching each other: $\phi_1 := \mathcal{F}_{[10,20]}(||p_1 - p_2|| \leq 10 \lor ||p_1 - p_3|| \leq 10 \lor ||p_2 - p_3|| \leq 10) \land \mathcal{G}_{[20,60]}(||p_1 - p_2|| \leq 15)$; (2) moving to given points: $\phi_2 := (||p_3 - [-5,-5,0]^T|| \leq 10) \land \mathcal{U}_{[20,50]}(||p_1 - p_2|| \leq 10) \land \mathcal{F}_{[10,20]}(||p_1 - [0,30,0]^T|| \leq 10) \land \mathcal{G}_{[50,60]}(||p_2 - [30,0,30]^T|| \leq 10) \land \mathcal{G}_{[50,60]}(||p_3 - [30,30,0]^T|| \leq 10)$; and (3) staying within a defined area: $\phi_3 := \mathcal{G}_{[50,60]}(||p_i^T, p_j^T, p_k^T||_\infty \leq 40)$. The norms $|| \cdot ||$ and $|| \cdot ||_\infty$ denote the euclidean and the maximum norm.
The overall task is given as the conjunction $\phi := \phi_1 \land \phi_2 \land \phi_3$. In contrast to [10], the considered task also contains disjunctions.

For the construction of the CBF $b_0(t, x)$, all rules from Section III-A (R0-R7) are applied; the controller is designed according to Section III-B. The simulation is implemented in Julia using Jump [5] and run on an Intel Core i5-10310U with 16GB RAM. The controller is evaluated with 50Hz input and control input is applied using a zero-order hold; the computation of the control input took 16ms on average. The trajectories of the agents resulting from the simulation are depicted in Figure 3, the evolution of $b_0$ and the applied inputs in Figure 4. Since $b_0(t, \varphi(t)) \geq 0 \ \forall t \in [0, 60]$, we conclude that the specified constraints are satisfied. Besides, the inputs are indeed continuous a.e.

V. CONCLUSION

In this paper, we constructed a nonsmooth time-varying CBF for Signal Temporal Logic tasks including disjunctions and derived a controller that ensures their satisfaction. By using a nonsmooth approach, we avoided the problem of vanishing gradients on the CBF that occurs when employing smoothed approximations of the minimum and maximum operators. Moreover, by partitioning the state space into sections and designing an optimization problem for each of them, we could determine the respective elementary barrier function “of relevance”. This allowed us to avoid the usage of differential inclusions in our derivation, thereby to reduce the conservativeness of our results and to apply a non-smooth approach to time-varying CBFs.

REFERENCES

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in 18th European Control Conference (ECC), 2019, pp. 3420–3431.
[2] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in 53rd IEEE Conference on Decision and Control, 2014, pp. 6271–6278.
[3] C. Bela, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Kлавins, and G. J. Pappas, “Symbolic planning and control of robot motion [grand challenges of robotics],” IEEE Robotics Automation Magazine, vol. 14, no. 1, pp. 61–70, 2007.
[4] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, “A feedback stabilization and collision avoidance scheme for multiple independent non-point agents,” Automatica, vol. 42, no. 2, pp. 229–243, 2006.
[5] I. Dunning, J. Huchette, and M. Lubin, “Jump: A modeling language for mathematical optimization,” SIAM Review, vol. 59, no. 2, pp. 295–320, 2017.
[6] G. Fainekos, H. Kress-Gazit, and G. Pappas, “Temporal logic motion planning for mobile robots,” in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 2020–2025.
[7] A. F. Filippov, “Differential equations with discontinuous right-hand side,” in Fifteen Papers on Differential Equations, ser. American Mathematical Society Translations, vol. 42, no. 2. American Mathematical Society, 1964, pp. 199–231.
[8] ——, Differential equations with discontinuous right hand sides, ser. Mathematics and its applications Soviet series, 18. Dordrecht: Kluwer Academic Publishers, 1988.
[9] P. Glosfelter, J. Cortes, and M. Egerstedt, “A nonsmooth approach to controller synthesis for boolean specifications,” IEEE Transactions on Automatic Control, pp. 5160–5174, 2020.
[10] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for signal temporal logic tasks,” IEEE Control Systems Letters, vol. 3, no. 1, pp. 96–101, 2019.
[11] 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, vol. 7, no. 4, pp. 1916–1928, 2020.
[12] Y. Liu, J. J. Zhu, R. L. Williams, and J. Wu, “Omni-directional mobile robot controller based on trajectory linearization,” Robotics and Autonomous Systems, vol. 56, no. 5, pp. 461 – 479, 2008.
[13] S. Loizou and K. Kyriakopoulos, “Automatic synthesis of multi-agent motion tasks based on ltl specifications,” in 43rd IEEE Conference on Decision and Control (CDC), vol. 1, 2004, pp. 153–158.
[14] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 152–166.
[15] Q. Nguyen and K. Sreenath, “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints,” in American Control Conference (ACC), 2016, pp. 322–328.
[16] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris, “Multi-objective control for multi-agent systems using lyapunov-like barrier functions,” in 52nd IEEE Conference on Decision and Control, 2013, pp. 1478–1483.
[17] S. Prajna and A. Jadbabaie, “Safety verification of hybrid systems using barrier certificates,” in Hybrid Systems: Computation and Control, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 477–492.
[18] R. T. Rockafellar, Variational analysis, 1st ed., ser. Grundlehren der mathematischen Wissenschaften. Springer Berlin, Heidelberg, 2009.
[19] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” IFAC Proceedings Volumes, vol. 40, no. 12, pp. 462–467, 2007, 7th IFAC Symposium on Nonlinear Control Systems.
[20] A. Wiltz and D. V. Dimarogonas, “Handling disjunctions in signal temporal logic based control through nonsmooth barrier functions,” 2022, [Online]. Available: https://arxiv.org/abs/2208.10803.
[21] W. Xiao and C. Belta, “High order control barrier functions,” IEEE Transactions on Automatic Control, 2021.