Continuous Reachability Task Transition Using Control Barrier Functions [⋆]

Mohit Srinivasan [∗] **Cesar Santoyo** [∗] **Samuel Coogan** [∗∗]

[∗] *Georgia Institute of Technology, School of Electrical & Computer Engineering, Atlanta, GA, 30318, USA,*
Email: {mohit.srinivasan, csantoyo} @gatech.edu
[∗∗] *Georgia Institute of Technology, School of Electrical & Computer Engineering and School of Civil & Environmental Engineering, Atlanta, GA, 30318, USA, Email: sam.coogan@gatech.edu*

Abstract: In this paper, a method to achieve smooth transitions between sequential reachability tasks for a continuous time mobile robotic system is presented. Control barrier functions provide formal guarantees of forward invariance of safe sets and finite-time reachability and are able to enforce task execution. Barrier functions used in quadratic programs result in implementation of controllers with real-time performance guarantees. Existing approaches for multi-objective task execution using control barrier functions leverage discretely switched, sequential quadratic programs to achieve successive tasks. However, discrete switching can lead to control input discontinuities which can affect a robot's performance. Hence, we propose a method which ensures continuous transitions between sequential quadratic programs. In particular, a time varying component to the barrier function constraint is introduced which allows for a smooth transition between objectives. Robotic implementation results are also provided.

Keywords: Mobile robots, Autonomous robotic systems, Robotics technology

1. INTRODUCTION

Increased complexity of autonomous robotic systems and demands for safety guarantees have made it imperative that formal guarantees of safety and performance are a cornerstone of control synthesis. In particular, synthesized actions are often required to be continuous inputs which do not result in spasmodic motions. For example, in systems such as humanoid robots Kim et al. (2019), Han and Park (2013), it is critical that the control inputs to the manipulator joints are continuous in nature, and do not suffer from discontinuities. To that end, we present a framework which satisfies a system's performance and safety specifications while ensuring continuity in the control law when transitioning to another objective.

Control barrier functions (CBFs) used in conjunction with quadratic programs (QPs) lead to real-time implementation of controllers which satisfy safety and reachability requirements. For example, Wang et al. (2016) utilize CBFs to ensure collision avoidance amongst teams of robots. Ames et al. (2014) employ CBFs to guarantee speed and car separation objectives in the context of adaptive cruise control. More recently, Srinivasan and Coogan (2019) leveraged CBFs for motion planning specifications for teams of robots. Pierpaoli et al. (2019) execute a sequence of tasks and transition from an existing graph structure to the desired one using CBFs. In these papers, the authors transition discretely between tasks using sequential QPs. Each QP encodes a different barrier function constraint based on the task to be satisfied.

However, as is shown in Kim et al. (2019), sudden switching between constraints can lead to discontinuities in the control input which adversely affect the stability and functionality of the robot. Hence, in this paper, we address this issue by formulating a barrier function constraint with time dependent properties which allow for continuous transition between different tasks. More specifically, each barrier function is endowed with a time-varying transition coefficient which acts as a buffer signal that appropriately switches the constraint on or off. The authors in Kim et al. (2019) use a similar time varying parameter to smoothly transition between different tasks in the context of humanoid robots and manipulators. However, the authors do not provide any formal guarantees on the satisfaction of the newly added task or the continuity of the proposed controller.

The contributions of this paper are threefold. First, we formulate a new barrier function constraint which allows for smooth transitions between different reachability tasks to be satisfied by a mobile robotic system while simultaneously satisfying safety constraints. With a simple example, we demonstrate that discrete switching between successive QPs leads to a discontinuity in the control law whereas the proposed constraint allows for a continuous transition of the control. Second, we provide an algorithm for online implementation of the proposed barrier function based

controller. In addition, we demonstrate controller continuity. Lastly, we present experimental results which verify the proposed framework on a differential drive mobile robot.

The paper is organized as follows: Section 2 provides mathematical background on control barrier functions, and the QP-based controller. Section 3 details a motivating example which illustrates why a discrete transition between sequential QPs can lead to discontinuous control inputs, and also formalizes the problem statement addressed in this paper. Section 4 discusses the proposed barrier function constraint which guarantees continuity of the control law. Section 5 and 6 describe the applicability of the proposed controller, and concluding remarks, respectively.

2. MATHEMATICAL BACKGROUND

Consider a control affine dynamical system of the form
$$\dot{x} = f(x) + g(x)u, \tag{1}$$
where $x \in \mathcal{D} \subset \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is the control input applied to the system, and f and g are locally Lipschitz continuous functions.

2.1 Zeroing Control Barrier Functions (ZCBFs)

Let $\mathcal{C} = \{x \in \mathcal{D} \mid h(x) \geq 0\}$ be a safe set defined as the super zero level-set of a continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$. We define a continuous function $\mu : \mathbb{R} \rightarrow \mathbb{R}$ as a class κ function if $\mu(0) = 0$ and it is strictly increasing.

Definition 1 (Ames et al. (2019)). *A continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$ is a zeroing control barrier function (ZCBF) if there exists a locally Lipschitz class κ function μ such that for all $x \in \mathcal{D}$*
$$\sup_{u \in \mathbb{R}^m} \{L_f h(x) + L_g h(x)u + \mu(h(x))\} \geq 0 \ ,$$
where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$, and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$.

In particular, choosing control inputs from the set
$$\mathcal{U}_{\text{safe}}(x) = \{u \in \mathbb{R}^m \mid L_f h(x) + L_g h(x)u + \mu(h(x)) \geq 0\}$$
renders $\mathcal{C}$ forward invariant. Since class $\mathcal{K}$ functions fall under the category of minimal functions, using the formalism in Konda et al. (2019), we can guarantee forward invariance of $\mathcal{C}$.

Proposition 1 (Theorem 6, Konda et al. (2019)). *Let $\mathcal{C} \subset \mathcal{D}$ be a safe set defined as $\mathcal{C} = \{x \in \mathcal{D} \mid h(x) \geq 0\}$ where $h : \mathcal{D} \rightarrow \mathbb{R}$, and suppose $x(0) \in \mathcal{C}$. If h is a ZCBF, then any continuous feedback controller satisfying $u(x) \in \mathcal{U}_{safe}(x)$ for all $x \in \mathcal{C}$ renders the set $\mathcal{C}$ forward invariant.*

2.2 Finite-time Control Barrier Functions (FCBFs)

Let $\Gamma = \{x \in \mathcal{D} \mid h(x) \geq 0\}$ be a target set defined as the super zero level-set of a continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$. Li et al. (2018) introduce finite-time control barrier functions (FCBFs) which guarantee finite-time convergence to a desired target set in the domain, as formalized next.

Definition 2. *A continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$ is a finite-time control barrier function (FCBF) if there exists parameters $\gamma > 0$ and $\rho \in [0, 1)$ such that for all $x \in \mathcal{D}$*
$$\sup_{u \in \mathbb{R}^m} \{L_f h(x) + L_g h(x)u + \gamma \cdot \text{sign}(h(x)) \cdot |h(x)|^\rho\} \geq 0.$$
where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$, and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$.

Given a target set Γ, choosing control inputs from the set
$$\mathcal{U}_{\text{target}}(x) = \{u \in \mathbb{R}^m \mid L_f h(x) + L_g h(x)u +$$
$$\gamma \cdot \text{sign}(h(x)) \cdot |h(x)|^\rho \geq 0\}$$
allows the system to converge to Γ in finite time, as formalized in the following proposition.

Proposition 2 (Proposition III.1, Li et al. (2018)). *Let $\Gamma \subset \mathcal{D}$ be a target set defined as $\Gamma = \{x \in \mathcal{D} \mid h(x) \geq 0\}$ where $h : \mathcal{D} \rightarrow \mathbb{R}$. If h is a FCBF, then, for any initial condition $x_0 \in \mathcal{D} \backslash \Gamma$ and any continuous feedback control $u : \mathcal{D} \rightarrow \mathbb{R}^m$ satisfying $u(x) \in \mathcal{U}_{target}(x)$ for all $x \in \mathcal{D}$, the system will be driven to the set Γ in a finite time $0 < T < \infty$; that is, $x(T) \in \Gamma$.*

2.3 Quadratic Program Based Controller

Given a FCBF and/or a ZCBF, control synthesis can be encoded as a QP which is amenable to efficient online computation of feasible control inputs. In particular, for fixed $x \in \mathcal{D}$, the requirement that $u(x) \in \mathcal{U}_{\text{safe}}(x)$ and/or $u(x) \in \mathcal{U}_{\text{target}}(x)$ becomes a linear constraint and we define a minimum energy QP as
$$\begin{aligned} \underset{u \in \mathbb{R}^m}{\text{minimize}} \quad & ||u||_2^2 \\ \text{s.t} \quad & u \in \mathcal{U}_{\text{target}}(x) \text{ and/or } u \in \mathcal{U}_{\text{safe}}(x). \end{aligned} \tag{2}$$

If feasible for all time, (2) returns the point-wise in time, minimum energy control action which guarantees that the system satisfies the safety constraints dictated by the ZCBF (Proposition 1) and/or the reachability constraints dictated by the FCBF (Proposition 2). While the QP can be constrained individually by either the reachability or safety constraints, we are specifically interested in achieving reachability tasks while simultaneously enforcing safety.

3. MOTIVATING EXAMPLE & PROBLEM STATEMENT

Suppose the system of interest is an omnidirectional robot with single integrator dynamics $\dot{x} = u$, where $x \in \mathbb{R}^2$ is the state, and $u \in \mathbb{R}^2$ is the control input applied to the robot. Consider a domain $\mathcal{D} \subset \mathbb{R}^2$ containing two regions of interest- region A and region B. Let $\mathcal{A} = \{x \in \mathcal{D} \mid h_A(x) \geq 0\}$ and $\mathcal{B} = \{x \in \mathcal{D} \mid h_B(x) \geq 0\}$, where $h_A : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $h_B : \mathbb{R}^2 \rightarrow \mathbb{R}$ are continuously differentiable functions. Suppose the task to be satisfied by the robot is to visit region A first followed by region B. Adhering to the methodology suggested in Srinivasan and Coogan (2019), one can formulate this task as a sequence of two QPs.

Quadratic Program 1: The first QP is defined to be
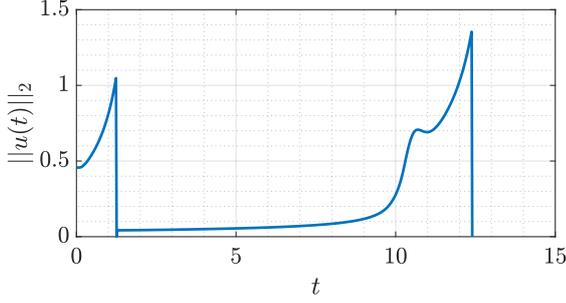
Fig. 1. The control input generated for the sequence of tasks discussed in Example 1. The sudden switching of the reachability constraint from QP 1 to QP 2 yields a discontinuity in the control law as is seen here. In Section 5, we show that using our proposed framework, a continuous control input is obtained.

$$\min_{u \in \mathbb{R}^2} \quad ||u||_2^2$$
$$\text{s.t} \quad \frac{\partial h_A(x)}{\partial x} \cdot u \geq -\gamma \cdot \text{sign}(h_A(x)) \cdot |h_A(x)|^\rho$$

where $\gamma > 0$, and $\rho \in [0, 1)$. This QP is solved until the system reaches $\mathcal{A}$.

Quadratic Program 2: Next, the QP constraint is switched to a new one to reflect the new task, i.e., convergence to region B. Hence, the second QP is defined to be

$$\min_{u \in \mathbb{R}^2} \quad ||u||_2^2$$
$$\text{s.t} \quad \frac{\partial h_B(x)}{\partial x} \cdot u \geq -\gamma \cdot \text{sign}(h_B(x)) \cdot |h_B(x)|^\rho$$

where $\gamma > 0$, and $\rho \in [0, 1)$.

The sudden change in the constraint yields a discontinuous control law as illustrated in Fig 1. A similar example of this problem is discussed in Kim et al. (2019) and Jarquín et al. (2013). Hence, discretely switching from one QP to the next creates discontinuities in the control input to the robot, which is undesirable. To address this problem, we propose a new barrier function constraint with time dependent coefficients which guarantee continuity of the control law. Instead of solving a sequence of QPs with discrete transitions, we propose a single QP which smoothly transitions between different barrier function constraints.

To that end, consider a continuous time mobile robotic system in control affine form as in (1). We assume that there exists a sequence of tasks to be executed by the system. Below, we provide a formal definition for a task.

Definition 3. *Given a target set $\Gamma \subset \mathcal{D}$ and a safety set $\Sigma \subset \mathcal{D}$, the task $\mathcal{T}_{(\Gamma, \Sigma)}$ is defined as the reachability problem requiring that the system reach Γ in finite time, while staying in Σ for all time.*

Define $\mathcal{T} = \{\mathcal{T}_{(\Gamma_1, \Sigma)}, \ldots, \mathcal{T}_{(\Gamma_n, \Sigma)}\}$ where each $\mathcal{T}_{(\Gamma_i, \Sigma)}$ is a task that the system must satisfy in finite time and each sequential task is distinct, i.e., $\mathcal{T}_{(\Gamma_i, \Sigma)} \neq \mathcal{T}_{(\Gamma_{i+1}, \Sigma)}$ for all i.

Informally, each task $\mathcal{T}_{(\Gamma_i, \Sigma)}$ consists of reachability and invariance constraints. Invariance constraints are assumed to be global constraints which do not change between consecutive tasks; however, the reachability constraints are distinct between tasks. These constraints are encoded in a

QP which is solved until the system reaches the target set Γ_i. Suppose there exists m finite-time barrier functions and each target set Γ_i for the task $\mathcal{T}_{(\Gamma_i, \Sigma)}$ is characterized as the intersection of some subset of the m barrier functions. Let $\mathcal{I} = \{1, 2, \ldots, m\}$ be the index set for the finite-time barrier functions. Given a target set Γ_i for a task i, the following definition formalizes the finite-time barrier functions which characterize Γ_i.

Definition 4 (Reachability Set Map). *The reachability set map $\Delta : \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\} \to 2^{\mathcal{I}}$ yields the finite-time barrier functions characterizing the target set such that $\Gamma_i = \bigcap_{\forall j \in \Delta(\Gamma_i)} \{x \in \mathcal{D} \mid h_j(x) \geq 0\}$ for all $i \in \{1, 2, \ldots, n\}$.*

Next, we formalize the problem statement addressed in this paper.

Problem Statement 1. *Given a system as in* (1)*, and a sequence of tasks $\mathcal{T} = \{\mathcal{T}_{(\Gamma_1, \Sigma)}, \ldots, \mathcal{T}_{(\Gamma_n, \Sigma)}\}$, synthesize a continuous controller such that the system satisfies each $\mathcal{T}_{(\Gamma_i, \Sigma)}$ within a finite time, while smoothly transitioning between sequential tasks.*

4. BARRIER FUNCTION BASED SMOOTH TASK TRANSITION

Smoothly transitioning between tasks requires a transition function which gradually winds down the weight(s) of the constraint(s) corresponding to an accomplished task in the quadratic program (QP) while ramping up the weight(s) of the constraint(s) of the successive task. In this section, we propose a framework to achieve this.

4.1 QP Based Algorithm

Consider a sequence of n reachability tasks represented as $\mathcal{T} = \{\mathcal{T}_{(\Gamma_1, \Sigma)}, \ldots, \mathcal{T}_{(\Gamma_n, \Sigma)}\}$, and m finite time barrier functions indexed by the set $\mathcal{I}$. Introduce transition function $\alpha_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for all $i \in \mathcal{I}$. Then, for all $t \geq 0$, we formulate a time varying feedback control QP of the form

$$u^*(t, x) = \min_{u \in \mathbb{R}^m} \quad ||u||_2^2$$
$$\text{s.t} \quad \sum_{i=1}^{m} \left\{ \alpha_i(t)(L_f h_i(x(t)) + L_g h_i(x(t))u(x(t))) \right\} +$$
$$\sum_{i=1}^{m} h_i(x(t)) \frac{\partial \alpha_i(t)}{\partial t}$$
$$\geq -\gamma \cdot \tanh\left(-\ln\left(\sum_{i=1}^{m} \exp(-\alpha_i(t)h_i(x(t))) \right) \right)$$
$$u \in \mathcal{U}_{\text{safe}}(x)$$
$$||u||_\infty \leq M, \tag{3}$$

where $\alpha_i(t)$ is determined from Algorithm 2 for all $i \in \mathcal{I}$, and $\gamma > 0$. The first constraint captures the reachability part of each task specification, whereas the second constraint represents the safety requirements to be satisfied by the system. The last constraint represents actuator limits where $M > 0$. This optimization problem must be solved point-wise in time i.e. in a sampled-data fashion.

We construct Algorithm 1 and Algorithm 2, which use (3) to generate the control inputs required to satisfy the given

sequence of reachability tasks for the system. Algorithm 1 utilizes the function characterized in Algorithm 2. In particular, Algorithm 1 is executed for all $t \geq 0$. The transition functions $\alpha_i(t)$ for all $i \in \mathcal{I}$ and for all $t \geq 0$ are chosen as per Algorithm 2 point-wise in time. The functions $\kappa^\uparrow : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $\kappa^\downarrow : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are strictly increasing and decreasing continuously differentiable functions, respectively. Assuming T_i is the time instant at which the robot reaches target set Γ_i, the functions $\kappa^\uparrow$ and $\kappa^\downarrow$ are chosen such that $\kappa^\uparrow(t - T_i) \mid_{t=T_i} = 0$ and $\kappa^\downarrow(t - T_i) \mid_{t=T_i} = 1$. Choices for $\kappa^\uparrow$ and $\kappa^\downarrow$ include functions such as sine, cosine, hyperbolic tangent, sigmoid, etc.

Algorithm 1 Smooth Transition Between Sequential Reachability Tasks

---
1: **procedure**
2: **Input :** h_i for all $i \in \mathcal{I}$, $\gamma > 0$
3:    **for** $i \in \{1, \ldots, n\}$ **do**   ▷ Loop through each task
4:       $T_i \leftarrow 0$
5:       $s \leftarrow 0$ ▷ Flag variable to indicate transition
6:       **while** $x \notin \Gamma_i$ **do**   ▷ Reachability Phase
7:          COMPUTE-$\alpha(t, T_i, x, i, s)$
8:          Solve the QP (3)
9:          Apply $u(t, x)$ to the system
10:       **end while**
11:       $T_i \leftarrow t$ ▷ Record time instant when $x(T_i) \in \Gamma_i$
12:       $s \leftarrow 1$
13:       **while** $\alpha_k < 1$ & $\alpha_j > 0$ $\forall$ $k \in \Delta(\Gamma_{i+1})$, $j \in \Delta(\Gamma_i)$ **do**   ▷ Transition Phase
14:          COMPUTE-$\alpha(t, T_i, x, i, s)$
15:          Solve the QP (3)
16:          Apply $u(t, x)$ to the system
17:       **end while**
18:    **end for**
19: **end procedure**

Algorithm 2 Compute Transition Function

1: **function** COMPUTE-$\alpha(t, T_i, x, i, s)$
2:    **if** $s = 0$ **then**
3:       $\alpha_j \leftarrow 1$ for all $j \in \Delta(\Gamma_i)$
4:       $\alpha_k \leftarrow 0$ for all $k \in \mathcal{I} \setminus \Delta(\Gamma_i)$
5:    **else if** $s = 1$ **then**
6:       $\alpha_j \leftarrow \kappa^\downarrow(t - T_i)$ for all $j \in \Delta(\Gamma_i)$
7:       $\alpha_l \leftarrow \kappa^\uparrow(t - T_i)$ for all $l \in \Delta(\Gamma_{i+1})$
8:    **end if**
9: **end function**

4.2 Continuity of QP-based Controller & Reachability Task Satisfaction Guarantee

The following theorem provides a composite barrier function constraint which allows for smooth task transition.

Lemma 1. *Consider a task $\mathcal{T}_{(\Gamma,\Sigma)}$ with the target set Γ defined as $\Gamma = \bigcap_{i \in \mathcal{P}} \{x \in \mathcal{D} \mid h_i(x) \geq 0\}$ where $\mathcal{P} = \{1, 2, \ldots, k\}$, with each barrier function h_i bounded. That is, $h_i(x) \leq M_i$ for all $x \in \mathcal{D}$ where $M_i > 0$. If there exists a continuous controller $u : \mathcal{D} \to \mathbb{R}^m$ such that for all $x \in \mathcal{D}$ and for all $t \geq 0$,*

$$\sum_{i \in \mathcal{P}} \left\{ (L_f h_i(x(t)) + L_g h_i(x(t)) u(x(t))) \right\}$$
$$\geq -\gamma \cdot \tanh\left(-\ln\left(\sum_{i \in \mathcal{P}} \exp(-h_i(x(t))) \right) \right) \quad (4)$$

then there exists a time instance $0 < T < \infty$ such that $x(T) \in \Gamma$.

Proof. By contradiction, suppose for some $x(0) \in \mathcal{D} \setminus \Gamma$, the control law that satisfies (4) is such that there does not exist a time $0 < T < \infty$ so that $x(T) \in \Gamma$.

Hence we have $\min_{i \in \mathcal{P}}(\{h_i(x(t))\}) < 0$. From Boyd and Vandenberghe (2004), p. 72, we therefore have

$$-\ln\left(\sum_{i \in \mathcal{P}} \exp(-h_i(x(t))) \right) < 0$$

for all $t \geq 0$. Since $\tanh(x) < 0$ for all $x < 0$, we have for all $t \geq 0$

$$\tanh\left(-\ln\left(\sum_{i \in \mathcal{P}} \exp(-h_i(x(t))) \right) \right) = -\beta < 0$$

Observe that the inequality (4) can be rewritten as

$$\frac{d}{dt} \sum_{i \in \mathcal{P}} h_i(x(t)) \geq \gamma \cdot \beta \quad (5)$$

By integration of (5) using the fundamental theorem of calculus and from (4), we thus get

$$\sum_{i \in \mathcal{P}} h_i(x(t)) \geq \gamma \cdot \beta \cdot t + \sum_{i \in \mathcal{P}} h_i(x(0))$$

Observe that as $t \to \infty$, $\sum_{i \in \mathcal{P}} h_i(x(t)) \to \infty$. However, this is a contradiction since the barrier functions are bounded, and hence we have $\sum_{i \in \mathcal{P}} h_i(x(t)) < \sum_{i \in \mathcal{P}} M_i$ for all $t \geq 0$.
Hence there exists a $0 < T < \infty$ such that $x(T) \in \Gamma$. ■

The following theorem reformulates the constraint (4) taking into account the transition periods of the transition functions given a sequence of n tasks to be executed by the system.

Theorem 1. *Consider m bounded finite-time barrier functions, i.e., $h_i(x) \leq M_i$ for all $x \in \mathcal{D}$ and where $M_i > 0$ for all $i \in \{1, 2, \ldots, m\}$. Given a sequence of tasks $\mathcal{T} := \{\mathcal{T}_{(\Gamma_1, \Sigma)}, \ldots, \mathcal{T}_{(\Gamma_n, \Sigma)}\}$ with the corresponding transition functions $\alpha_j(t)$ chosen according to Algorithm 2 for all $j \in \Delta(\Gamma_i)$, for all $i \in \{1, \ldots, n\}$, and for all $t \geq 0$, if $u : \mathbb{R}_{\geq 0} \times \mathcal{D} \to \mathbb{R}^m$ is a continuous controller such that for all $x \in \mathcal{D}$ and $t \geq 0$, we have*

$$\sum_{i=1}^m \left\{ \alpha_i(t)(L_f h_i(x(t)) + L_g h_i(x(t)) u(t, x(t))) \right\} +$$
$$\sum_{i=1}^m h_i(x(t)) \frac{\partial \alpha_i(t)}{\partial t}$$
$$\geq -\gamma \cdot \tanh\left(-\ln\left(\sum_{i=1}^m \exp(-\alpha_i(t) h_i(x(t))) \right) \right), \quad (6)$$

then there exists a sequence of finite time instances $0 < T_1 < T_2 < \ldots < T_n < \infty$ such that $x(T_i) \in \Gamma_i$ for all $i \in \{1, \ldots, n\}$ i.e. the task sequence $\mathcal{T}$ is solved.

Proof. The proof is analogous to the proof of Lemma 1. Consider the first task $\mathcal{T}_{(\Gamma_1, \Sigma_1)}$. From Lemma 1 we know that there exists a finite time $0 < T_1 < \infty$ such that $x(T_1) \in \Gamma_1$. Suppose by contradiction, for some $x(T_1) \in \mathcal{D} \backslash \Gamma_2$, the constraint (6) is satisfied for all $t \geq T_1$ but there does not exist a finite time $0 < T_2 < \infty$ such that $x(T_2) \in \Gamma_2$. Hence we have that $\min_{i \in \{1,2,\ldots,m\}} (\{\alpha_i(t)h_i(x(t))\}) = \min_{i \in \Delta(\Gamma_2)} (\{\alpha_i(t)h_i(x(t))\}) < 0$. Following a proof methodology similar to Lemma 1, we can prove that there exists a finite time $0 < T_2 < \infty$ such that $x(T_2) \in \Gamma_2$. Following a successive proof by induction for tasks 3 to n, we can thus conclude that there exists a sequence of finite time instances $0 < T_1 < T_2 < \ldots < T_n < \infty$ such that $x(T_i) \in \Gamma_i$ for all $i \in \{1, 2, \ldots, n\}$ i.e. the task sequence $\mathcal{T}$ is solved. $\blacksquare$

Since we are interested in guaranteeing continuity of the controller, the following theorem proves that the proposed controller (3) as used in Algorithm 1 is continuous.

Theorem 2. *If Algorithm 1 is feasible for all $t \geq 0$, then the control input, computed by solving (3), applied to the system is continuous.*

Proof. Consider an indicator variable for the time, θ with $\dot{\theta} = 1$. Considering the new state of the system as $\hat{x} = [x \ \theta]^T$, the constraint (6) can be reformulated as

$$\sum_{i=1}^{m} \left\{ \alpha_i(\hat{x})(L_f h_i(\hat{x}) + L_g h_i(\hat{x})u(\hat{x})) \right\} + \sum_{i=1}^{m} h_i(\hat{x}) \frac{\partial \alpha_i(\hat{x})}{\partial \theta} +$$
$$\gamma \cdot \tanh\left(-\ln\left(\sum_{i=1}^{m} \exp(-\alpha_i(\hat{x})h_i(\hat{x})) \right) \right) \geq 0.$$

The reformulated inequality is now the barrier function constraint (4) for the new time invariant system with the augmented state $\hat{x}$. Similarly, the constraint $u(x) \in \mathcal{U}_{\text{safe}}(x)$ can be reformulated in terms of the new state. These constraints are quasi-convex in the control u for all $\hat{x} \in \mathbb{R}^{n+1}$ and the cost is quasi-convex. In addition, the input is constrained over a compact set (third constraint in (3)). However, since $\alpha_i(\hat{x})$ for all $i \in \mathcal{I}$ is chosen point-wise in time as per Algorithm 2, the continuity of the controller must be established by analyzing the transition phase, reachability phase, and the time instant at which the transition between phases occurs as per Algorithm 1. Since all the assumptions of Proposition 8 in Konda et al. (2019) are satisfied, the controller computed from Algorithm 1 is continuous for the entirety of the transition and the reachability phase. Since the functions $\kappa^{\uparrow}$ and $\kappa^{\downarrow}$ are chosen such that, $\kappa^{\uparrow}(t - T_i) |_{t=T_i} = 0$ and $\kappa^{\downarrow}(t - T_i) |_{t=T_i} = 1$, the constraint (6) does not suffer from discontinuities when switching between the reachability and the transition phase. Thus, the controller computed as per Algorithm 1 is continuous for all $t \geq 0$. $\blacksquare$

5. ROBOTIC IMPLEMENTATION

In this section, we present the implementation [1] of the theoretical framework discussed in Section 4 in the Georgia Tech Robotarium by Pickem et al. (2017).

Consider a differential drive mobile robot with dynamics
$$\dot{x} = v \cdot cos(\phi)$$
$$\dot{y} = v \cdot sin(\phi)$$
$$\dot{\phi} = \omega$$
where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are the position coordinates of the robot, $\phi \in [-\pi, \pi)$ is the orientation, $v \in \mathbb{R}$ is the linear velocity input, and ω is the angular velocity input. Denote $\tilde{x} = [x \ y \ \phi]^T$, $\hat{x} = [x \ y]^T$, and $\mathcal{D} \subset \mathbb{R}^3$. Then we have $f(\tilde{x}) = 0$ and $g(\tilde{x}) = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix}$.

Let $u = [v \ \omega]^T$ be the input vector to the robot. The experiment was conducted on the Robotarium multi-robot testbed at Georgia Tech. For more details on the hardware specifications and platform setup, please refer Pickem et al. (2017). Since the differential drive robot model is non-holonomic which leads to controllability issues, we use a technique known as the Near Identity Diffeomorphism (NID) to control the robot, as detailed in Pickem et al. (2017).

The workspace consists of four regions of interest - three goal regions (A, B and C) and an obstacle O. Fig. 2 illustrates the domain of interest. Here, the robot must initially visit region A, followed by region B and ultimately region C while avoiding the obstacle. The sequence of tasks is given by $\mathcal{T} = \{\mathcal{T}_{(\Gamma_A, \Sigma_O)}, \mathcal{T}_{(\Gamma_B, \Sigma_O)}, \mathcal{T}_{(\Gamma_C, \Sigma_O)}\}$, where $\Gamma_A = \{\tilde{x} \in \mathcal{D} \mid h_1(\tilde{x}) \geq 0\}$, $\Gamma_B = \{\tilde{x} \in \mathcal{D} \mid h_2(\tilde{x}) \geq 0\}$, $\Gamma_C = \{\tilde{x} \in \mathcal{D} \mid h_3(\tilde{x}) \geq 0\}$, and $\Sigma_O = \{\tilde{x} \in \mathcal{D} \mid h_4(\tilde{x}) \geq 0\}$, where each $h_i : \mathcal{D} \to \mathbb{R}$ for all $i \in \{1, 2, 3, 4\}$ is a continuously differentiable function. For regions A, B and C, the functions are given by $h_1(\tilde{x}) = 1 - (\hat{x} - C_A)^T P_A(\hat{x} - C_A)$, $h_2(\tilde{x}) = 1 - (\hat{x} - C_B)^T P_B(\hat{x} - C_B)$ and $h_3(\tilde{x}) = 1 - (\hat{x} - C_C)^T P_C(\hat{x} - C_C)$ where P_A, P_B, and P_C are diagonal matrices with the inverses of the dimensions of the ellipsoids as the non-zero entries, and C_A, C_B and C_C are the centers of each ellipsoidal region. The obstacle is modeled using a weighted polar L_p function as described in Srinivasan et al. (2019) with parameters $p = 6$, $\sigma = (0.7, 0.2)$, $\theta_\kappa = \frac{\pi}{2}$, $c = 1$ and $\theta_0 = \text{sign}(\kappa) \cdot \frac{\pi}{2}$.

At each time step t, Algorithm 1 is executed and the control input is applied to the robot. The QP that is solved to execute the entire sequence of tasks is

$$u^*(t, \tilde{x}) = \min_{u \in \mathbb{R}^2} \quad ||u||_2^2$$

$$\text{s.t} \ \sum_{i=1}^{3} \left\{ \frac{d\alpha_i(t)h_i(\tilde{x}(t))}{dt} \right\}$$
$$\geq -\gamma \cdot \tanh\left(-\ln\left(\sum_{i=1}^{3} \exp(-\alpha_i(t)h_i(\tilde{x}(t))) \right) \right)$$
$$L_f h_4(\tilde{x}(t)) + L_g h_4(\tilde{x}(t))u(t, \tilde{x}(t)) \geq -\gamma h_4(\tilde{x}(t))^3$$
$$||u||_\infty \leq 10$$

where $L_f h_i(\tilde{x}(t)) = \frac{\partial h_i(\tilde{x}(t))}{\partial \tilde{x}} f(\tilde{x}) = 0$ and $L_g h_i(\tilde{x}(t)) =$

---

[1] https://github.com/gtfactslab/2020_ContinuousTaskBarriers
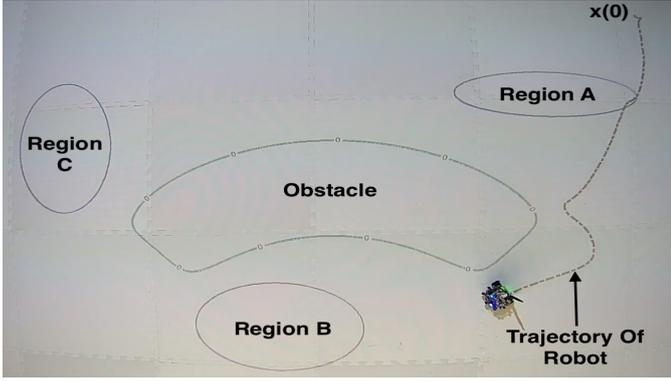
Fig. 2. A still shot of the implementation of Algorithm 1 conducted on the Robotarium. The robot first visits region A, followed by region B, and lastly region C, while avoiding the obstacle.
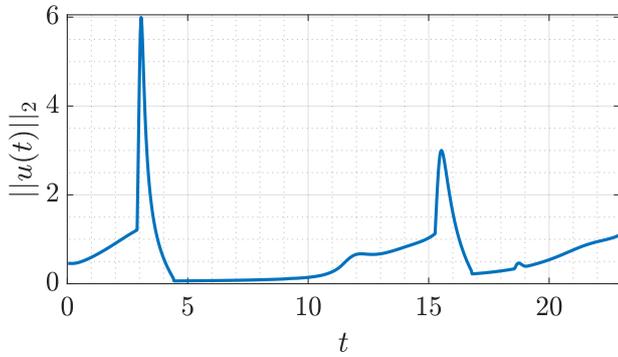
Fig. 3. The control input u generated in the Robotarium simulator for the task specification as discussed in the experimental setup. Observe that the control generated by our framework is continuous, which is in contrast to Fig 1.

$\frac{\partial h_i(\tilde{x}(t))}{\partial \tilde{x}} g(\tilde{x})$ for all $i \in \{1,2,3,4\}$. The average time to solve the QP was between 3ms to 7ms. The increasing and decreasing functions in Algorithm 2 are chosen as $\kappa^\uparrow = \sin^2(t - T_i)$ and $\kappa^\downarrow(t) = \cos^2(t - T_i)$ respectively, where T_i is the recorded time instant at which the robot reaches region i. Fig. 3 demonstrates the continuous change in the solution space as the robot is executing the QP. Observe that the discontinuous switching in the control input is avoided due to the modified constraint in (6). This is in contrast to the discontinuities shown in Fig 1 where the traditional discrete QP switching method was used. A video of the implementation is provided. [2]

6. CONCLUSION

We proposed a control barrier function based method to ensure smooth transition between different reachability tasks for a control-affine robotic system. A new composite barrier function constraint was introduced by endowing individual barrier functions with time varying transition functions, which allow for smooth transitions between different objectives. In order to facilitate real-time implementation capabilities, we proposed an algorithm which incorporates the proposed barrier function constraint. Lastly,

we proved that the proposed controller is continuous. Robotic implementation results are also provided which validate the proposed theory.

REFERENCES

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, 3420–3431. doi: 10.23919/ECC.2019.8796030.

Ames, A.D., Grizzle, J.W., and Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, 6271–6278. doi: 10.1109/CDC.2014.7040372.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Han, H. and Park, J. (2013). Robot control near singularity and joint limit using a continuous task transition algorithm. *International Journal of Advanced Robotic Systems*, 10(10), 346. doi:10.5772/56714. URL https://doi.org/10.5772/56714.

Jarquín, G., Escande, A., Arechavaleta, G., Moulard, T., Yoshida, E., and Parra-Vega, V. (2013). Real-time smooth task transitions for hierarchical inverse kinematics. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 528–533. doi:10.1109/HUMANOIDS.2013.7030024.

Kim, S., Jang, K., Park, S., Lee, Y., Lee, S.Y., and Park, J. (2019). Continuous task transition approach for robot controller based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 4(2), 1603–1610. doi:10.1109/LRA.2019.2896769.

Konda, R., Ames, A.D., and Coogan, S. (2019). Characterizing Safety: Minimal Barrier Functions from Scalar Comparison Systems. *arXiv e-prints*, arXiv:1908.09323.

Li, A., Wang, L., Pierpaoli, P., and Egerstedt, M. (2018). Formally correct composition of coordinated behaviors using control barrier certificates. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3723–3729. IEEE.

Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M. (2017). The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1699–1706. doi:10.1109/ICRA.2017.7989200.

Pierpaoli, P., Li, A., Srinivasan, M., Cai, X., Coogan, S., and Egerstedt, M. (2019). A Sequential Composition Framework for Coordinating Multi-Robot Behaviors. *arXiv e-prints*, arXiv:1907.07718.

Srinivasan, M., Hyun, N.P., and Coogan, S. (2019). Weighted polar finite time control barrier functions with applications to multi-robot systems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 7031–7036.

Srinivasan, M. and Coogan, S. (2019). Control Of Mobile Robots Using Barrier Functions Under Temporal Logic Specifications. *arXiv e-prints*, arXiv:1908.04903.

Wang, L., Ames, A., and Egerstedt, M. (2016). Safety barrier certificates for heterogeneous multi-robot systems. In *2016 American Control Conference (ACC)*, 5213–5218. doi:10.1109/ACC.2016.7526486.

[2] Video of the implementation- https://youtu.be/eKhXiJkQH8w