Control Of Mobile Robots Using Barrier Functions Under Temporal Logic Specifications

Mohit Srinivasan, *Student Member, IEEE*, and Samuel Coogan, *Member, IEEE*

***Abstract*—In this paper, we propose a framework for the control of mobile robots subject to temporal logic specifications using barrier functions. Complex task specifications can be conveniently encoded using linear temporal logic (LTL). In particular, we consider a fragment of LTL which encompasses a large class of motion planning specifications for a robotic system. Control barrier functions (CBFs) have recently emerged as a convenient tool to guarantee reachability and safety for a system. In addition, they can be encoded as affine constraints in a quadratic program (QP). In the case of complex system specifications, we show that following QP based methods in existing literature can lead to infeasibility and hence we provide a method of composition of multiple barrier functions in order to mitigate infeasibility. A scheme to prioritize different barrier functions which allows the user to encode the notion of priority based control, is also introduced. We prove that the resulting system trajectory synthesized by the proposed controller satisfies the given specification. Robotic simulation and experimental results are provided in addition to the theoretical framework.**

***Index Terms*—Control Barrier Functions, Linear Temporal Logic, Mobile Robots, Quadratic Programs.**

I. Introduction

System specifications to be satisfied by mobile robotic systems are increasing in complexity. For example, motion planning for systems such as robotic manipulators [1], personal assistants [2], and quadrotors [3] involves complex specifications to be satisfied by the system. Safety critical systems such as the power grid [4] and automation floors [5] rely on distributed controllers in order to function in the desired manner. These controllers are again tasked with satisfying complex specifications. Hence, failure of these controllers can lead to a collapse of the safety critical infrastructure [6]. To that end, synthesizing controllers with formal guarantees on their correct functioning is of key importance.

In this paper, we present a control architecture for the control of mobile robotic systems subject to linear temporal logic specifications using control barrier functions, which addresses some of the challenges in the previously discussed applications. In particular, we address the issue of situations where proposed methods in existing literature can render the controller infeasible. With the synthesis of the controller, we then shift focus towards providing formal guarantees regarding the proposed controller framework. In particular, we prove that the system trajectory generated by the proposed controller satisfies the given specification.

This work was supported in part by DARPA under Grant N66001-17-2-4059 and the National Science Foundation under Grant 1749357

Mohit Srinivasan is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332 USA (e-mail: mohit.srinivasan@gatech.edu).

Samuel Coogan is with the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332 (email: sam.coogan@gatech.edu)

Manuscript received month day, 20xx; revised month day, 20xx.

A. Background

Barrier functions were first introduced in optimization. A historical account of their use can be found in Chapter 3 in [7]. Usage of barrier functions is now common throughout the control, verification and robotics literature due to their natural relationship with Lyapunov-like functions. Control barrier function (CBF) based quadratic programs (QPs) were first used in [8], [9] in the context of automotive applications such as adaptive cruise control (ACC). Recently, control barrier functions have been used in the context of multi-agent systems to guarantee collision avoidance between robots [3], [10], [11]. Given a minimum distance to be maintained between the robots, the safety set is encoded as the super zero level set of a zeroing control barrier function (ZCBF) [9]. The authors then use a QP based controller with the ZCBFs as affine constraints in order to guarantee forward invariance of this safety set. This in turn implies that the robots never collide. Such a framework has also been applied to quadrotors [3] where the safety set is considered to be a super ellipsoid which allows quadrotors to avoid collisions with each other.

ZCBFs guarantee asymptotic convergence to desired sets [9]. However, since we focus on motion planning specifications, we require finite time reachability guarantees. Recently, [12], [13] have introduced finite time control barrier functions for finite time reachability specifications. In [12], finite time barrier functions were used to achieve smooth transitions between different behaviors in a multi-agent system. The key objective in [12] was to ensure composability of different formation behaviors by making sure that the multi-agent communication graph is appropriate for the next desired formation, whereas in [13], a method for the composition of multiple finite time barrier functions was introduced.

Finite and infinite horizon specifications which are useful for mobile robotic systems can be conveniently encoded using linear temporal logic (LTL). The power of LTL originates from the wealth of tools available in the model checking literature [14] which allows for generating trajectories for the robots given a specification in temporal logic. LTL based control of robotic systems has been well studied and standard methods first create a finite abstraction of the original dynamical system [15], [16], [17], [18]. This abstraction can informally be viewed as a labeled graph that represents possible behaviors of the system. Given such a finite abstraction, controllers can be automatically constructed using an automata-based approach [17], [14], [19], [20]. However, abstracting the state space is computationally expensive especially with complex system dynamics and specifications.

In our framework, we avoid the difficulties associated with computation of any automaton from the specification or a discretization of the state space. Since CBFs can be conveniently encoded within a QP, the controller is amenable to real time implementations without the need for an abstraction of the state space or the system dynamics. Other authors have explored discretization free techniques as well. The authors in [21] discuss the use of time varying control barrier functions for signal temporal logic tasks (STL). By using a time varying barrier function which shrinks the invariant set with time, the authors guarantee finite time reachability as well as safety. The authors in [22], [23], [24] discuss control methods for STL tasks. However, the methods proposed result in computationally expensive mixed integer linear programs. Control methods in the discrete time non-deterministic setting have been explored by [25]. Learning based frameworks are discussed by the authors in [26], [27], [28]. Control techniques for continuous-time multi-agent systems given fragment of STL tasks has been presented in [29]. The authors in [30] discuss a similar continuous time method. However, a non-convex optimization problem may have to be solved.

B. Contributions

There are two primary contributions of this work. The first and main contribution is an automatic framework which synthesizes a control barrier functions (CBFs) based quadratic program (QP) controller given a user defined specification. Then, we provide formal guarantees that the CBFs based QP controller produces a system trajectory that satisfies the given specification. The trajectory generated by the proposed controller is analyzed and the guarantees of CBFs translate to guarantees on the trajectory.

Second, we address the issue of controller infeasibility, a common difficulty in CBF based real-time control, in case of complex system specifications. We illustrate a situation where the method of encoding multiple finite time reachability objectives individually in a QP based CBF framework such as [12] fails. We show that encoding each reachability specification as a separate constraint in the QP is too restrictive when the system needs to execute complex specifications. Hence, a relaxed version of finite time reachability for multiple reachability objectives is required. We therefore propose a composition of multiple finite time control barrier functions which yields a larger solution space of the corresponding QP than existing methods in literature.

We also address the issue of infeasibility when some barrier functions are in conflict with others. In particular, we propose a prioritization scheme, similar to the method discussed in [31], in order to relax the zeroing control barrier functions. By including the relaxation parameter in the cost, the QP will satisfy the finite time reachability constraint while minimally violating the invariance constraints (if they are in conflict with the reachability constraint). We would like to emphasize that this method is not the main focus of this paper. However, it is a reformulation of [9] and [31] in order to account for finite time barrier function constraints. These contributions are detailed in Section IV.

A preliminary version of this work was presented in our conference paper [13] where we formulated the notion of composition of multiple finite time control barrier functions. In the present paper, we extend those results in order to synthesize an automated framework (full solution) to transition from a specification belonging to a fragment of LTL to the barrier function based controller.

This paper is organized as follows. Section II introduces control barrier functions, linear temporal logic and the quadratic program based controller used for trajectory generation of the system. In Section III, we discuss the problem statement that is addressed in this paper. Section IV introduces the idea of composite finite time control barrier functions [13] and the prioritization scheme for different zeroing barrier functions. In Section V, we propose the QP based controller and develop the theoretical framework which provides a formal guarantee that the proposed controller synthesizes a system trajectory that satisfies the given specification. Section VI discusses a multi-agent system case study with simulation and experimental results. Section VII provides concluding remarks.

II. Mathematical Background

In this section, we provide background on control barrier functions (CBFs) and the guarantees on invariance and reachability of sets obtained from them, linear temporal logic (LTL) which is the specification language, and the quadratic program (QP) based controller with the CBFs as constraints which will be used to synthesize the trajectory for a control affine robotic system.

A. Control Barrier Functions (CBFs)

Consider a continuous time control affine dynamical system

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

where f and g are locally Lipschitz continuous, $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state of the system, and $u \in \mathbb{R}^m$ is the control input applied to the system.

Before we introduce the notion of control barrier functions, we define an extended class $\mathcal{K}$ function [32] $\alpha : \mathbb{R} \to \mathbb{R}$ as a function that is strictly increasing and $\alpha(0) = 0$.

Definition 1 (Zeroing Control Barrier Function (ZCBF))**.** *A continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ is a* zeroing control barrier function (ZCBF) *if there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that for all $x \in \mathcal{X}$,*

$$\sup_{u \in \mathbb{R}^m} \left\{ L_f h(x) + L_g h(x) u + \alpha(h(x)) \right\} \geq 0 \tag{2}$$

where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$ are the Lie derivatives of h along f and g respectively. □

Let the set of control inputs that satisfy (2) at any given state $x \in \mathcal{X}$ be defined as

$$\mathcal{U}_\Sigma(x) = \left\{ u \in \mathbb{R}^m | L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0 \right\}. \quad (3)$$

One can guarantee forward invariance of desired sets under the existence of a suitable zeroing control barrier function as formalized in the following proposition.

Proposition 1 (Corollary 1, [9])**.** *Let $\Sigma \subseteq \mathcal{X}$ be a safety set defined as $\Sigma = \{x \in \mathcal{X} | h(x) \geq 0\}$ where $h : \mathcal{X} \to \mathbb{R}$. If h is a zeroing control barrier function, then any Lipschitz continuous feedback controller satisfying $u \in \mathcal{U}_\Sigma(x)$ renders the set Σ forward invariant for the system* (1). ■

We now define finite time convergence control barrier functions which guarantee finite time convergence to desired sets in the state space.

Definition 2 (Finite Time Convergence Control Barrier Function (FCBF))**.** *A continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ is a* finite time convergence control barrier function *if there exist parameters $\rho \in [0,1)$ and $\gamma > 0$ such that for all $x \in \mathcal{X}$,*

$$\sup_{u \in \mathbb{R}^m} \left\{ L_f h(x) + L_g h(x) u + \gamma \cdot sign(h(x)) \cdot |h(x)|^\rho \right\} \geq 0 \quad (4)$$

where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$. □

Let the set of control inputs that satisfy (4) at any state $x \in \mathcal{X}$ be given by

$$\mathcal{U}_\Gamma(x) = \left\{ u \in \mathbb{R}^m \middle| L_f h(x) + L_g h(x) u + \gamma \cdot sign(h(x)) \cdot |h(x)|^\rho \geq 0 \right\} \quad (5)$$

If h is a finite time convergence control barrier function, then there exists a control input u that drives the state of the system x to the target set $\{x \in \mathcal{X} | h(x) \geq 0\}$ in finite time, as formalized next.

Proposition 2 (Proposition III.1, [12])**.** *Let $\Gamma \subseteq \mathcal{X}$ be a target set defined as $\Gamma = \{x \in \mathcal{X} | h(x) \geq 0\}$ where $h : \mathcal{X} \to \mathbb{R}$. If h is a finite time convergence control barrier function for* (1)*, then, for any initial condition $x_0 \in \mathcal{X}$ and any Lipschitz continuous feedback control $u : \mathcal{X} \to \mathbb{R}^m$ satisfying $u \in \mathcal{U}_\Gamma(x)$ for all $x \in \mathcal{X}$, the system will be driven to the set Γ in a finite time $0 < T < \infty$ such that $x(T) \in \Gamma$, where the time bound is given by $T = \frac{|h(x_0)|^{1-\rho}}{\gamma(1-\rho)}$. Moreover, Γ is forward invariant so that the system remains in Γ for all $t \geq T$.* ■

Zeroing barrier functions and finite time barrier functions will form the basis for our control synthesis methodology. Next, we discuss the temporal language used to specify complex robotic system specifications in our framework.

B. Linear Temporal Logic

Complex and rich system properties can be expressed succinctly using linear temporal logic (LTL). The power of LTL lies in the wealth of tools available in the model checking literature [14] which can be leveraged for the synthesis of controllers in the continuous domain. LTL formulas are developed using atomic propositions which label regions of interest within the state space. These formulas are built using a specific grammar. LTL formulas without the next operator are given by the following grammar [14]:

$$\phi = \pi | \neg\phi | \phi \vee \phi | \phi \mathcal{U} \phi \quad (6)$$

where π is a member of the set of atomic propositions denoted by Π, and ϕ is a propositional formula that represents an LTL specification.

We use the standard graphical notation for the temporal operators including $\Box$ ("Always"), $\Diamond$ ("Eventually"), $\Diamond\Box$ ("Persistence") and $\Box\Diamond$ ("Recurrence"). From the negation ($\neg$) and the disjunction ($\vee$) operators, we can define the conjunction ($\wedge$), implication ($\rightarrow$), and equivalence ($\leftrightarrow$) operators. We can thus derive for example, the eventually ($\Diamond$) and always ($\Box$) operators as $\Diamond\phi = \top\mathcal{U}\phi$ and $\Box\phi = \neg\Diamond\neg\phi$ respectively. Below we provide informal interpretations of these operators with respect to an LTL formula ϕ.

- $\Diamond\phi$ is satisfied if ϕ is satisfied sometime in the future. That is, ϕ is satisfied in some future time step.
- $\Box\phi$ is satisfied if ϕ is satisfied for all time. That is, ϕ is satisfied at each time step.
- $\Diamond\Box\phi$ is satisfied if ϕ becomes satisfied at some time step in the future and remains satisfied for all following steps.
- $\Box\Diamond\phi$ is satisfied if ϕ is satisfied infinitely often at some time step in the future.

Next we discuss the QP based controller which will be used for the synthesis of the system trajectory.

C. Quadratic Program (QP) based controller

Given a finite time convergence control barrier function or a zeroing control barrier function h, the constraints (2) and (4) are affine in the control input u, and hence they can be conveniently encoded as affine constraints in a QP. Hence this formulation is amenable to efficient online computation of feasible control inputs. In particular, for fixed $x \in \mathcal{X}$, the requirement that $u \in \mathcal{U}_\Gamma(x)$ and/or $u \in \mathcal{U}_\Sigma(x)$ becomes a linear constraint and we define a minimum energy quadratic program (QP) as

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & ||u||_2^2 \\ \text{s.t} \quad & u \in \mathcal{U}_\Gamma(x) \text{ and/or } u \in \mathcal{U}_\Sigma(x). \end{aligned} \quad (7)$$

We note that (7) can encode both finite time reachability as well as forward invariance requirements as constraints in the QP. This QP when solved returns the pointwise minimum energy control law that drives the system to the goal set Γ in finite time and/or guarantees invariance of Σ. We will reference this idea of a QP based controller throughout this paper in the context of our theoretical framework.

Remark 1. *We note that multiple zeroing control barrier functions and multiple finite time barrier functions can be encoded as separate constraints in the QP. In this case, we solve a single QP with multiple barrier function constraints. For example, see [9], [12].*

Next, we formulate the problem statement that is addressed in this paper.

III. Problem Statement

In this paper, we consider a continuous time mobile robotic system in control affine form

$$\dot{x} = f(x) + g(x)u, \tag{8}$$

where f and g are locally Lipschitz continuous, $x \in \mathcal{X} \subseteq \mathbb{R}^n$, and $u \in \mathbb{R}^m$. Assume $\mathcal{X}$ to be the compact domain in the state space for the system.

We assume that $\mathcal{X}$ is divided into regions of interest which are labeled by a set of atomic propositions $\Pi = \{\pi_1, \pi_2, \pi_3, \dots, \pi_n\}$ with the labeling function $L : \mathcal{X} \to 2^\Pi$ so that $\pi \in \Pi$ is true at $x \in \mathcal{X}$ if and only if $\pi \in L(x)$. For each $\sigma \in 2^\Pi$, we have $L^{-1}(\sigma) = \{x \in \mathcal{X} | \sigma = L(x)\}$. Let $\Pi_{aug} = \{\pi_1, \pi_2, \dots, \pi_n, \overline{\pi_1}, \overline{\pi_2}, \dots, \overline{\pi_n}\}$ be the augmented set of atomic propositions where we define $\overline{\pi_i} = \neg\pi_i$ for all $i \in \{1, 2, \dots, n\}$. The set Π_{aug} is also called the set of *literals* [14]. Thus, we identify $\neg\overline{\pi}_i = \pi_i$ for all $i \in \{1, 2, \dots, n\}$. In addition, define

$$S(\Pi_{aug}) = \{J \subset \Pi_{aug} | \pi \in J \implies \neg\pi \notin J \text{ for all } \pi \in \Pi_{aug}\} \tag{9}$$

$$P(\Pi_{aug}) = \{J \subset \Pi_{aug} | (\pi_i \in J) \oplus (\overline{\pi_i} \in J) \text{ for all } i \in \{1, 2, \dots, n\}\} \tag{10}$$

where $\oplus$ is the exclusive disjunction operator. Observe that $P(\Pi_{aug}) \subset S(\Pi_{aug})$. A subset of Π_{aug} belongs to the family $S(\Pi_{aug})$ if it does not contain an atomic proposition and its negation simultaneously, and it further belongs to $P(\Pi_{aug})$ if it contains each atomic proposition exclusive or its negation.

We consider a fragment of LTL, denoted by $LTL_{robotic}$, that covers a large class of motion planning tasks expected from a robotic system. Our proposed fragment is a modification of the one considered in [33].

Definition 3 (Fragment of LTL)**.** *The fragment $LTL_{robotic}$ is defined as the class of LTL specifications of the form*

$$\phi = \phi_{globe} \wedge \phi_{reach} \wedge \phi_{rec} \wedge \phi_{act} \tag{11}$$

where $\phi_{globe} = \Box\psi_1$, $\phi_{reach} = \bigwedge_{j \in \mathcal{I}_2} \Diamond\psi_2^j$, $\phi_{rec} = \bigwedge_{j \in \mathcal{I}_3} \Box\Diamond\psi_3^j$ and $\phi_{act} = \Diamond\Box\psi_4$. Here $\mathcal{I}_2$ and $\mathcal{I}_3$ are finite index sets and ψ_1, ψ_2^j for all j, ψ_3^j for all j and ψ_4 are propositional formulas of the form $\psi_i = \bigwedge_{\forall\pi \in J_i} \pi$ with $J_i \in S(\Pi_{aug})$ for all $i \in \{1, 4\}$, $\psi_i^j = \bigwedge_{\forall\pi \in J_i^j} \pi$ with $J_i^j \in S(\Pi_{aug})$ for all $i \in \{2, 3\}$ and for all $j \in \mathcal{I}_i$. □

For any propositional formula ψ, we define the following.

Definition 4 (Proposition Set)**.** *The* proposition set *for a propositional formula ψ, denoted $[\![\psi]\!]$, is the set of all states that satisfy ψ. That is,*

$$[\![\psi]\!] = \{x \in \mathcal{X} | L(x) \models \psi\} \tag{12}$$

where $L(x) \models \psi$ signifies that ψ is true under the evaluation for which all and only propositions in $L(x) \subset \Pi$ are true. □

We assume that for each atomic proposition $\pi \in \Pi$, there exists a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ such that $[\![\pi]\!] = \{x \in \mathcal{X} | h_\pi(x) \geq 0\}$. In this paper, similar to the assumption in [9], we assume that $L_g h_\pi(x) \neq 0$ for all $x \in \mathcal{X}$. We ignore the measure-zero set $\{x \in \mathcal{X} | h_\pi(x) = 0\}$, and identify $[\![\overline{\pi}]\!] = \{x \in \mathcal{X} | h_\pi(x) < 0\}$ for each $\pi \in \Pi$. Thus we define $h_{\overline{\pi}}(x) = -h_\pi(x)$ for all $\pi \in \Pi$.

In addition, with a slight overload of the operator $[\![\cdot]\!]$, for any $\sigma \subseteq \Pi_{aug}$, we define

$$[\![\sigma]\!] \triangleq [\![\bigwedge_{\pi \in \sigma} \pi]\!] = \bigcap_{\pi \in \sigma} [\![\pi]\!] = \bigcap_{\pi \in \sigma} \{x \in \mathcal{X} | h_\pi \geq 0\} \subset \mathcal{X}. \tag{13}$$

Intuitively, (13) represents the set of states such that the labeling function evaluated at these states returns all and only those propositions belonging to σ.

The fragment $LTL_{robotic}$ encompasses a class of specifications which cover properties such as finite time reachability, persistence, recurrence, and invariance. These properties are useful to express a number of common robotic system specifications. We observe that the propositional formulas ψ appearing in Definition 3 are in positive normal form (PNF) and do not include the disjunction operator. We now define the problem statement that is addressed in this paper.

Problem Statement. *Given a specification in $LTL_{robotic}$ as in* (11) *which is to be satisfied by a mobile robotic system with dynamics as in* (8)*, synthesize a controller which produces a system trajectory that satisfies the given specification.*

Before we detail the theoretical framework to address the above problem statement, we discuss scenarios where the QP based controller could be infeasible.

IV. Feasibility of QP based Controller

Given a specification ϕ in $LTL_{robotic}$, in this section, we focus on scenarios where using existing methods in literature [9], [12], [34] will render the controller infeasible, and provide solutions for the same. Subsection A discusses Theorem 1 which appeared in our conference paper [13], while subsection B proposes a relaxed formulation of the QP based controller.

A. Composite Finite Time Control Barrier Functions

Consider two robots R_1 and R_2 as shown in the workspace in Fig 1. Suppose R_1 is sensing information from R_2 and hence must always stay within the sensing radius of R_2. Suppose we have two regions of interest A, B and the base C. Let $\mathcal{D}$ represent a corridor in the state space (denoted by the dotted lines in Fig 1) where R_1 must maintain a very small distance of connectivity with R_2. This could represent, for example, an area with very poor network connectivity and hence the robots must resort to communication over small distances. Let the specification for the multi-agent system be given as $\phi = \Diamond(\pi_1^A \wedge \pi_2^B) \wedge \Diamond(\pi_1^C \wedge \pi_2^C) \wedge \Box\pi_{conn}$ where π_1^A is the proposition that is true when R_1 is in A, π_2^B is the proposition that is true when R_2 is in B, and π_{conn} is the proposition that is true when the robots must maintain connectivity at all times. In other words, R_1 must visit A, R_2 must visit B and then both must return to the base C. In addition, R_1 must always stay connected with R_2. The workspace is as shown in Fig 1.

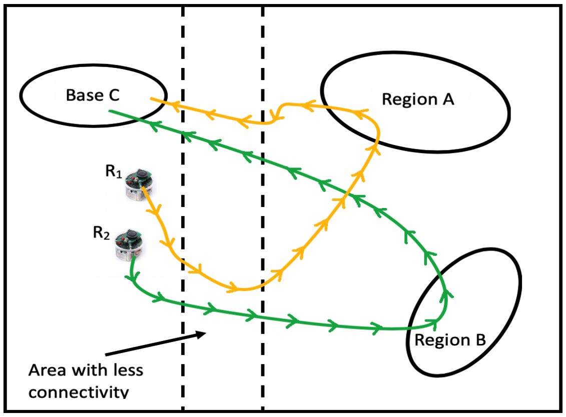

Figure 1: Representative trajectories for R_1 and R_2 that satisfy the specification $\phi = \Diamond(\pi_1^A \wedge \pi_2^B) \wedge \Diamond(\pi_1^C \wedge \pi_2^C) \wedge \Box\pi_{conn}$. The area with less connectivity is the corridor $\mathcal{D}$. Observe that R_1 and R_2 need to maintain a small distance of connectivity within the corridor D.

By following the method proposed in [12], the QP that is to be solved is as follows:

$$\begin{aligned}
&\min_{u\in\mathbb{R}^4} \quad ||u||_2^2 \\
&\text{s.t} \quad L_f h_A(x_1) + L_g h_A(x_1)u \geq -\gamma \cdot sign(h_A(x_1)) \cdot |h_A(x_1)|^\rho \\
&\qquad L_f h_B(x_2) + L_g h_B(x_2)u \geq -\gamma \cdot sign(h_B(x_2)) \cdot |h_B(x_2)|^\rho \\
&\qquad L_f h_{conn}(x) + L_g h_{conn}(x)u \geq -\alpha(h_{conn}(x))
\end{aligned} \tag{14}$$

where α is a locally Lipschitz extended class κ function, $\gamma > 0$, $\rho \in [0,1)$, x_1 is the state of R_1, x_2 is the state of R_2, $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ is the total state of the system, h_A is the finite time barrier function which represents A, h_B is the finite time barrier function which represents B, and h_{conn} is the zeroing barrier function which dictates the connectivity radius to be maintained by R_1 with R_2.

However, this QP becomes infeasible at the point when R_1 and R_2 reach the corridor $\mathcal{D}$. This is because the first constraint in (14) dictates that R_1 make progress towards A, but the third constraint dictates that R_1 move closer to R_2 and hence move away from A. This leads to an empty solution space thus rendering the QP infeasible. This shows that the above formulation of encoding multiple reachability objectives as individual constraints is too restrictive.

In light of the above scenario, we propose a method in which we compose multiple finite time control barrier functions. By ensuring that the total sum of the finite time barrier functions is always increasing, we can allow for decrease in the values of some of the individual barrier functions thereby allowing some robots to move away from their desired sets temporarily. This provides a larger solution space for the QP. This is formalized in the following theorem.

Theorem 1. *Consider a dynamical system in control affine form as in (8). Given $\Gamma \subset \mathcal{X}$ defined by a collection of $q \geq 1$ functions $\{h_i(x)\}_{i=1}^q$ such that $\Gamma = \bigcap_{i=1}^q \{x \in \mathcal{X} | h_i(x) \geq 0\}$ and for $i = \{1,2,3,...,q'\}$ with $q' < q$, $h_i(x)$ is bounded i.e. $h_i(x) < M_i$ for all $x \in \mathcal{X}$, for $M_i > 0$.[1] If there exists a collection $\{\alpha_i\}_{i=1}^{q'}$ with $\alpha_i \in \mathbb{R}_{>0}$, parameters $\gamma > 0$, $\rho \in [0,1)$ and a continuous controller $u(x)$ where $u : \mathcal{X} \to \mathbb{R}^m$, such that for all $x \in \mathcal{X}$*

$$\sum_{i=1}^{q'} \Big\{ \alpha_i (L_f h_i(x) + L_g h_i(x) u(x)) \Big\} + \gamma \cdot sign\Big(\min \Big\{ h_1(x), h_2(x), \ldots, h_{q'}(x) \Big\} \Big) \geq 0 \tag{15}$$

$$L_f h_i(x) + L_g h_i(x) u(x) + \gamma \cdot sign(h_i(x)) \cdot |h_i(x)|^\rho \geq 0 \quad \forall\ i \in \{q'+1, \ldots, q\} \tag{16}$$

then under the feedback controller $u(x)$, for all initial conditions $x_0 \in \mathcal{D}$, there exists $0 < T < \infty$ such that $x(T) \in \Gamma$.

Proof. By contradiction, suppose for some $x_0 \in \mathcal{X} \backslash \Gamma$ the control law $u(x)$ that satisfies (15) and (16) is such that there does not exist a finite time $0 < T < \infty$ so that $x(T) \in \Gamma$. In particular, then for all $t > 0$, $min\Big\{ h_1(x(t)), h_2(x(t)), \ldots, h_q(x(t)) \Big\} < 0$, where $x(t)$ is the solution to (1) initialized at $x(0)$ under the control law $u(x)$. By (16) for all $t > T_i = \frac{|h_i(x_0)|^{1-\rho}}{\gamma(1-\rho)}$, we have $h_i(x(t)) \geq 0$ for all $i = \{q'+1, \ldots, q\}$ by Proposition 1. To that end, if we define $T' = \max_{i=q'+1,\ldots,q} \{T_i\}$, then for all $t > T'$ we have, $\min \Big\{ h_1(x(t)), h_2(x(t)), \ldots, h_{q'}(x(t)) \Big\} < 0$. In particular, observe that

$$\frac{d}{dt} \sum_{i=1}^{q'} \Big\{ \alpha_i h_i(x(t)) \Big\} = \sum_{i=1}^{q'} \Big\{ \alpha_i (L_f h_i(x) + L_g h_i(x) u(x)) \Big\} \tag{17}$$

so that by integration of (17) using the fundamental theorem of calculus and (15), we have

$$\sum_{i=1}^{q'} \Big\{ \alpha_i h_i(x(t)) \Big\} \geq \gamma \cdot (t - T') + \sum_{i=1}^{q'} \Big\{ \alpha_i h_i(x(T')) \Big\}.$$

We observe that as $t \to \infty$, $\sum_{i=1}^{q'} \Big\{ \alpha_i h_i(x(t)) \Big\} \to \infty$. But this is a contradiction since $h_i(x(t))$ for $i = \{1, 2 \ldots, q'\}$ is bounded i.e. $\sum_{i=1}^{q'} \Big\{ \alpha_i h_i(x(t)) \Big\} < \sum_{i=1}^{q'} \alpha_i M_i$. Thus, there exists a $0 < T < \infty$ such that $x(T) \in \bigcap_{i=1}^{q} \{x \in \mathbb{R}^n | h_i(x) \geq 0\}$. ■

Theorem 1 allows a system to reach an intersection of multiple regions in the state space using a single barrier certificate constraint, thus providing a larger set of feasible control laws than what would result if multiple constraints were included in the QP. We remark that [12] proposes a more restrictive solution to the constrained reachability problem with desired level sets being individually defined by multiple

[1] If all the functions are bounded, then $q' = q$ and so we will have only (15) as a constraint in the QP $\forall i \in \{1, 2, \ldots, q\}$

finite time barrier functions in a QP. In particular, [12] allows for the set of control laws $\mathcal{U}$ given by

$$\underline{\mathcal{U}}(x) = \Bigg\{ u \in \mathbb{R}^m \Bigg| L_f h_i(x) + L_g h_i(x) u(x) + \gamma \cdot sign(h_i(x)) \cdot |h_i(x)|^\rho \geq 0 \quad \forall \text{ i} \in \{1, \dots, q\} \Bigg\}, \quad (18)$$

Note that this is equivalent to taking $q' = 0$ in Theorem 1. To that end define,

$$\mathcal{U}(x) = \Bigg\{ u \in \mathbb{R}^m \Bigg| (15) \text{ and } (16) \text{ are satisfied} \Bigg\}. \quad (19)$$

Then we can formulate the following corollary

Corollary 1. *The set $\mathcal{U}(x)$ is a superset to the set $\underline{\mathcal{U}}(x)$. That is, $\underline{\mathcal{U}}(x) \subset \mathcal{U}(x)$ for all $x \in \mathcal{X}$.*

Corollary 1 provides an intuition for the main takeaway of Theorem 1. The resulting set of control laws that arise from Theorem 1 is larger than $\underline{\mathcal{U}}$ and hence, this allows for more flexibility. In Section VI, we provide a detailed analysis of the use of Theorem 1 applied to a specification very similar to the one shown in Fig 1.

B. Prioritization of Zeroing Control Barrier Functions

In this subsection, we introduce a methodology for prioritizing different zeroing control barrier functions. In particular, our proposed formulation is similar to [31] where different tasks represented by multiple zeroing barrier functions are prioritized for a multi-agent system. Our proposed method is different in the sense that, in addition to the zeroing barrier functions, we also incorporate finite time control barrier functions which are treated as hard constraints in the QP based controller.

Consider the following motivating example. Suppose we have a goal region $\mathcal{G} = \{x \in \mathcal{X} | h_\mathcal{G}(x) \geq 0\}$ where $h_\mathcal{G} : \mathcal{X} \to \mathbb{R}$ is a finite time control barrier function, encapsulated by an obstacle $\mathcal{O} = \{x \in \mathcal{X} | h_\mathcal{O}(x) \leq 0\}$ where $h_\mathcal{O} : \mathcal{X} \to \mathbb{R}$ is a zeroing control barrier function. Suppose the specification for the robot is $\phi = \Diamond \mathcal{G} \wedge \Box \neg \mathcal{O}$. By following the method proposed in existing works such as [10], [11], [3], [12], the QP that is to be solved is as follows:

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & ||u||_2^2 \\ \text{s.t} \quad & L_f h_\mathcal{G}(x) + L_g h_\mathcal{G}(x) u \geq -\gamma \cdot sign(h_\mathcal{G}(x)) \cdot |h_\mathcal{G}(x)|^\rho \\ & L_f h_\mathcal{O}(x) + L_g h_\mathcal{O}(x) u \geq -\alpha(h_\mathcal{O}(x)) \end{aligned} \quad (20)$$

where $\gamma > 0$, $\rho \in [0, 1)$ and α is a locally Lipschitz extended class κ function.

However, since the obstacle is encapsulated by the goal, the two constraints are in conflict and hence the QP will be infeasible. In order to tackle scenarios such as the one above, we propose a relaxed formulation of the QP similar to the one in [9], [31].

Consider p zeroing control barrier functions and n finite time control barrier functions. Let $\mathcal{P}$ be the index sets for the zeroing barrier functions. Some or all of the zeroing barrier functions may be in conflict with the finite time barrier function. The generalized relaxed QP is of the form,

$$\begin{aligned} \min_{v = \left[u^T, \varepsilon_1, \dots, \varepsilon_p\right]^T \in \mathbb{R}^{m+p}} \quad & ||u||_2^2 + \frac{1}{2} \Xi^T W \Xi \\ \text{s.t} \quad & (15) \text{ holds} \\ & L_f h_i(x) + L_g h_i(x) u \geq -\alpha(h_i(x)) - \varepsilon_i \text{ , } \forall i \in \mathcal{P} \end{aligned} \quad (21)$$

where $\Xi = \left[\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p\right]^T \in \mathbb{R}^p$, $W \in \mathbb{R}^{p \times p}$ is a diagonal matrix with the diagonal elements as $(w_1, w_2, \dots, w_p)$ where $w_i \in \mathbb{R}_{>0}$ is a weight associated with the the slack variable ε_i for all $i \in \{1, 2, \dots, p\}$, and α_i is a locally Lipschitz extended class κ function. The weight matrix W allows one to encode the notion of "priority" for the barrier functions. For example, if the weight w_i corresponding to the slack variable ε_i is large, then then i^{th} zeroing barrier function constraint has higher priority over other constraints.

Remark 2. *Similar to the discussion in Remark 2 in [9], if the reachability and invariance constraints are not in conflict, then with an appropriate choice of the weight matrix W, we will have $\varepsilon_i \approx 0$ for some $i \in \{1, 2, \dots, q\}$. Also, note that we extend the formulation provided in [9] from two constraints to multiple constraints.*

The relaxed QP returns a control law that allows the system to reach the desired level set in a finite time while minimally violating the invariance constraints if there is a conflict with the finite time barrier function. We present the following case study which uses the relaxation based controller (21).

1) Example: Consider a robot with single integrator dynamics

$$\dot{x} = u \quad (22)$$

where $x \in \mathcal{X} \subset \mathbb{R}^2$, and $u \in \mathbb{R}^2$. Let $\mathcal{D} \subset \mathcal{X}$ be a compact domain in the state space. The workspace is as shown in Fig 2.

Suppose we have two unsafe regions A and B and a goal region C. Let C be contained within A and B. Suppose the specification to be satisfied by the robot is $\phi = \Diamond C \wedge \Box(\neg A \wedge \neg B)$. From Fig 2, we observe that satisfaction of ϕ is impossible without entering the regions A or B. However, suppose that region A has greater priority than region B and hence violation of B is allowed to some extent.

With this additional flexibility, we can employ the proposed QP as in (21) with the weights $w_A \in \mathbb{R}_{>0}$ set to be a large value and $w_B \in \mathbb{R}_{>0}$ set to be a small value. We then solve (21) which gives us a family of trajectories (depending on the values of the weights w_A and w_B) of the robot as shown in Fig 2. Observe that with different weights w_A and w_B for the regions in the QP, we obtain a different trajectory. This allows one to also encode the notion of priority in the QP.

V. Synthesis and Analysis of Quadratic Program based Controller

In this section, we detail the theoretical framework which provides formal guarantees that the quadratic program (QP) based controller indeed produces a system trajectory that satisfies the given specification. We also describe the methodology to synthesize the barrier funtion based QP controller given an $LTL_{robotic}$ specification.

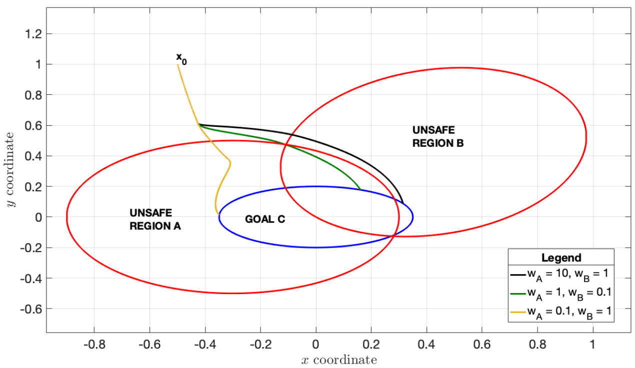

Figure 2: A family of trajectories for the robot generated by the relaxed QP (21). By changing the values of the entries in the weight matrix W, one can encode the notion of priority for different regions in the state space as can be seen from the various trajectories.

A. Lasso Type Constrained Reachability Objectives

From Section III, recall that for any $\sigma \in 2^{\Pi}$, we have $L^{-1}(\sigma) = \{x \in \mathcal{X} | \sigma = L(x)\}$. We define a *trace* as a sequence of sets of atomic propositions. The *trace of the trajectory* of a continuous time dynamical system is defined as the sequence of propositions satisfied by the trajectory. This is formalized in the definition below.

Definition 5 (Trace of a trajectory [35])**.** *An infinite sequence $\sigma = \sigma_0 \sigma_1 \dots$ where $\sigma_i \subseteq \Pi$ for all $i \in \mathbb{N}$ is the* trace of a trajectory $x(t)$ *if there exists an associated sequence $t_0 t_1 t_2 \dots$ of time instances such that $t_0 = 0$, $t_k \to \infty$ as $k \to \infty$ and for each $m \in \mathbb{N}$, $t_m \in \mathbb{R}_{\geq 0}$ satisfies the following conditions:*

- $t_m < t_{m+1}$
- $x(t_m) \in L^{-1}(\sigma_m)$
- *If $\sigma_m \neq \sigma_{m+1}$, then for some $t_m' \in [t_m, t_{m+1}]$, $x(t) \in L^{-1}(\sigma_m)$ for all $t \in (t_m, t_m')$, $x(t) \in L^{-1}(\sigma_{m+1})$ for all $t \in (t_m', t_{m+1})$, and either $x(t_m') \in L^{-1}(\sigma_m)$ or $x(t_m') \in L^{-1}(\sigma_{m+1})$.*
- *If $\sigma_m = \sigma_{m+1}$ for some m, then $\sigma_m = \sigma_{m+k}$ for all $k > 0$ and $x(t) \in L^{-1}(\sigma_m)$ for all $t \geq t_m$.* □

The last condition of the above definition implies that a trace contains a repeated set of atomic propositions only if this set holds for all future time. This is useful to capture for example, a stability condition of the system. By forbidding repetitions in other cases, we ensure that a particular trajectory possesses a unique trace. This exclusion is without loss of generality since we only consider $LTL_{robotic}$ specifications without the next operator.

It is well established that if there exists a trace that satisfies a LTL specification, then there exists a trace which satisfies the specification in *lasso* or *prefix-suffix* form [14], where a trace σ is in lasso form if it is comprised of a finite horizon prefix σ_{pre} and a finite horizon suffix σ_{suff} that is repeated infinitely often. Both σ_{pre} and σ_{suff} are finite sequences of sets of atomic propositions such that the trace σ is equal to the prefix followed by an infinite repetition of the suffix. Such a lasso-type trace is denoted as $\sigma = \sigma_{pre}(\sigma_{\text{suff}})^{\omega}$, where ω denotes infinite repetition. Atomic propositions of a continuous time dynamical system are subsets of the domain, and, hence, it is possible to interpret such lasso traces as sequences of constrained reachability problems in lasso form, which forms the basis of our control synthesis methodology. This is formalized in the following definitions.

Definition 6 (Constrained reachability objective)**.** *Given a target set $\Gamma \subset \mathcal{X}$ and a safety set $\Sigma \subset \mathcal{X}$, the* constrained reachability objective*, denoted by $R(\Sigma, \Gamma)$, is defined as the reachability problem to be solved so that the state of the system reaches the set Γ in finite time while remaining in Σ until it reaches Γ.* □

The constrained reachability objective for a system (8) is solved from a given initial condition in Σ if a control policy is found which drives the state of the system to Γ while remaining in Σ until it reaches Γ.

Definition 7 (Lasso Type Constrained Reachability Sequence)**.** *A* lasso-type constrained reachability sequence *is a sequence of constrained reachability objectives in lasso form such that each subsequent safety set is compatible with the prior goal set. That is, a lasso-type constrained reachability sequence has the form*

$$\mathcal{R}_{lasso} = \Big(R_1 R_2 \dots R_p\Big)\Big(R_{p+1}, R_{p+2} \dots R_{p+\ell}\Big)^{\omega}, \tag{23}$$

where $p > 0$, $\ell \geq 1$, and each $R_j = R(\Sigma_j, \Gamma_j)$ for some $\Gamma_j, \Sigma_j \subset \mathcal{X}$ satisfying $\Gamma_j \subseteq \Sigma_{j+1}$ for all $j \in \{1, 2, \dots, p+\ell-1\}$ and $\Gamma_{p+\ell} \subseteq \Sigma_{p+1}$. The sequence $(R_1 R_2 \dots R_p)$ is a finite horizon prefix objective and $(R_{p+1}, R_{p+2} \dots R_{p+\ell})$ is a finite horizon suffix objective that is repeated infinitely often. □

The lasso-type constrained reachability sequence is considered feasible if each constituent reachability objective is solved successfully in sequence. Note that if $p = 0$, then the finite prefix has length zero and the lasso sequence is then given by

$$\mathcal{R}_{lasso} = \Big(R_1, R_2 \dots R_\ell\Big)^{\omega}. \tag{24}$$

By the preceding discussion, if there exists a trace that satisfies a given $LTL_{robotic}$ specification, then there exists a lasso-type constrained reachability sequence which, if feasible, guarantees that the system satisfies the $LTL_{robotic}$ specification. One can view the lasso type reachability sequence as a bridge between the $LTL_{robotic}$ specification and the set based approach of our proposed controller.

B. Construction Of Lasso-type Reachability Sequence

Consider a $LTL_{robotic}$ specification ϕ as in (11). Given ϕ, our first objective is to generate the lasso-type constrained reachability sequence of the form (23).

Definition 8 (Lasso Template)**.** *Given a $LTL_{robotic}$ specification ϕ, a* lasso template *is an enumeration of the form*

$$\mathcal{O}_2 : \{1, 2, \dots, k\} \to \mathcal{I}_2 \tag{25}$$
$$\mathcal{O}_3 : \{1, 2, \dots, \ell\} \to \mathcal{I}_3 \tag{26}$$

where the index sets $\mathcal{I}_2$ and $\mathcal{I}_3$ are as per Definition 3 and $k = |\mathcal{I}_2|$ and $\ell = \max\{|\mathcal{I}_3|, 1\}$. □

Note that it is computationally straightforward to obtain *some* lasso template simply by arbitrarily enumerating the elements of the index sets $\mathcal{I}_2$ and $\mathcal{I}_3$.

A lasso-type reachability sequence of the form (23) or (24) is constructed using Algorithm 1. Define $k = |\mathcal{I}_2|$, $\ell = \max\{|\mathcal{I}_3|, 1\}$, and p is defined as,

$$p = \begin{cases} k+1 & \text{if } J_4 \neq \emptyset \\ k & \text{otherwise} \end{cases}$$

where $\mathcal{I}_2$, $\mathcal{I}_3$, and J_4 are as per Definition 3. Algorithm 1 describes the methodology to obtain a lasso sequence as in (23) or (24) from a given specification. For example, suppose we have a complex specification where all operators (ϕ_{globe}, ϕ_{reach}, ϕ_{rec}, ϕ_{act}) in Definition 3 appear in the specification ϕ. We define

$$\Gamma_i = [\![\psi_2^{\mathcal{O}_2(i)}]\!] \text{ for all } i = 1, 2, \ldots, p-1 \tag{27}$$

$$\Sigma_i = [\![\psi_1]\!] \text{ for all } i = 1, 2, \ldots, p \tag{28}$$

$$\Gamma_p = [\![\psi_4]\!] \tag{29}$$

$$\Gamma_{p+i} = [\![\psi_3^{\mathcal{O}_3(i)}]\!] \text{ for all } i = 1, 2, \ldots, \ell \tag{30}$$

$$\Sigma_i = [\![\psi_1]\!] \cap [\![\psi_4]\!] \text{ for all } i = p+1, p+2, \ldots, p+\ell. \tag{31}$$

Then we construct the lasso-type reachability sequence of the form (23) with $R_i = R_i(\Sigma_i, \Gamma_i)$ for all $i = 1, 2, \ldots, p+\ell$. Recall that the proposition sets appearing in (27)–(31) are given by

$$[\![\psi_i]\!] = \bigcap_{\forall \pi \in J_i} [\![\pi]\!] \text{ and, } [\![\psi_i^j]\!] = \bigcap_{\forall \pi \in J_i^j} [\![\pi]\!], \tag{32}$$

where ψ_i, J_i for all $i \in \{1, 4\}$, ψ_i^j, J_i^j for all $i \in \{2, 3\}$ and for all $j \in \mathcal{I}_i$ are as per Definition 3. Note that if $J_i = \emptyset$ (resp. $J_i^j = \emptyset$), then $[\![\psi_i]\!] = \emptyset$ (resp., $[\![\psi_i^j]\!] = \emptyset$) for any $i \in \{1, 4\}$ (resp., $i \in \{2, 3\}$ and $j \in \mathcal{I}_i$). The above construction is summarized in Algorithm 1 which covers all possible specifications.

C. Synthesis of Quadratic Program based Controller

We next encode the reachability objectives as finite time and zeroing control barrier functions in a QP. This is described in Algorithm 2. Each Γ_i is encoded with finite time control barrier function(s) with (5) or (19) as constraint(s) whereas each Σ_i is encoded with zeroing control barrier function(s) with (3) as constraint(s) in the QP. The designer is free to choose a locally Lipschitz α function for (2). In order to solve a particular reachability objective $R_i(\Sigma_i, \Gamma_i)$ where $i \in \{1, 2, \ldots, n\}$, we solve a QP as in (7). Note that solving a QP in real time is typically done in a few milliseconds, and hence Algorithm 2 is amenable to real time implementation on robotic platforms.

D. Analysis Of Trajectory Generated by QP Controller

Observe that there is a one-to-one correspondence between elements of $P(\Pi_{aug})$ and subsets of Π. Let $\iota : 2^\Pi \to P(\Pi_{aug}) \subset 2^{\Pi_{aug}}$ be the canonical bijective mapping for a subset $\sigma \in 2^\Pi$ with the corresponding mapping $\iota(\sigma) \in P(\Pi_{aug})$ given by,

$$\pi \in \sigma \iff \pi \in \iota(\sigma) \text{ and } \pi \notin \sigma \iff \overline{\pi} \in \iota(\sigma). \tag{33}$$

Algorithm 1 Lasso-type Reachability Sequence Generator

Input : ϕ, $\mathcal{O}_2$, $\mathcal{O}_3$
Output: $\mathcal{R}_{lasso}$
if $J_4 \neq \emptyset$ **then**
  $p \leftarrow k+1$
  if $k \neq 0$ **then**
    $\Gamma_i = [\![\psi_2^{\mathcal{O}_2(i)}]\!]$ for all $i = 1, 2, \ldots, p-1$
    $\Sigma_i = [\![\psi_1]\!]$ for all $i = 1, 2, \ldots, p-1$
  end if
  $\Gamma_p = [\![\psi_4]\!]$
  $\Sigma_p = [\![\psi_1]\!]$
  $\Gamma_{p+i} = [\![\psi_3^{\mathcal{O}_3(i)}]\!]$ for all $i = 1, 2, \ldots, \ell$
  if $J_1 = \emptyset$ **then**
    $\Sigma_{p+i} = [\![\psi_4]\!]$ for all $i = 1, 2, \ldots, \ell$
  else
    $\Sigma_{p+i} = [\![\psi_1]\!] \cap [\![\psi_4]\!]$ for all $i = 1, 2, \ldots, \ell$
  end if
  $R_i = R_i(\Sigma_i, \Gamma_i)$ for all $i = 1, 2, \ldots, p+\ell$
  Return $\mathcal{R}_{lasso}$ as in (23)
else
  $p \leftarrow k$
  if $p \neq 0$ **then**
    $\Gamma_i = [\![\psi_2^{\mathcal{O}_2(i)}]\!]$ for all $i = 1, 2, \ldots, p$
    $\Gamma_{p+i} = [\![\psi_3^{\mathcal{O}_3(i)}]\!]$ for all $i = 1, 2, \ldots, \ell$
    $\Sigma_i = [\![\psi_1]\!]$ for all $i = 1, 2, \ldots, p+\ell$
    $R_i = R_i(\Sigma_i, \Gamma_i)$ for all $i = 1, 2, \ldots, p+\ell$
    Return $\mathcal{R}_{lasso}$ as in (23)
  else
    $\Gamma_{p+i} = [\![\psi_3^{\mathcal{O}_3(i)}]\!]$ for all $i = 1, 2, \ldots, \ell$
    $\Sigma_{p+i} = [\![\psi_1]\!]$ for all $i = 1, 2, \ldots, \ell$
    $R_i = R_i(\Sigma_i, \Gamma_i)$ for all $i = 1, 2, \ldots, p+\ell$
    Return $\mathcal{R}_{lasso}$ as in (24)
  end if
end if

Algorithm 2 Quadratic Program based Controller

Input : $\mathcal{R}_{lasso}$
if $p \neq 0$ **then**
  for $i = 1, 2, \ldots, p$ **do**
    Encode Γ_i with FCBFs
    Encode Σ_i with ZCBFs
    while $x \notin \Gamma_i$ **do**
      Solve $R(\Sigma_i, \Gamma_i)$ as in (7)
    end while
  end for
end if
while true **do**
  for $i = p+1, \ldots, p+\ell$ **do**
    Encode Γ_i with FCBFs
    Encode Σ_i with ZCBFs
    while $x \notin \Gamma_i$ **do**
      Solve $R(\Sigma_i, \Gamma_i)$ as in (7)
    end while
  end for
end while

For notational convenience, we do not explicitly differentiate between a subset $\sigma \subset \Pi$ and its mapping $\iota(\sigma) \in P(\Pi_{aug})$.

Given Algorithm 2, we now provide formal guarantees which prove that the QP from Algorithm 2 indeed produces a system trajectory which satisfies the system specification.

Definition 9 (Descendant). *Given a $LTL_{robotic}$ specification ϕ with a lasso template $\mathcal{O}_2$ and $\mathcal{O}_3$ as in Definition 8, a* descendant *of the lasso template is any infinite length sequence of the form*

$$\sigma = \left\{\sigma_{1,1}\sigma_{1,2}\dots\sigma_{1,n_1}\right\}\left\{\sigma_{2,1}\sigma_{2,2}\dots\sigma_{2,n_2}\right\}\dots \left\{\sigma_{p,1}\sigma_{p,2}\dots\sigma_{p,n_p}\right\}\dots, \quad (34)$$

where $\sigma_{i,j} \in P(\Pi_{aug})$ for all $i = 1,2,\dots,\ j = 1,2,\dots,n_i$ and

1) $J_1 \subseteq \sigma_{i,j}$ *for all* $i \in \{1,2,\dots,p\}$ *and for all* $j \in \{1,2,\dots,n_i\}$
2) $J_2^{\mathcal{O}_2(i)} \subseteq \sigma_{\mathcal{O}_2(i),n_{\mathcal{O}_2(i)}}$ *for all* $i \in \{1,2,\dots,k\}$
3) $J_4 \subseteq \sigma_{p,n_p}$
4) $J_3^{\mathcal{O}_3(i)} \subseteq \sigma_{m,n_m}$ *where* $m = p + d\ell + \mathcal{O}_3(i)$ *for all* $d \in \{0,1,2\dots\}$ *and for all* $i \in \{1,2,\dots,\ell\}$
5) $J_1 \cup J_4 \subseteq \sigma_{i,j}$ *for all* $i \in \{p+1,\dots\}$ *and for all* $j \in \{1,2,\dots,n_i\}$. □

Intuitively, a descendant σ of a given template is a sequence of atomic propositions visited by the system such that the it respects the safety sets Σ_i and also reaches the target sets Γ_i in a finite time for all $i \in \{1,2,\dots,p+l\}$. In (34), each set $\sigma_i = \left\{\sigma_{i,1}\sigma_{i,2}\dots\sigma_{i,n_i}\right\}$ corresponds to the i^{th} constrained reachability objective in the lasso sequence (23) or (24) and the set $\sigma_p = \left\{\sigma_{p,1}\sigma_{p,2}\dots\sigma_{p,n_p}\right\}$ is the last constrained reachability objective in the finite prefix part of the lasso sequence after which the sequence switches to the suffix.

Proposition 3. *Given a lasso template as in Definition 8 for a $LTL_{robotic}$ specification ϕ as in* (11)*, any descendant σ of this template is such that $\sigma \models \phi$.*

Proof. Let $\phi = \phi_{globe} \wedge \phi_{reach} \wedge \phi_{rec} \wedge \phi_{act}$ be a specification as in (11). Let $\mathcal{O}_2$ and $\mathcal{O}_3$ be a lasso template for the specification. Let σ be a descendant of the lasso template as in Definition 9.

We provide a proof by construction by considering four individual cases for the specification ϕ. Then, since conjunction preserves the results from these cases [14], we combine them to provide a proof for the entire fragment of LTL.

Case 1: Suppose $\phi = \phi_{globe} = \square\psi_1$ for $\psi_1 = \bigwedge_{m=1}^{n} \pi_m$, where $\pi_m \in \Pi_{aug}$. Thus we have $J_1 = \{\pi_1,\dots,\pi_n\}$, $J_2^{\mathcal{O}_2(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,k\}$, $J_3^{\mathcal{O}_3(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,\ell\}$ and $J_4 = \{\emptyset\}$. A descendant trace of the template is as per Definition 9. Thus, from condition 1 in Definition 9 , we observe that $J_1 = \{\pi_1,\dots,\pi_n\} \subseteq \sigma_{i,j}$ for all $i \in \{1,2,\dots\}$ and for all $j \in \{1,2,\dots,n_i\}$. Hence, we can conclude that $\sigma \models \phi_{globe}$.

Case 2: Suppose $\phi = \phi_{act} = \lozenge\square\psi_4$ for $\psi_4 = \bigwedge_{m=1}^{n} \pi_m$ where $\pi_m \in \Pi_{aug}$. Thus we have $J_1 = \{\emptyset\}$, $J_2^{\mathcal{O}_2(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,k\}$, $J_3^{\mathcal{O}_3(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,\ell\}$ and $J_4 = \{\pi_1,\dots,\pi_n\}$. A descendant trace of the template has a closed form expression as in Definition 9. Thus, from condition 3 in Definition 9, we have $J_4 \subseteq \sigma_{p,n_p}$, and from condition 5 in Definition 9, we observe that $J_4 = \{\pi_1,\dots,\pi_n\} \subseteq \sigma_{i,j}$ for all $i \in \{p+1,p+2,\dots\}$ and for all $j \in \{1,2,\dots,n_i\}$. Hence, we can conclude that $\sigma \models \phi_{act}$.

Case 3: Suppose $\phi = \phi_{reach} = \bigwedge_{j\in\mathcal{I}_2} \lozenge\psi_2^j$. Thus we have $J_1 = \{\emptyset\}$, $J_3^{\mathcal{O}_3(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,\ell\}$ and $J_4 = \{\emptyset\}$. A descendant trace of the template has a closed form expression as in Definition 9. Thus, from condition 2 in the definition, we observe that $J_2^{\mathcal{O}_2(m)} = \{\pi_1,\dots,\pi_n\} \subseteq \sigma_{\mathcal{O}_2(m),n_{\mathcal{O}_2(m)}}$ for all $m \in \{1,2,\dots,k\}$. Hence, we can conclude that $\sigma \models \phi_{reach}$.

Case 4: Suppose $\phi = \phi_{rec} = \bigwedge_{j\in\mathcal{I}_3} \square\lozenge\psi_3^j$. Thus we have $J_1 = \{\emptyset\}$, $J_2^{\mathcal{O}_2(i)} = \{\emptyset\}$ for all $i \in \{1,2,\dots,k\}$ and $J_4 = \{\emptyset\}$. A descendant trace of the template has a closed form expression as in Definition 9. Thus, from condition 4 in the definition, we observe that $J_3^{\mathcal{O}_3(q)} = \{\pi_1,\dots,\pi_n\} \subseteq \sigma_{m,n_m}$ for all $m = p + dl + \mathcal{O}_3(q)$, for all $d \in \{0,1,2\dots\}$ and for all $q \in \{1,2,\dots,\ell\}$. Hence, we can conclude that $\sigma \models \phi_{rec}$.

Thus, by combining the results from Cases 1, 2, 3 and 4 with conjunction [14], we can conclude that σ satisfies $\phi = \phi_{globe} \wedge \phi_{reach} \wedge \phi_{rec} \wedge \phi_{act}$. That is, $\sigma \models \phi$. ■

Next we state Theorem 2 which provides a theoretical guarantee that if Algorithm 2 is feasible, then the trace of the resulting system trajectory satisfies the specification.

Theorem 2. *Given a $LTL_{robotic}$ specification ϕ and a lasso template $\mathcal{O}_2$ and $\mathcal{O}_3$ as in Definition 8, let $\mathcal{R}_{lasso}$ be the lasso-type constrained reachability sequence as in* (23) *generated from Algorithm 1. If Algorithm 2 is feasible, then the trace of the system trajectory $x(t)$ satisfies ϕ.*

Proof. As per Algorithm 2, each Σ_i is encoded as constraint(s) with zeroing control barrier function(s) for all $i \in \{1,2,\dots,p+\ell\}$. From Proposition 1, this guarantees forward invariance of the atomic propositions that need to remain true or need to remain false. Since the QP from Algorithm 2 is feasible, conditions 1 and 5 from Definition 9 are satisfied. Since each Γ_i is encoded as constraint(s) with finite time control barrier function(s) for all $i \in \{1,2,\dots,p+\ell\}$, from Theorem 1 we can guarantee finite time convergence to atomic propositions that need to be reached in finite time. This satisfies conditions 2, 3 and 4 of Definition 9. Thus, all conditions in Definition 9 are satisfied. Since the QP is feasible, we conclude that Algorithm 2 generates a descendant σ of the lasso template.

From Proposition 3, we know that given a lasso template, any descendant σ of the lasso template is such that it satisfies the specification. From the previous analysis, we know that the QP from Algorithm 2 produces a descendant of the lasso template. The mapping ι being bijective and combining Proposition 3 with the previous analysis, we can conclude that QP from Algorithm 2 produces a trace of the trajectory of the system that satisfies the given specification. That is, $[\![\iota^{-1}(\sigma)]\!] = [\![\sigma]\!] \models \phi$. ■

Note that while Algorithm 2 and Theorem 2 assume that the QP (7) is feasible, one can always use the relaxed QP (21) for feasibility. In that case, although feasibility of the controller is more likely, Theorem 2 may no longer hold since the relaxation parameters ε can be non-zero so that the corresponding atomic propositions are no longer satisfied. However, such a situation is not considered in this paper.

VI. Case Study

In this section, we provide a case study that details the barrier functions based QP framework which synthesizes a system trajectory that satisfies the specification. This case study was implemented in the Robotarium multi-robot testbed at Georgia Tech [36]. The Robotarium consists of differential drive mobile robots which can be programmed using either MATLAB or Python.

The unicycle kinematics which describe the differential drive robots is given by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos(\phi) & 0\\ \sin(\phi) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix},$$

where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ represent the position of the robot, $\phi \in (-\pi, \pi]$ represents it's orientation, $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ are the linear and angular velocity inputs to the robot respectively. Before we proceed with the case study, we point out that we treat the robots as single integrators and then using the Near Identity Diffeomorphism (NID) technique presented in [37], we can map the single integrator velocities to unicycle velocities. The unicycle kinematics is a more accurate model for the differential drive robot.

In particular, the near identity mapping is given by

$$\begin{bmatrix}v\\ \omega\end{bmatrix} = R^T(\phi)\begin{bmatrix}1 & 0\\ 0 & \frac{1}{L}\end{bmatrix}\begin{bmatrix}\dot{x_L}\\ \dot{y_L}\end{bmatrix}$$

where $R(\phi)$ is the rotation matrix, $\dot{x_L}$ and $\dot{y_L}$ is the velocity of a point located at a distance L ahead of the centre of mass (CoM) of the robot. The method of treating the unicycle as a single integrator is a common approach in existing literature and is also the method used in the Robotarium™ test bed [36].

Consider a team of three robots: one surveillance robot ($\mathtt{R}_3$) and two attack robots ($\mathtt{R}_1$ and $\mathtt{R}_2$). The surveillance robot needs to collect information regarding the position of two targets, and then return back to the base. Once the information has been relayed to the base by the surveillance robot, the attack robots must visit the targets infinitely often. In addition to this, the attack robots must stay connected with each other at all times, and all the robots must avoid a danger zone where they can be attacked.

Let $\mathcal{D} \subset \mathbb{R}^2$ be the workspace for each robot and let $\mathcal{D} \times \mathcal{D} \times \mathcal{D} \subset \mathbb{R}^6$ be the domain of the three robot system with regions $\mathcal{A} = \{A, B, C, O\}$. The dynamics for each agent $i \in \{1,2,3\}$ is

$$\dot{x}_i = u_i \tag{35}$$

where $x_i \in \mathcal{D}$ and $u_i \in \mathbb{R}^2$.

Target 1 is labelled as A, target 2 is labelled as B the base is labelled as C, and O is the danger zone (obstacle). The set of atomic propositions is given by $\Pi = \{\pi_i^r, \overline{\pi}_i^r\} \cup \{\pi^{conn}, \overline{\pi}^{conn}\}$ for all $i \in \{1,2,3\}$ and $r \in \{A, B, C, O\}$. The regions A, B, C are defined as $[\![\pi_i^r]\!] = \{x \in \mathcal{D}^3 | h_r(x_i) \geq 0\}$ for all $r \in \{A,B,C,O\}$ and for all $i \in \{1,2,3\}$. For each $[\![\pi_i^r]\!]$ with $i \in \{1,2,3\}$, $r \in \{A,B,C,O\}$, let

$$\pi_i^r = \begin{cases} 1 & \text{if } x_i \in r\\ 0 & \text{otherwise.}\end{cases} \tag{36}$$

This means $\pi_i^r = 1$ if and only if agent i is in region r. The additional connectivity constraint that must be maintained by $\mathtt{R}_1$ and $\mathtt{R}_2$ is given as $h_{conn}(x) \geq 0$ where

$$h_{conn}(x) = d_{conn}^2(x) - ||x_2 - x_1||^2, \tag{37}$$

where $d_{conn} : \mathcal{D} \times \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ is the connectivity distance between the two agents that needs to be maintained, and $||x_2 - x_1||$ is the inter-agent distance. We consider

$$d_{conn}^2(x) = (x_{2,1} + \delta_1)^2 + \delta_2, \tag{38}$$

where δ_1 and δ_2 are constants, and $x_{2,1}$ is the x coordinate of $\mathtt{R}_2$ in the workspace. The connectivity set corresponding to the proposition π^{conn} is defined as $[\![\pi^{conn}]\!] = \{x \in \mathcal{D}^3 | h_{conn}(x) \geq 0\}$. Such a constraint captures a situation in which the robots have poor connectivity in certain areas of the workspace, which requires them to maintain a closer distance with each other. In areas where the robots have strong connectivity, they are free to maintain a larger distance from each other.

The $LTL_{robotic}$ specification for the task described previously is

$$\begin{aligned}\phi = (\Diamond\pi_3^A \wedge \Diamond\pi_3^B \wedge \Diamond\pi_3^C) \wedge \Box\Diamond(\pi_1^A \wedge \pi_2^B) \wedge \Box\Diamond(\pi_1^C \wedge \pi_2^C)\\ \wedge \Box(\pi^{conn} \wedge \neg\pi_1^O \wedge \neg\pi_2^O \wedge \neg\pi_3^O).\end{aligned} \tag{39}$$

From the formalism in Definition 7 and Algorithm 1, we obtain the lasso-type constrained reachability objective,

$$\mathcal{R}_{lasso} = \Big(R_1(\Sigma_1, \Gamma_1) R_2(\Sigma_2, \Gamma_2) R_3(\Sigma_3, \Gamma_3)\Big) \tag{40}$$

$$\Big(R_4(\Sigma_4, \Gamma_4) R_5(\Sigma_5, \Gamma_5)\Big)^{\omega} \tag{41}$$

where $\Sigma_i = [\![\pi^{conn}]\!] \cap [\![\overline{\pi_1}^O]\!] \cap [\![\overline{\pi_2}^O]\!] \cap [\![\overline{\pi_3}^O]\!]$ for $i = 1,2,3,4,5$, $\Gamma_1 = [\![\pi_3^A]\!]$, $\Gamma_2 = [\![\pi_3^B]\!]$, $\Gamma_3 = [\![\pi_3^C]\!]$, $\Gamma_4 = [\![\pi_1^A]\!] \cap [\![\pi_2^B]\!]$, $\Gamma_5 = [\![\pi_1^C]\!] \cap [\![\pi_2^C]\!]$.

Next, we use Algorithm 2 to generate the pointwise controller for the system.

The first reachability objective is encoded in the QP as

$$\begin{aligned}\frac{\partial(h_A(x_3))}{\partial x}u &\geq -\gamma \cdot \mathrm{sign}(h_A(x_3)) \cdot |h_A(x_3)|^{\rho}\\ \frac{\partial h_O(x_i)}{\partial x}u &\geq -\gamma \cdot h_O(x_i)^3, \text{ for all } i \in \{1,2,3\}\\ \frac{\partial h_{conn}(x)}{\partial x}u &\geq -\gamma \cdot h_{conn}(x)^3.\end{aligned}$$

The second reachability objective is encoded in the QP as

$$\begin{aligned}\frac{\partial(h_B(x_3))}{\partial x}u &\geq -\gamma \cdot \mathrm{sign}(h_B(x_3)) \cdot |h_B(x_3)|^{\rho}\\ \frac{\partial h_O(x_i)}{\partial x}u &\geq -\gamma \cdot h_O(x_i)^3, \text{ for all } i \in \{1,2,3\}\\ \frac{\partial h_{conn}(x)}{\partial x}u &\geq -\gamma \cdot h_{conn}(x)^3.\end{aligned}$$

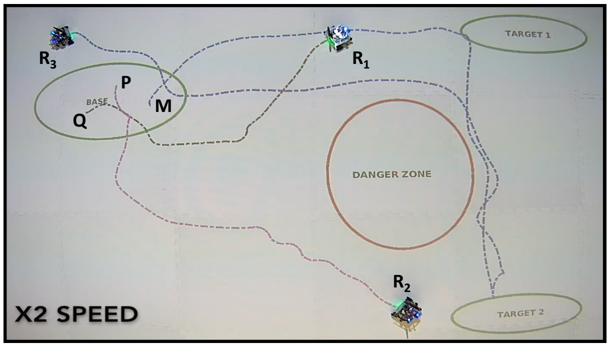

Figure 3: A still shot of the trajectories for the robots $\mathtt{R}_1$, $\mathtt{R}_2$ and $\mathtt{R}_3$ for the specification ϕ as in (39). Observe that $\mathtt{R}_1$ moves temporarily away from target 1 temporarily in order to satisfy the connectivity constraint, but Theorem 1 results in feasible solutions at those points. From the figure, we observe that the robots maintain connectivity and avoid the danger zone at all times.

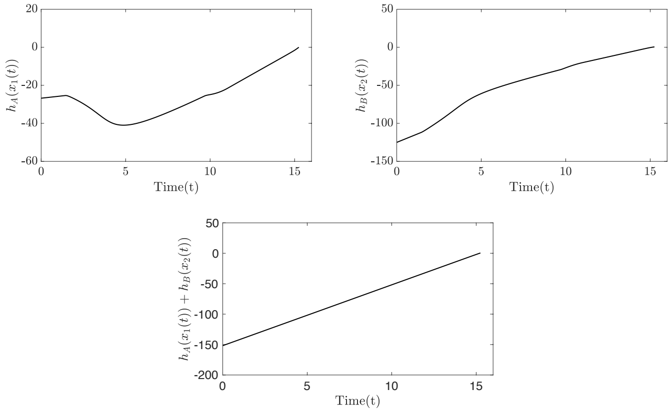

Figure 4: Level sets for goal A and goal B. Even though $\mathtt{R}_1$ moves away from goal A for a brief moment, the net total progress towards the goals is increasing at all times thus guaranteeing feasibility within the connectivity corridor.

The third reachability objective is encoded in the QP as

$$\frac{\partial(h_C(x_3))}{\partial x}u \geq -\gamma\cdot \text{sign}(h_C(x_3))\cdot|h_C(x_3)|^\rho$$
$$\frac{\partial h_O(x_i)}{\partial x}u \geq -\gamma\cdot h_O(x_i)^3, \text{ for all } i\in\{1,2,3\}$$
$$\frac{\partial h_{conn}(x)}{\partial x}u \geq -\gamma\cdot h_{conn}(x)^3.$$

The fourth reachability objective is encoded in the QP as

$$\frac{\partial(h_A(x_1)+h_B(x_2))}{\partial x}u \geq -\gamma\cdot \text{sign}(\min\{h_A(x_1),h_B(x_2)\})$$
$$\frac{\partial h_O(x_i)}{\partial x}u \geq -\gamma\cdot h_O(x_i)^3, \text{ for all } i\in\{1,2,3\}$$
$$\frac{\partial h_{conn}(x)}{\partial x}u \geq -\gamma\cdot h_{conn}(x)^3.$$

The fifth reachability objective is encoded in the QP as

$$\frac{\partial(h_C(x_1)+h_C(x_2))}{\partial x}u \geq -\gamma\cdot \text{sign}(\min\{h_C(x_1),h_C(x_2)\})$$
$$\frac{\partial h_O(x_i)}{\partial x}u \geq -\gamma\cdot h_O(x_i)^3, \text{ for all } i\in\{1,2,3\}$$
$$\frac{\partial h_{conn}(x)}{\partial x}u \geq -\gamma\cdot h_{conn}(x)^3.$$

If $\mathcal{U}_i(x)$ is the set of feasible control laws that satisfies all the constraints for each reachability objective, then for all $i=\{1,2,3,4,5\}$ the QP that is solved is,

$$\begin{aligned} \min_{u\in\mathbb{R}^6} \quad & ||u||_2^2 \\ \text{s.t} \quad & u\in\mathcal{U}_i(x). \end{aligned} \tag{42}$$

From Theorem 2 we conclude that these trajectories indeed satisfy the specification ϕ. The switching between the current reachability objective to the next is automatic. It occurs when the state of the system reaches the desired set of states. That is, the switching from reachability objective i to objective $i+1$ occurs when $x\in\Gamma_i$ for all $i\in\{1,2,3,4\}$.

In Fig 3, Q, P and M are the initial conditions for robots $\mathtt{R}_1$, $\mathtt{R}_2$ and $\mathtt{R}_3$ respectively. In sections of the trajectory for R_1, we see that it moves away from target 1. However, because of Theorem 1, the QP is feasible along this portion of the trajectory. This can be seen more clearly in Fig 4. Observe that even though R_1 moves temporarily away from A, the net progress towards the targets is increasing and hence the QP is returns a feasible solution solution. At all times, $\mathtt{R}_1$ and $\mathtt{R}_2$ stay connected as per the distance dictated by (37) and avoid the danger zone, as seen in Fig 3. Thus, we see that by solving this sequence of constrained reachability objectives, the multi-agent system satisfies the specification. Fig 3 is a still shot of the experiment conducted on the Robotarium™ multi-robot testbed at Georgia Tech [36] [2] [3].

VII. Concluding Remarks

In this paper, we provided a framework for the control of mobile robotic systems with control affine dynamics. In particular, we used control barrier functions and temporal logic as the tools to develop this framework. First, we discussed issues regarding feasibility of the QP based controller. We provided a new method to compose multiple finite time barrier functions in order to obtain a larger feasible solution set as compared to existing methods in literature. We also proposed a modified QP based controller which prioritizes different zeroing control barrier functions. Second, we developed a fully automated framework which synthesizes a barrier function based controller given a specification. Last, we provided formal guarantees that the QP based controller generates a system trajectory that satisfies the given specification.

References

[1] S. Chinchali, S. C. Livingston, U. Topcu, J. W. Burdick, and R. M. Murray, "Towards formal synthesis of reactive controllers for dexterous robotic manipulation," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 5183–5189.

[2] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1470–1477.

[2] Video of Robotarium experiment -(https://youtu.be/EK1Zxcg-eSE)

[3] MATLAB simulation code- https://github.com/gtfactslab/TRO_CBFs-LTL_Robotarium_Experiment.git

[3] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3293–3298.

[4] "Trustworthy cyber infrastructure for the power grid (tcip-g)," Mar 2019. [Online]. Available: https://www.dhs.gov/science-and-technology/trustworthy-cyber-infrastructure-power-grid-tcip-g

[5] M. Simon, "Your first look inside amazon's robot warehouse of tomorrow," Jun 2019. [Online]. Available: https://www.wired.com/story/amazon-warehouse-robots/

[6] [Online]. Available: https://www.dhs.gov/cisa/critical-infrastructure-sectors

[7] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM review*, vol. 44, no. 4, pp. 525–597, 2002.

[8] 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*, Dec 2014, pp. 6271–6278.

[9] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *arXiv preprint arXiv:1612.01554*, 2016.

[10] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, June 2017.

[11] L. Wang, A. Ames, and M. Egerstedt, "Safety barrier certificates for heterogeneous multi-robot systems," in *2016 American Control Conference (ACC)*, July 2016, pp. 5213–5218.

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

[13] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 1991–1996.

[14] C. Baier and J.-P. Katoen, *Principles of model checking*, 2008.

[15] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 971–984, July 2000.

[16] C. Belta and L. C. G. J. M. Habets, "Controlling a class of nonlinear systems on rectangles," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1749–1759, Nov 2006.

[17] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, Feb 2008.

[18] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2817–2830, Nov 2012.

[19] A. Bhatia, M. R. Maly, L. E. Kavraki, and M. Y. Vardi, "Motion planning with complex goals," *IEEE Robotics Automation Magazine*, vol. 18, no. 3, pp. 55–64, Sep. 2011.

[20] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[21] 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.

[22] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 81–87.

[23] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, 2019.

[24] Z. Liu, B. Wu, J. Dai, and H. Lin, "Distributed communication-aware motion planning for multi-agent systems from stl and spatel specifications," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 4452–4457.

[25] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. E. Z. Soudjani, "Shrinking horizon model predictive control with chance-constrained signal temporal logic specifications," in *2017 American Control Conference (ACC)*, May 2017, pp. 1740–1746.

[26] D. Aksaray, A. Jones, Z. Kong, M. Schwager, and C. Belta, "Q-learning for robust satisfaction of signal temporal logic specifications," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 6565–6570.

[27] D. Muniraj, K. G. Vamvoudakis, and M. Farhood, "Enforcing signal temporal logic specifications in multi-agent adversarial environments: A deep q-learning approach," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4141–4146.

[28] P. Várnai and D. V. Dimarogonas, "Prescribed performance control guided policy improvement for satisfying signal temporal logic tasks," *CoRR*, vol. abs/1903.04340, 2019. [Online]. Available: http://arxiv.org/abs/1903.04340

[29] L. Lindemann and D. V. Dimarogonas, "Decentralized robust control of coupled multi-agent systems under local signal temporal logic tasks," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 1567–1573.

[30] Y. V. Pant, H. Abbas, R. A. Quaye, and R. Mangharam, "Fly-by-logic: Control of multi-drone fleets with temporal logic objectives," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, April 2018, pp. 186–197.

[31] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An optimal task allocation strategy for heterogeneous multi-robot systems," *arXiv preprint arXiv:1903.08641*, 2019.

[32] H. K. Khalil, *Nonlinear systems*, vol. 3.

[33] E. M. Wolff, U. Topcu, and R. M. Murray, "Efficient reactive controller synthesis for a fragment of linear temporal logic," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5033–5040.

[34] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug 2017.

[35] T. Wongpiromsarn, U. Topcu, and A. Lamperski, "Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3344–3355, 2016.

[36] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1699–1706.

[37] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential ϵ-tracking and ϵ-stabilization of first-order nonholonomic SE(2) vehicles," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 6, May 2002, pp. 4690–4695 vol.6.

Mohit Srinivasan is currently pursuing his PhD degree in Electrical and Computer Engineering (ECE) from Georgia Institute of Technology, Atlanta, Georgia, USA. He earned his Master of Science in Electrical and Computer Engineering from Georgia Tech and Bachelor of Technology in Electrial Engineering from Veermata Jijabai Technological Institute (V.J.T.I), Mumbai, India. He was previously a research intern at Landis+Gyr, Atlanta. His research interests are primarily in multi-agent systems, control theory, and robotics.

Samuel Coogan is an Assistant Professor at Georgia Tech in the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering. Prior to joining Georgia Tech in July 2017, he was an assistant professor in the Electrical Engineering Department at UCLA from 2015-2017. He received the B.S. degree in Electrical Engineering from Georgia Tech and the M.S. and Ph.D. degrees in Electrical Engineering from the University of California, Berkeley. His research is in the area of dynamical systems and autonomy and focuses on developing scalable tools for verification and control of networked, cyber-physical systems with an emphasis on transportation systems. He received a CAREER award from NSF in 2018, a Young Investigator Award from the Air Force Office of Scientific Research in 2018, and the Outstanding Paper Award for the IEEE Transactions on Control of Network Systems in 2017.