2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Control Barrier Functions for Abstraction-Free Control Synthesis under Temporal Logic Constraints

Luyao Niu and Andrew Clark

Abstract— **Temporal logic has been widely used to express complex task specifications for cyber-physical systems (CPSs). One way to synthesize a controller for CPS under temporal logic constraints is to first abstract the CPS as a discrete transition system, and then apply formal methods. This approach, however, is computationally demanding and its scalability suffers due to the curse of dimensionality. In this paper, we propose a control barrier function (CBF) approach to abstraction-free control synthesis under a linear temporal logic (LTL) constraint. We first construct the deterministic Rabin automaton of the specification and compute an accepting run. We then compute a sequence of LTL formulae, each of which must be satisfied during a particular time interval, and prove that satisfying the sequence of formulae is sufficient to satisfy the LTL specification. Finally, we compute a control policy for satisfying each formula by constructing an appropriate CBF. We present a quadratic program to compute the controllers, and show the controllers synthesized using the proposed approach guarantees the system to satisfy the LTL specification, provided the quadratic program is feasible at each time step. A numerical case study is presented to demonstrate the proposed approach.**

I. Introduction

Cyber-physical systems (CPSs) are assigned increasingly complex objectives including reactive and sequential tasks in domains such as autonomous vehicles, advanced manufacturing, and health care systems. Temporal logics [1] such as linear temporal logic (LTL) are widely adopted to specify properties and verify behaviors of CPSs due to their rich and rigorous expressiveness. Consequently, temporal logic based control has gained research attention in areas including robotics [2] and traffic network control [3].

While control synthesis for CPSs focuses on generating closed-loop controllers in the continuous domain, the semantics of LTL are defined in a discrete manner, which are normally captured by finite state automata. Motivated by formal methods in the context of model checking [1], control synthesis for CPSs are normally lifted from continuous domain to discrete domain by computing a finite abstraction of the CPS. The finite abstraction can be generated by partitioning the state space and control input space of CPSs with consistency guarantees (e.g., simulation and bisimulation relations [4]). Typical forms of finite abstractions of CPSs include finite transition system [5], Markov decision process [6], [7], and stochastic game [8].

Abstraction-based approaches are computationally demanding and suffer from the curse of dimensionality. To avoid the computation required for constructing the finite abstraction, researchers have investigated control synthesis on the continuous state space. Techniques include formulating the LTL constraint as a mixed integer linear program [9], a sequence of stochastic reachability problems [10], and a mixed continuous-discrete HJB equation [11]. Solving for the controller thus requires numerical methods which can be computationally expensive.

In this paper, we consider control synthesis for CPSs under LTL constraints without computing a finite abstraction. We aim to compute a feedback controller such that a control affine system satisfies a given LTL specification from the fragment of LTL without next operator. We present a control barrier function (CBF) based framework to compute the controller. There are two main advantages of our approach compared to the state of the art. First, we avoid both finite-state abstraction of the CPS and approximate solution of the HJB equation, and thus reduce the computational complexity. Second, we introduce time-varying *guard functions* that render our approach feasible for a broad class of LTL properties. To summarize, this paper makes the following contributions.

- We present a CBF-based approach to synthesize a controller for CPSs under LTL constraints. The proposed approach is provably correct and avoids explicit construction of a finite abstraction of CPSs.
- We construct a sequence of formulae that correspond to an accepting trace of the CPS, and develop a methodology to construct CBFs for each formula. When designing the CBF, we introduce the concept of guard function, which implicitly encodes the time that each formula needs to be satisfied and enhances the feasibility of the CBF constraints.
- We compute the set of controllers that satisfy each constructed formula using two types of time varying CBFs. We show that by satisfying the CBF constraints for all formulae, the LTL specification is satisfied.
- A numerical case study on robotic motion planning is presented as evaluation. The proposed approach successfully synthesizes a controller for a specification that is infeasible under the state of art.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III gives preliminary background on CBFs and LTL. The problem of interest is formulated in Section IV. Our solution approach is presented in Section V. Section VI gives a numerical case study on

L. Niu and A. Clark are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA. {lniu,aclark}@wpi.edu

This work was supported by the National Science Foundation and the Office of Naval Research via grants CNS-1941670 and N00014-17-S-B001.

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

robotic motion planning. Section VII concludes the paper.

II. Related Work

Verification and control synthesis methodologies for CPS under temporal logic constraints have been proposed based on constructing an abstraction of the dynamical system [2], [5]–[8]. However, it is normally computationally expensive to construct an abstraction of the dynamical system.

To avoid constructing the finite abstraction, controller synthesis in continuous domain under LTL constraints has been studied. An optimization based approach was proposed in [9]. The authors encoded the LTL constraints as mixed-integer linear constraints. The authors of [10] studied control synthesis under co-safe LTL specifications. They computed the controller by solving a sequence of stochastic constrained reachability problems, which relied on solving nonlinear PDEs. The authors of [11] formulated a hybrid system using the automaton and the continuous system, and computed the controller by solving a mixed continuous-discrete HJB equation. Normally PDEs and HJB equations are approximately solved using computationally expensive numerical approaches such as Finite-Difference [12]. In this work, we give a different CBF based approach to synthesize a controller by solving a quadratic program at each time.

The concept of CBF extends barrier function to CPSs with control inputs [13]. Recent works have developed CBF based approaches for safety critical system [14], [15]. LTL specifications, however, capture a richer set of properties that cannot be modeled by safety constraints alone.

CBFs have gained popularity for CPSs under LTL constraints [16], [17] and signal temporal logic (STL) constraints [18], [19]. The authors of [16], [17] studied the problem of control synthesis of multi-robot system under LTL constraints using CBFs. Under their approaches, they solved a sequence of reachability problems using CBFs, and proposed a relaxation of CBF based constraints. This work differs from [16], [17] in the following two aspects. First, our approach considers a broader fragment of LTL compared to [16]. A fragment of LTL specification named LTL$_{Robotic}$ is considered in [16], while we consider LTL without next operator. Second, we propose a different approach to resolve the infeasibility between CBF constraints compared to [16], [17]. As we will demonstrate later, our proposed approach synthesizes a controller for a specification that is infeasible using the approaches in [16], [17], and thus serves as a complement to [16], [17].

III. Preliminaries

A function $f : \mathbb{R}^n \times [0, \infty) \mapsto \mathbb{R}$ is Lipschitz continuous on $\mathcal{X} \subset \mathbb{R}^n$ if there exists some constant $K > 0$ such that $\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$ for all $x_1, x_2 \in \mathcal{X}$, where $\|\cdot\|$ is the Euclidean norm. A function f is locally Lipschitz continuous if there exist constants $\tau > 0$ and $K > 0$ such that $\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$ for all $\|x_1 - x_2\| \leq \tau$.

A continuous function $\alpha : [0, a) \mapsto [0, \infty)$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. A continuous function $\alpha : [-b, a) \mapsto (-\infty, \infty)$ is said to belong to extended class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$ for some $a, b > 0$. The following lemma gives smooth approximations of $\min$ and $\max$ operators.

Lemma 1 (Approximation of $\min$ and $\max$ Operators [20])**.** *Consider a set of functions $h_i(x, t)$. Then for $\lambda > 0$*

$$\min_i h_i(x,t) \geq -\ln\left(\sum_{i=1}^{k} \exp\left(-h_i(x,t)\right)\right),$$
$$\max_i h_i(x,t) \geq \frac{\sum_i h_i(x,t)\exp(\lambda h_i(x,t))}{\sum_i \exp(\lambda h_i(x,t))}.$$

A. Linear Temporal Logic without Next Operator

In this subsection, we present background on LTL without next operator, denoted as LTL$_{\setminus\bigcirc}$. An LTL$_{\setminus\bigcirc}$ formula is defined inductively as $\varphi = \top \mid \pi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathbf{U} \varphi_2$ [1]. Other Boolean and temporal operators are defined inductively. LTL specification with implication ($\phi \implies \psi$) is defined as $\neg\phi \wedge \psi$. LTL specification involving eventually operator $\diamond\pi$ is equivalent to $\top\mathbf{U}\pi$. LTL specification involving always operator $\Box\pi$ can be rewritten as $\neg\diamond\neg\pi$. When the context is clear, we abuse the notions of LTL specification and LTL$_{\setminus\bigcirc}$ in the remainder of this paper.

Given an atomic proposition set Π, we augment Π so that every $\pi \in \Pi$ implies $\neg\pi \in \Pi$. We interpret LTL formulae over infinite words in 2^{Π}. Formula $\phi \wedge \psi$ is true iff ϕ and ψ are both true. Formula $\phi\mathbf{U}\psi$ is true iff ϕ remains true until ψ becomes true. A word η that satisfies an LTL formula ϕ is denoted as $\eta \models \phi$.

A deterministic Rabin automaton (DRA) can be used to represent an LTL formula φ. A DRA is defined as follows.

Definition 1 (Deterministic Rabin Automaton (DRA) [1])**.** *A Deterministic Rabin Automaton (DRA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is the finite set of alphabet, $\delta : Q \times \Sigma \mapsto Q$ is a finite set of transitions, $q_0 \in Q$ is the initial state, and $F = \{(B(1), \Gamma(1)), \ldots, (B(S), \Gamma(S))\}$ is a finite set of Rabin pairs such that $B(s), \Gamma(s) \subseteq Q$ for all $s = 1, \ldots, S$ with S being a positive integer.*

Every LTL formula can be represented by a DRA. Translation from LTL to DRA has been extensively studied [1, Ch.5], and is beyond the scope of this paper. Given a state $q \in Q$, we define the set of neighbor states of q as $\mathcal{N}(q) = \underset{\sigma\in\Sigma}{\cup} \delta(q, \sigma) \setminus \{q\}$, i.e., a state $q' \neq q$ is a neighbor state of q if there exists some transition δ such that q can transit to q'. A run η on $\mathcal{A}$ is an accepting run if there exists a pair $(B(s), \Gamma(s))$ such that η intersects with $B(s)$ finitely many times and intersects with $\Gamma(s)$ infinitely many times. Given the DRA $\mathcal{A}$, we can find the set of accepting runs on $\mathcal{A}$ by a graph search algorithm [1]. Any accepting run of the automaton can be divided into two parts including a prefix η_{pref} and a suffix η_{suff} that is repeated infinitely often [1, Ch.4]. Both the prefix and suffix can be regarded as finite runs, and η can be represented as $\eta = \eta_{pref}(\eta_{suff})^{\omega}$.

IV. Problem Formulation

Consider a continuous-time control-affine system

$$\dot{x} = f(x) + g(x)u, \quad x(0) = x_0, \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the system state, and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is input provided by the controller. The initial state at time $t = 0$ is denoted as $x(0) = x_0$. Vector fields f and g are locally Lipschitz continuous. Given the current system state x, a feedback controller is a function $\mu : \mathcal{X} \times [0, \infty) \mapsto \mathcal{U}$.

Let Π be a finite set of atomic propositions. We define a labeling function $L : \mathcal{X} \mapsto 2^{\Pi}$ that maps any state $x \in \mathcal{X}$ to a subset of atomic propositions that hold true at x. We also define $[\![\pi]\!] = \{x | \pi \in L(x)\}$ to be the set of states that satisfies the atomic proposition $\pi \in \Pi$. In this work, we assume that $[\![\pi]\!]$ is a closed set for all $\pi \in \Pi$, and $[\![\pi]\!]$ can be represented as $[\![\pi]\!] = \{x | Z_\pi(x) \geq 0\}$, where $Z_\pi : \mathbb{R}^n \mapsto \mathbb{R}$ is a bounded and continuously differentiable function. We slightly overload the notation $[\![\cdot]\!]$, and define the states that satisfy a subset of atomic propositions $P \in 2^{\Pi}$ as

$$[\![P]\!] = \begin{cases} \mathcal{X} \setminus \cup_{\pi \in \Pi} [\![\pi]\!] & \text{if } P = \emptyset \\ \cap_{\pi \in P} [\![\pi]\!] \setminus \cup_{\pi \in \Pi \setminus P} [\![\pi]\!] & \text{otherwise} \end{cases} \tag{2}$$

That is, $[\![P]\!]$ is the subset of system states $\mathcal{X}$ that satisfy all and only propositions in P [21].

We define the trajectory of system (1) as $\mathbf{x} : [0, \infty) \mapsto \mathcal{X}$ that maps from any time $t \geq 0$ to the system state $x(t)$. We then define the trace of a trajectory $\mathbf{x}$ as follows.

Definition 2 (Trace of Trajectory [21])**.** *An infinite sequence $Trace(\mathbf{x}) = P_0, P_1, \ldots$, where $P_i \in 2^{\Pi}$ for all $i = 0, 1, \ldots$ is a trace of a trajectory $\mathbf{x}$ if there exists an associated time sequence $t_0, t_1, \ldots$ of time instants such that*

1) $t_0 = 0$
2) $t_\tau \to \infty$ *as* $\tau \to \infty$
3) $t_i < t_{i+1}$
4) $x(t_i) \in [\![P_i]\!]$
5) *if $P_i \neq P_{i+1}$, then there exists some $t'_i \in [t_i, t_{i+1}]$ such that $x(t) \in [\![P_i]\!]$ for all $t \in (t_i, t'_i)$, $x(t) \in [\![P_{i+1}]\!]$ for all $t \in (t'_i, t_{i+1})$, and either $x(t'_i) \in [\![P_i]\!]$ or $x(t'_i) \in [\![P_{i+1}]\!]$.*

The trace of the system trajectory gives the sequence of atomic propositions satisfied by the system, and thus bridges the system behavior with temporal logic specification. Given a controller μ, we denote the trajectory under controller μ as $\mathbf{x}^\mu$. The trace of trajectory $\mathbf{x}^\mu$ is denoted by $Trace(\mathbf{x}^\mu)$. Suppose a specification φ belonging to $\text{LTL}_{\setminus \bigcirc}$ is given to system (1). If $Trace(\mathbf{x}^\mu) \models \varphi$, we say system (1) satisfies φ under controller μ, or controller μ satisfies φ. We state the problem of interest as follows.

Problem 1. *Compute a feedback controller $\mu : \mathcal{X} \times [0, \infty) \mapsto \mathcal{U}$ under which system (1) satisfies the given LTL specification φ belonging to $LTL_{\setminus \bigcirc}$. That is, compute a controller μ such that $Trace\,(\mathbf{x}^\mu) \models \varphi$.*

V. Solution Approach

We present a framework to solve Problem 1 in this section. We first introduce two types of CBFs. Given an LTL specification φ, we then present how to design CBFs using the automaton of the LTL specification. We construct a sequence of LTL formulae that correspond to an accepting run on the DRA of the LTL specification. Then we define a time varying CBF for each formula. We show that satisfying each formula is equivalent to guaranteeing the positivity of the corresponding CBF. Then we compute the controllers that ensure the CBF associated with each formula to be positive, and hence satisfies the LTL specification.

A. Control Barrier Function

In the following, we introduce time varying zeroing CBF (ZCBF) [22] and finite time convergence CBF (FCBF).

Definition 3 (Time Varying Zeroing CBF (ZCBF) [22])**.** *Consider a dynamical system (1) and a continuously differentiable function $h : \mathcal{X} \times [0, \infty) \mapsto \mathbb{R}$. If there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that for all $x(t) \in \mathcal{X}$ the following inequality holds*

$$\sup_{u \in \mathcal{U}} \left\{ \frac{\partial h(x,t)}{\partial x} f(x) + \frac{\partial h(x,t)}{\partial x} g(x) u + \frac{\partial h(x,t)}{\partial t} + \alpha(h(x,t)) \right\} \geq 0, \tag{3}$$

then function h is a ZCBF.

Given a ZCBF h, the set of controllers satisfying (3) is represented as $\mathcal{U}_Z(x,t) = \{\mu | \frac{\partial h(x,t)}{\partial x} f(x) + \frac{\partial h(x,t)}{\partial x} g(x) \mu(x,t) + \frac{\partial h(x,t)}{\partial t} + \alpha(h(x,t)) \geq 0\}$. The following proposition [22] characterizes $\mathcal{U}_Z(x,t)$.

Proposition 1. *Let $\mathcal{C}(t) = \{x | h(x,t) \geq 0\}$, where $h : \mathcal{X} \times [0, \infty) \mapsto \mathbb{R}$. Consider a feedback controller $\mu(x,t) \in \mathcal{U}_Z(x,t)$. If h is a ZCBF, then for all $x \in \mathcal{C}(t)$ and $t \geq 0$, $\mu(x,t)$ guarantees the set $\mathcal{C}(t)$ to be forward invariant.*

Motivated by finite time convergence CBF in [23], we define time varying FCBF as follows.

Definition 4 (Time Varying Finite Time Convergence CBF (FCBF))**.** *Consider a dynamical system (1) and a continuously differentiable function $h : \mathcal{X} \times [0, \infty) \mapsto \mathbb{R}$. If there exist constants $\rho \in [0, 1)$ and $\gamma > 0$ such that for all $x(t) \in \mathcal{X}$ the following inequality holds*

$$\sup_{u \in \mathcal{U}} \left\{ \frac{\partial h(x,t)}{\partial x} f(x) + \frac{\partial h(x,t)}{\partial x} g(x) u + \frac{\partial h(x,t)}{\partial t} + \gamma \cdot \mathit{sgn}(h(x,t)) |h(x,t)|^\rho \right\} \geq 0, \tag{4}$$

then function h is a FCBF.

Given an FCBF h, the set of controllers that satisfy (4) is represented as $\mathcal{U}_F(x,t) = \{\mu | \frac{\partial h(x,t)}{\partial x} f(x) + \frac{\partial h(x,t)}{\partial x} g(x) \mu(x,t) + \frac{\partial h(x,t)}{\partial t} + \gamma \cdot \text{sgn}(h(x,t)) |h(x,t)|^\rho \geq 0\}$. The following proposition extends the result in [23] on time invariant FCBF and characterizes $\mathcal{U}_F(x,t)$.

Proposition 2. *Let $\mathcal{C}(t) = \{x | h(x,t) \geq 0\}$, where $h : \mathcal{X} \times [0,\infty) \mapsto \mathbb{R}$. Consider a feedback controller $\mu(x,t) \in \mathcal{U}_F(x,t)$. If h is an FCBF, then for any initial state $x_0 \in \mathcal{X}$, controller $\mu(x,t)$ guarantees that the system will be steered to the set $\mathcal{C}(t)$ within finite time $0 < T < \infty$ such that $x(T) \in \mathcal{C}(T)$. The convergence time $T = \frac{|h(x_0,0)|^{(1-\rho)}}{\gamma(1-\rho)}$. Moreover, the system remains in $\mathcal{C}(t')$ for all $t' \geq T$.*

Proof. The proof follows [23]. Construct a Lyapunov function $V(x,t) = \max\{-h(x,t), 0\}$. We can verify that $V(x,t) = 0$ for all $x(t) \in \mathcal{C}(t)$, $V(x,t) > 0$ for all $x(t) \in \mathcal{X} \setminus \mathcal{C}(t)$, and $\frac{\mathrm{d}}{\mathrm{d}t}V(x,t) \leq \gamma V(x,t)^\rho$. By Theorem 4.1 in [24], finite time stability holds for system (1). Thus if $x_0 \in \mathcal{C}(t)$, controllers in $\mathcal{U}_F(x,t)$ render $\mathcal{C}(t)$ forward invariant. If $x_0 \notin \mathcal{C}(t)$, then the system converges to $\mathcal{C}(t)$ within finite time $T = \frac{|h(x_0,0)|^{(1-\rho)}}{\gamma(1-\rho)}$. □

B. Design of Control Barrier Functions

In this subsection, we first construct a sequence of LTL formulae so that satisfying all formulae is equivalent to satisfying the given LTL formula φ. Then we show how to design time varying CBFs for each formula.

Given an LTL specification φ, we compute the DRA associated with φ, and pick an accepting run $\eta = q_0, \ldots, q_J, (q_{J+1}, \ldots, q_{J+N})^\omega$ of the DRA. Although a similar idea is used in [16], [17], in which a sequence of reachability problems is selected, as we will show later, our proposed CBF design enhances the feasibility of control synthesis. The complexity of constructing the DRA is doubly exponential in the size of the formula in the worst-case as in the existing works on both abstraction-free and abstraction-based LTL synthesis [6], [10], [21]. However, we note that several important classes of LTL formulas have DRAs of polynomial size [25], and that our approach mitigates the exponential complexity of computing a finite state abstraction.

Rewriting η into prefix-suffix form, we have that the sequence of states $q_0, \ldots, q_J$ forms the prefix η_{pref}, and the sequence of states $q_{J+1}, \ldots, q_{J+N}$ forms the suffix η_{suff}. We denote the transition from state q_j to q_{j+1} as η_j, and denote the input word of transition η_j as ϕ_j. The input word ϕ_j is in the form of conjunction or disjunction of atomic propositions [1], i.e., $\phi_j = \pi_1 \bowtie \ldots \bowtie \pi_k$ where $\pi_i \in \Pi$ is an atomic proposition for all $i = 1, \ldots, k$ and $\bowtie \in \{\wedge, \vee\}$.

Given the accepting run η, we construct a sequence of formulae $\{\psi_j | j = 0, 1, \ldots, J+N\}$, where ψ_j corresponds to transition η_j as follows. We denote the input word corresponding to the self-transition at state q_j as Φ_j, i.e., $\delta(q_j, \Phi_j) = q_j$. We also note that ϕ_j is the input word corresponding to a transition from q_j to q_{j+1}. We then construct a formula ψ_j corresponding to transition η_j as

$$\psi_j = \Phi_j \ \mathbf{U} \ \square \left(\phi_j \wedge \Phi_{j+1}\right). \tag{5}$$

Formula ψ_j indicates that no transition starting from state q should occur except self-transition and η_j. Since both prefix and suffix of η are over finite horizon, only a finite number of formulae $\{\psi_j | j = 0, 1, \ldots, J+N\}$ are generated.

We then assign a sequence of time instants $t_1 < \ldots < t_J$ as the deadlines of each transition $\eta_0, \eta_1, \ldots, \eta_J$ of η_{pref}. The deadlines of the transitions of the suffix can be generated as $n\Delta + t_{J+1}$, where n is a nonnegative integer and $\Delta \geq 0$. We additionally let $t_0 = 0 < t_1$. We define the active time of each formula ψ_j as $[t_j, t_{j+1}]$, during which formula ψ_j must be satisfied. There are two advantages of defining the active time of each formula ψ_j. First, although each formula ψ_j needs to be interpreted over infinite runs, the active time enables us to interpret each ψ_j over finite runs. That is, formula ψ_j needs to be satisfied during $[t_j, t_{j+1}]$. For time $t > t_{j+1}$, formula ψ_j can be violated. Second, the active time allows our approach to satisfy multiple, sequential constraints (e.g., reaching disjoint regions A and B) that cannot be satisfied simultaneously.

Given a time interval $[t, t']$ and controller μ, we let $\mathbf{x}^\mu([t,t'])$ be the system trajectory during time interval $[t, t']$ under controller μ. The trace of $\mathbf{x}^\mu([t,t'])$ is denoted as $Trace\left(\mathbf{x}^\mu([t,t'])\right)$. Denote the system state under controller μ at time t as $\mathbf{x}^\mu(t)$. We then show the effectiveness of (5) by analyzing the relationship between satisfying each formula ψ_j and run η.

Lemma 2. *Let η be an accepting run and ψ_j be a formula in the form of* (5) *whose active time is $[t_j, t_{j+1}]$. If $Trace\left(\mathbf{x}^\mu([0,t_j])\right)$ steers the DRA from q_0 to q_j, and $Trace\left(\mathbf{x}^\mu([t_j,t_{j+1}])\right) \models \psi_j$, then the DRA transitions from state q_j to q_{j+1} during time interval $[t_j, t_{j+1}]$. Moreover, the DRA remains in state q_{j+1} until at least time t_{j+1}.*

Proof. We first prove that if $Trace\left(\mathbf{x}^\mu([0,t_j])\right)$ steers the DRA from q_0 to q_j, and $Trace\left(\mathbf{x}^\mu([t_j,t_{j+1}])\right) \models \psi_j$, then the DRA transitions from state q_j to q_{j+1} during time interval $[t_j, t_{j+1}]$. We prove by contradiction. Suppose the current state of the DRA is q_j and $Trace\left(\mathbf{x}^\mu([t_j,t_{j+1}])\right) \models \psi_j$, while the DRA transitions from state q_j to some state $q' \neq q_{j+1}$. By the semantics of until operator $\mathbf{U}$, there must exist some time $t \in [t_j, t_{j+1}]$ such that $L(\mathbf{x}^\mu(t')) \models \phi_j \wedge \Phi_{j+1}$ for all $t' \in [t, t_{j+1}]$ in order to make $Trace\left(\mathbf{x}^\mu([t_j,t_{j+1}])\right) \models \psi_j$ hold. Then by the semantics of and operator $\wedge$, $L(\mathbf{x}^\mu(t)) \models \phi_j \wedge \Phi_{j+1}$ implies that $L(\mathbf{x}^\mu(t)) \models \phi_j$. Since ϕ_j is the input associated with the transition from q_j to q_{j+1}, then $q' = q_{j+1}$. Otherwise, the DRA contains nondeterminism which conflicts Definition 1.

We then prove that the DRA remains in q_{j+1} until at least time t_{j+1}. Suppose the DRA transitions from q_{j+1} to some state q before t_{j+1}. This is equivalent to the fact that there exist some state $q \in \mathcal{N}(q_{j+1})$ and time $t \in [t_j, t_{j+1}]$ such that $L(\mathbf{x}^\mu(t)) \models \phi^q_{j+1}$, where ϕ^q_{j+1} is the input word associated with transition from state q_{j+1} to some neighbor state q. However, this contradicts Φ_{j+1}, and thus the DRA cannot transition to some state $q \in \mathcal{N}(q_{j+1})$. By the definition of neighbor states $\mathcal{N}(q_{j+1})$, the DRA can only take the self-transition at q_{j+1}. □

Inducting the results on Lemma 2 yields Corollary 1.

Corollary 1. *If $Trace\left(\mathbf{x}^\mu([t_j,t_{j+1}])\right) \models \psi_j$ for all $j = 0, 1, \ldots$, then $Trace\left(\mathbf{x}^\mu\right) \models \varphi$.*

Lemma 2 and Corollary 1 imply that, in order to ensure that the specification is satisfied, it suffices to ensure that the trajectory under controller μ satisfies each ψ_j within its active time $[t_j, t_{j+1}]$. In what follows, we construct a set of CBFs that will be used to ensure satisfaction of each ψ_j.

In the following, we design CBFs for Φ_j and $\phi_j \wedge \Phi_{j+1}$. We first define a CBF h_π for each atomic proposition π that is involved in ψ_j. We consider CBFs in the form of $h_\pi(x,t) = M_\pi(t) + Z_\pi(x)$ for all π, where $M_\pi(t)$ and $Z_\pi(x)$ are called guard function and state function, respectively. The state function $Z_\pi(x)$ is a function of state x that captures if the state x is in $[\![\pi]\!]$, i.e., $[\![\pi]\!] = \{x | Z_\pi(x) \geq 0\}$. The guard function $M_\pi(t) = \frac{E_\pi}{1+e^{-b_\pi(t+c_\pi)}} - \epsilon_\pi$ is a logistic function, where $E_\pi > 0$, $b_\pi > 0$, and $\epsilon_\pi \geq 0$. The guard function $M_\pi(t)$ is introduced so that each atomic proposition π, and hence ψ_j, only need to be satisfied during their active time.

We then show how to choose E_π, b_π, c_π, and ϵ_π for each π. First, if atomic proposition π is satisfied at time $t = 0$, then $h_\pi(x_0, 0) \geq 0$. If π is not satisfied at time $t = 0$, then $h_\pi(x_0, 0) < 0$. These two requirements are captured by (6a) and (6b). Second, given the deadline t_j of atomic proposition π, we have $M_\pi(t_j) \leq 0$, as shown in (6c). To summarize, we have the following inequalities:

$$\frac{E_\pi}{1+e^{-b_\pi c_\pi}} - \epsilon_\pi + Z_\pi(x_0) \geq 0, \text{ if } \pi \in L(x_0) \tag{6a}$$

$$\frac{E_\pi}{1+e^{-b_\pi c_\pi}} - \epsilon_\pi + Z_\pi(x_0) < 0, \text{ if } \pi \notin L(x_0) \tag{6b}$$

$$\frac{E_\pi}{1+e^{-b_\pi (t_j + c_\pi)}} - \epsilon_\pi \leq 0, \tag{6c}$$

$$E_\pi > 0, b_\pi > 0, \epsilon_\pi \geq 0 \tag{6d}$$

Inequalities (6) are solved as follows. We first pick some $b_\pi > 0$ and c_π such that $c_\pi \leq t_{j+1}$ if π is involved in Φ_j or ϕ_j, and $c_\pi \leq t_{j+2}$ if π is involved in Φ_{j+1}. Fixing the values of b_π and c_π, then E_π and ϵ_π can be obtained by solving the linear inequalities (6). We characterize the CBFs obtained by solving (6) using the following lemma.

Lemma 3. *Let t_j be the deadline of atomic proposition π, and h_π be the CBF obtained by solving* (6)*. For any $t \leq t_j$, if $h_\pi(x,t) \geq 0$, then $x(t) \in [\![\pi]\!]$.*

Proof. Inequality (6c) indicates that $M_\pi(t_j) \leq 0$, where t_j is the deadline of π. By (6d), the guard function $M_\pi(t)$ is monotone increasing. Therefore, for all $t \leq t_j$, $M_\pi(t) \leq 0$ holds. By the definition of $h_\pi(x,t)$, we have that $Z_\pi(x) = h_\pi(x,t) - M_\pi(t)$. Given $M_\pi(t) \leq 0$ for all $t \leq t_j$ and $h_\pi(x,t) \geq 0$, we have $Z_\pi(x) \geq 0$, and thus the system state $x(t)$ is in the region $\{x(t) | Z_\pi(x(t)) \geq 0\} = [\![\pi]\!]$. □

Given a CBF h_π for each atomic proposition π that is involved in ψ_j, we compute the CBFs for Φ_j and $\phi_j \wedge \Phi_{j+1}$. We note that Φ_j and $\phi_j \wedge \Phi_{j+1}$ are both in the forms of conjunctions/disjunctions of atomic propositions [1]. We utilize the following definition to construct the CBF for Φ_j and $\phi_j \wedge \Phi_{j+1}$.

Definition 5. *Consider a set of atomic proposition $\{\pi_i | i = 1, \ldots, k\}$. Let $h_{\pi_i} : \mathbb{R}^n \times [0, \infty) \mapsto \mathbb{R}$ be the CBF of each π_i defined as $h_{\pi_i}(x,t) = M_{\pi_i}(t) + Z_{\pi_i}(x)$ for each atomic proposition π_i, where $M_{\pi_i}(t)$ is computed by* (6) *and $Z_{\pi_i}(t)$ is defined as $\{x | Z_{\pi_i}(x) \geq 0\} = [\![\pi_i]\!]$. Consider a formula $\phi' = \pi_1 \bowtie \ldots \bowtie \pi_{k-1}$, where $\bowtie \in \{\wedge, \vee\}$. Let $h_{\phi'}$ be the CBF of ϕ'. Then the CBF of formula $\phi = \phi' \wedge \pi_k$ is*

$$h_\phi(x,t) = -\ln\left[\exp\left(-h_{\phi'}(x,t)\right) + \exp\left(-h_{\pi_k}(x,t)\right)\right]. \tag{7}$$

The CBF h_ϕ of formula $\phi = \phi' \vee \pi_k$ for some $\lambda > 0$ is

$$h_\phi(x,t) = \frac{h_{\phi'}(x,t)e^{\lambda h_{\phi'}(x,t)} + h_{\pi_k}(x,t)e^{\lambda h_{\pi_k}(x,t)}}{e^{\lambda h_{\phi'}(x,t)} + e^{\lambda h_{\pi_k}(x,t)}}. \tag{8}$$

Definition 5 recursively defines the CBF for a formula ϕ in the form of $\phi = \pi_1 \bowtie \ldots \bowtie \pi_k$, where $\bowtie \in \{\wedge, \vee\}$. By Lemma 1, we have that CBFs (7) and (8) bound $\min\{h_{\phi'}(x,t), h_{\pi_k}(x,t)\}$ and $\max\{h_{\phi'}(x,t), h_{\pi_k}(x,t)\}$ from below, respectively. When Φ_j and $\phi_j \wedge \Phi_{j+1}$ are in the forms of $\pi_1 \bowtie \ldots \bowtie \pi_k$, where $\bowtie \in \{\wedge, \vee\}$, their CBFs can be obtained by recursively applying Definition 5.

Algorithm 1 Algorithm for computing the CBFs for each formula ψ_j.

```
procedure CBF_DESIGN($\varphi$)
    Input: LTL specification $\varphi$
    Output: CBFs for each formula $\psi_j$
    Compute the DRA associated with LTL specification
      $\varphi$, and the set of accepting runs on the DRA.
    Pick an accepting run $\eta$ on the DRA, and identify
      each formula $\psi_j$ associated with each transition $\eta_j$ in $\eta$
      as (5).
    Specify a sequence of time $0 < t_1 < \ldots$ for
      accepting run $\eta$.
    Pick a set of feasible coefficients for relations (6) for
      each atomic proposition $\pi$ involved in $\psi_j$.
    Recursively compute CBFs for $\Phi_j$ and $\phi_j \wedge \Phi_{j+1}$
      using Definition 5.
    return CBFs for $\Phi_j$ and $\phi_j \wedge \Phi_{j+1}$
end procedure
```

The procedure we used to design the CBFs for each ψ_j is presented in Algorithm 1. We characterize the construction of CBFs for Φ_j and $\phi_j \wedge \Phi_{j+1}$ using the following proposition.

Lemma 4. *Let h_ϕ be the CBF obtained by Algorithm 1, where $\phi \in \{\Phi_j, \phi_j \wedge \Phi_{j+1}\}$. For any time $t \in [t_j, t_{j+1}]$, if $h_\phi(x,t) \geq 0$, then $L(x(t)) \models \phi$.*

Proof. We prove by induction. Consider $\phi = \pi_1 \wedge \pi_2$. Then h_ϕ is computed as (7). By Lemma 1, $h_\phi(x,t) \geq 0$ implies that $h_{\pi_1}(x,t) \geq 0$ and $h_{\pi_2}(x,t) \geq 0$. By Lemma 3, $x(t) \in [\![\pi_1]\!] \cap [\![\pi_2]\!]$, and thus ϕ is satisfied. Consider $\phi = \pi_1 \vee \pi_2$. Then h_ϕ is computed as (8). By Lemma 1, $h_\phi(x,t) \geq 0$ implies that $h_{\pi_1}(x,t) \geq 0$ or $h_{\pi_2}(x,t) \geq 0$. By Lemma 3, $x(t) \in [\![\pi_1]\!] \cup [\![\pi_2]\!]$, and thus ϕ is satisfied. These two cases serve as our induction base.

Suppose the lemma holds after applying (7) and (8) $k-1$ times, and denote the corresponding CBF as h_{ϕ}^{k-1}. If the k-th operation is a conjunction with atomic proposition π, then $h_{\phi}^{k}(x,t)$ is obtained by (7). By Lemma 1, $h_{\phi}^{k}(x,t)\geq 0$ implies that $h_{\pi}(x,t)\geq 0$ and $h_{\phi}^{k-1}(x,t)\geq 0$. By Lemma 3, $h_{\pi}(x,t)\geq 0$ implies $x(t)\in[\![\pi]\!]$. By our inductive hypothesis, $h_{\phi}^{k-1}(x,t)\geq 0$ implies $L(x(t))\models\phi$ after applying (7) and (8) $k-1$ times. Combining the arguments above, we have $L(x(t))\models\phi$ when the k-th operator is a conjunction. If the k-th operation is a disjunction with atomic proposition π, then $h_{\phi}^{k}(x,t)$ is obtained by (8). By Lemma 1, $h_{\phi}^{k}(x,t)\geq 0$ implies that $h_{\pi}(x,t)\geq 0$ or $h_{\phi}^{k-1}(x,t)\geq 0$. Similar to the conjunction case, we can conclude $L(x(t))\models\phi$ when the k-th operator is a disjunction. Since ϕ is in the form of conjunction and disjunction of finite number of atomic propositions, the lemma holds by induction. □

C. CBF-based Controller Synthesis

In this subsection, we present how to compute the controllers that satisfy each formula ψ_j in the form of (5). We then show that by satisfying all formulae ψ_j for all j, the LTL specification φ is satisfied.

Lemma 5. *Consider a formula ψ_j in the form of* (5) *whose active time is $[t_j,t_{j+1}]$. Let h_{Φ_j} and h_{Ω_j} be the CBFs of Φ_j and $\phi_j\wedge\Phi_{j+1}$ obtained using Algorithm 1, respectively. Then for all $t\in[t_j,t_{j+1}]$ any feedback controller in*

$$\mathcal{U}_{\psi_j}(x,t)=\begin{cases}\mathcal{U}_2(x,t), & \textit{if } L(x(t))\models\phi_j\wedge\Phi_{j+1},\\ \mathcal{U}_1(x,t)\cap\mathcal{U}_2(x,t), & \textit{if } L(x(t))\models\Phi_j,\\ \emptyset, & \textit{otherwise}\end{cases}$$

satisfies $Trace(\mathbf{x}^{\mu}([t_j,t_{j+1}]))\models\psi_j$, where

$$\mathcal{U}_1(x,t)=\Big\{\mu|\frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial x}f(x)+\frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial x}g(x)\mu(x,t)+\frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial t}+\alpha(h_{\Phi_j\Omega_j}(x,t))\geq 0\Big\},$$

$$\mathcal{U}_2(x,t)=\Big\{\mu|\frac{\partial h_{\Omega_j}(x,t)}{\partial x}f(x)+\frac{\partial h_{\Omega_j}(x,t)}{\partial x}g(x)\mu(x,t)+\frac{\partial h_{\Omega_j}(x,t)}{\partial t}+\gamma\cdot \textit{sgn}(h_{\Omega_j}(x,t))|h_{\Omega_j}(x,t)|^{\rho}\geq 0\Big\}$$

$$h_{\Phi_j\Omega_j}(x,t)=\frac{h_{\Phi_j}(x,t)e^{\lambda h_{\Phi_j}(x,t)}+h_{\Omega_j}(x,t)e^{\lambda h_{\Omega_j}(x,t)}}{e^{\lambda h_{\Phi_j}(x,t)}+e^{\lambda h_{\Omega_j}(x,t)}}.$$

Proof. By the semantics of until operator $\mathbf{U}$ and Definition 2, it suffices to show that there exists a time sequence t_j,T,t_{j+1} such that (i) $t_j<t_{j+1}$, (ii) $T\in[t_j,t_{j+1}]$, and (iii) $L(x(t))\models\Phi_j$ for all $t\in[t_j,T)$ and $L(x(t))\models\phi_j\wedge\Phi_{j+1}$ for all $t\in[T,t_{j+1}]$. We show that each of these conditions is satisfied.

First, condition $t_j<t_{j+1}$ holds by the construction of deadlines as given in Section V-B. To guarantee $T\in[t_j,t_{j+1}]$, we can tune parameters ρ and γ as given in Proposition 2 so that $T\leq t_{j+1}$. In the following, we show that there exists some time $T\in[t_j,t_{j+1}]$ such that $L(x(t))\models\Phi_j$ for all $t\in[t_j,T)$ and $L(x(t))\models\phi_j\wedge\Phi_{j+1}$ for all $t\in[T,t_{j+1}]$.

We start with the case where $L(x(t))\models\phi_j\wedge\Phi_j$. In this case, $T=t$ and we need to guarantee $L(x(t'))\models\phi_j\wedge\Phi_{j+1}$ for all $t'\in[t,t_{j+1}]$ so that formula ψ_j is satisfied. By Proposition 2, the set of controllers $\mathcal{U}_2(x,t)$ ensures that the system remains in the set $\{x|h_{\Omega_j}(x,t')\geq 0\}$ for all $t'\geq t$. By Lemma 4, we have that $L(x(t'))\models\phi_j\wedge\Phi_j$ for all $x(t')\in\{x(t)|h_{\Omega_j}(x,t')\geq 0\}$ for all $t'\in[t,t_{j+1}]$. Therefore, $Trace(\mathbf{x}^{\mu}([t,t_{j+1}]))\models\psi_j$ for all $t\in[t_j,t_{j+1}]$ in this case.

We then consider the case where $L(x(t))\models\Phi_j$. By Proposition 1, the set of controllers $\mathcal{U}_1(x,t)$ guarantees that the system remains in the set $\{x(t')|h_{\Phi_j\Omega_j}(x,t')\geq 0\}$ for all $t'\in[t,t_{t+1}]$. By Lemma 4, we have that $L(x(t'))\models\Phi_j$, or $L(x(t'))\models\phi_j\wedge\Phi_{j+1}$, or both, for all $x(t')\in\{x(t')|h_{\Phi_j\Omega_j}(x,t')\geq 0\}$ where $t'\in[t,t_{j+1}]$. By Proposition 2, the set of controllers $\mathcal{U}_2(x,t)$ ensures that the system will be steered to the set $\{x|h_{\Omega_j}(x,t)\geq 0\}$ at some time $T\geq t$ if $x(t)\notin\{x|h_{\Omega_j}(x,t)\geq 0\}$. Moreover, the system remains in $x(t')\in\{x|h_{\Omega_j}(x,t')\geq 0\}$ for all $t'\in[T,t_{j+1}]$. By Lemma 4, we have that $Trace(\mathbf{x}^{\mu}([t,t_{j+1}]))\models\Box(\phi_j\wedge\Phi_{j+1})$ for all $t\in[t_j,t_{j+1}]$. Thus, the controllers in $\mathcal{U}_1(x,t)\cap\mathcal{U}_2(x,t)$ guarantee $Trace(\mathbf{x}^{\mu}([t,t_{j+1}]))\models\psi_j$ for all $t\in[t_j,t_{j+1}]$.

Combining the arguments above, we have that the controllers in $\mathcal{U}_{\psi_j}(x,t)$ satisfy $Trace(\mathbf{x}^{\mu}([t_j,t_{j+1}]))\models\psi_j$. □

We note that the computation of controllers can be simplified when the formula ψ_j is in some simple forms. Consider a formula ψ_j as per (5). When $\Phi_j=\top$, then $\psi_j=\Box(\phi_j\wedge\Phi_{j+1})$. The controllers satisfying ψ_j in this case are given by the following corollary.

Corollary 2. *Suppose $\psi_j=\Box\phi$. Let h_ϕ be the CBF of ϕ that is obtained using Algorithm 1. Then any feedback controller in*

$$\mathcal{U}_{\psi_j}(x,t)=\begin{cases}\big\{\mu|\frac{\partial h_{\phi}(x,t)}{\partial x}f(x)+\frac{\partial h_{\phi}(x,t)}{\partial x}g(x)\mu(x,t)+\\ \frac{\partial h_{\phi}(x,t)}{\partial t}+\alpha(h_{\phi}(x,t))\geq 0\big\}, & \textit{if } L(x_0)\models\phi\\ \emptyset, \ \textit{otherwise}\end{cases}$$

satisfies ψ_j.

Proof. The proof follows by replacing Φ_j in Lemma 5 with unconditionally true $\top$. □

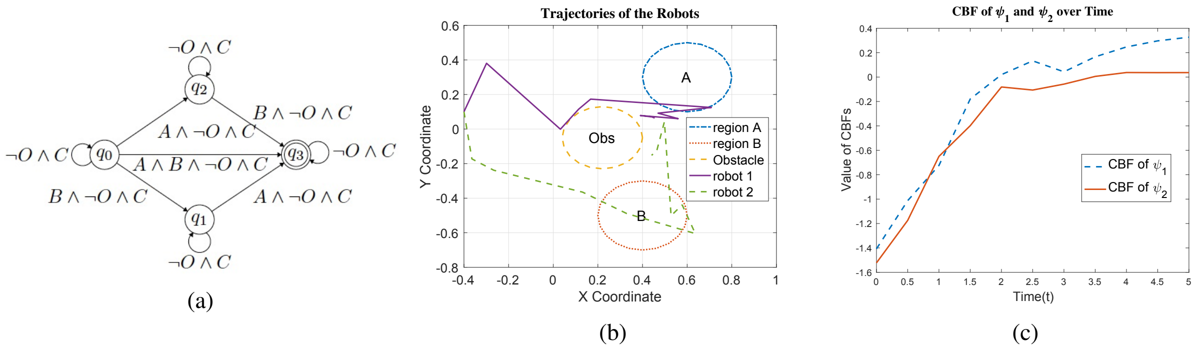
We finally show that by satisfying each formula ψ_j, specification φ is satisfied.

Theorem 1. *Applying the controllers in $\mathcal{U}_{\psi_j}(x,t)$ for all $t\geq 0$ as given in Lemma 5 renders $Trace(\mathbf{x}^{\mu})\models\varphi$.*

Proof. Applying μ for each time interval $[t_j,t_{j+1}]$, we have that $Trace(\mathbf{x}^{\mu}([t_j,t_{j+1}]))\models\psi_j$ due to Lemma 5. Then according to Lemma 2 and Corollary 1, we have that satisfying the sequence of formulae ψ_j for all j is equivalent to executing the run η on the DRA. Since η is an accepting run, we can conclude that $Trace(\mathbf{x}^{\mu})\models\varphi$. □

Using the treatment in [15], [26], we can compute the control input $u(t)$ at each time t by solving the quadratic program (QP):

$$\begin{aligned}\min_{u}\quad & u^{T}Pu\\ \text{s.t.}\quad & u(t)\in\mathcal{U}_{\psi_j}(x,t),\ \forall\psi_j\end{aligned}$$



(a)

(b)

(c)

Fig. 1: Fig. 1a shows the DRA associated with $\varphi = \Diamond A \wedge \Diamond B \wedge \Box(\neg O \wedge C)$. It has one Rabin pair $(\emptyset, \{q_3\})$. Fig. 1b presents the trajectories of both robot using the proposed approach in this paper. The trajectory of the first robot is plotted in solid line, while the trajectory of the second robot is plotted in dotted line. The first robot eventually reaches region A, and the second robot eventually reaches region B. Both robots successfully avoid the obstacle region O and remain close enough to each other. CBFs h_{ψ_1} and h_{ψ_2} are presented in Fig. 1c. h_{ψ_1} is plotted in dotted line and h_{ψ_2} is plotted in solid line. Both CBFs become positive at some time $t \geq 0$, at which the formulae corresponding to the CBFs are satisfied.

where $P \in \mathbb{R}^{m\times m}$ is a positive semi-definite matrix that quantifies the cost of control, and $\mathcal{U}_{\psi_j}(x,t)$ is given by Lemma 5. The constraint set of this QP requires that the CBF constraints as given in Lemma 5 must be satisfied for each formula ψ_j. We note that the deadlines of formulae ψ_j's corresponding to transitions in η_{suff} are generated periodically as $n\Delta + t_{J+1}$. We can impose the constraint $u(t) \in \mathcal{U}_{\psi_i}(x,t)$ for $i \in \{j,\ldots,j+J+N\}$, resulting in a finite number of constraints in the QP. The controllers for future time can then be generated by implementing the controllers periodically.

VI. Simulation

In this section, we present a numerical case study on a two robot homogeneous multi-agent system. Consider a multi-agent system consisting of two robots whose dynamics are given as $\dot{x} = u$, where $x \in \mathbb{R}^4$ is the system state, and $u \in \mathbb{R}^4$ is the input. The state variables x_1 and x_2 give the coordinate of the first robot, and state variables x_3 and x_4 give the coordinate of the second robot. The initial positions for both robots are $[-0.4, 0.1]$.

Each of the robots has its respective goal region, denoted as region A and region B. Both robots are required to eventually reach their goal regions. In the meantime, both robots need to avoid the obstacle region, denoted as O. Furthermore, the robots keep exchanging information with each other, and hence must remain close enough. The tasks can be represented by an LTL formula $\varphi = \Diamond A \wedge \Diamond B \wedge \Box(\neg O \wedge C)$, where C represents the connectivity specification. In this case study, region A is modeled as $\{x|Z_A(x) \geq 0\}$ where $Z_A(x) = 0.2 - \|[x_1,x_2]-[0.6,0.3]\|_2$. Region B is modeled as $\{x|Z_B(x) \geq 0\}$ where $Z_B(x) = 0.2 - \|[x_3,x_4]-[0.4,-0.5]\|_2$. The obstacle region is modeled as $\{x|Z_O(x) \geq 0\}$ where $Z_O(x) = 0.18 - \|x - [0.22,-0.05,0.22,-0.05]\|_2$. The connectivity between the two robots is given as $C = \{x|Z_C(x) \geq 0\}$ where $Z_C(x) = \sqrt{x_3+0.39} - \|[x_1,x_2]-[x_3,x_4]\|_2$.

We compare the proposed approach with the approach proposed in [17]. Since there is single CBF for each region of interest, the relaxation proposed in [17] coincides with the traditional CBF-based approach. We observe that no feasible trajectory is synthesized since there is no feasible solution to the QP. The infeasibility of the QP is caused by the fact that the connectivity constraint requires the robots to stay close to each other, while steering the robots into region A and B makes them violate the connectivity constraint. This infeasibility agrees with the example presented in [16].

In the following, we demonstrate our proposed approach. Given the LTL specification φ, the DRA representing φ is shown in Fig. 1a. The accepting runs of the DRA include $q_0q_1(q_3)^\omega$, $q_0(q_3)^\omega$, and $q_0q_2(q_3)^\omega$. We pick the run $\eta = q_0q_1(q_3)^\omega$ in this case study. Transition from q_0 to q_1 of the accepting run η corresponds to formula $\psi_0 = (\neg O \wedge C)\mathbf{U}\Box(B \wedge \neg O \wedge C)$. Transition from q_1 to q_2 corresponds to formula $\psi_1 = (\neg O \wedge C)\mathbf{U}\Box(A \wedge \neg O \wedge C)$. Self transition at q_3 corresponds to formula $\psi_3 = \Box(\neg O \wedge C)$. Next, we assign the active time for each formulae during which the formula needs to be satisfied. In this case study, we let ψ_1 be satisfied during $[0,2]$, and let ψ_2 be satisfied during $[2,4]$. Using (6), we then construct the CBFs for atomic propositions B and A as $h_B(x,t) = \frac{1}{1+e^{-(t-1.5)}} - 0.63 + 0.2 - \|[x_1,x_2]-[0.4,-0.5]\|_2$, $h_A(x,t) = \frac{1}{1+e^{-(t-0.5)}} - 0.9 + 0.2 - \|[x_1,x_2]-[0.6,0.3]\|_2$, respectively. Then the CBF for each formula can be constructed by Definition 5. For instance, $h_{\psi_3}(x,t) = -\ln(\exp(-Z_O(x)) + \exp(-Z_C(x)))$.

In the following, we formulate the QP to solve for the controllers so that the LTL specification φ is satisfied. According to Lemma 5, we have the following constraints for each formula ψ_j:

$$\frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial x}f(x) + \frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial x}g(x)\mu(x,t) + \frac{\partial h_{\Phi_j\Omega_j}(x,t)}{\partial t} + \alpha(h_{\Phi_j\Omega_j}(x,t)) \geq 0 \tag{9}$$

$$\frac{\partial h_{\Omega_j}(x,t)}{\partial x}f(x) + \frac{\partial h_{\Omega_j}(x,t)}{\partial x}g(x)\mu(x,t) + \frac{\partial h_{\Omega_j}(x,t)}{\partial t} + \gamma \cdot \mathrm{sgn}(h_{\Omega_j}(x,t))|h_{\Omega_j}(x,t)|^{\rho} \geq 0 \quad (10)$$

where Φ_j and Ω_j are defined as given in Lemma 5, and

$$h_{\Phi_j\Omega_j}(x,t) = \frac{h_{\Phi_j}(x,t)e^{\lambda h_{\Phi_j}(x,t)} + h_{\Omega_j}(x,t)e^{\lambda h_{\Omega_j}(x,t)}}{e^{\lambda h_{\Phi_j}(x,t)} + e^{\lambda h_{\Omega_j}(x,t)}}.$$

Formulating (9) and (10) for all ψ_j's form the constraint set of the QP.

By solving the QP, we obtain the controllers for both robots. The trajectories of the robots are presented in Fig. 1b. We make the following observations. First, both robots eventually reach their target regions, while avoiding the obstacle region. Furthermore, the second robot reaches region B before the first robot reaches region A. This can be observed from Fig. 1c. CBF h_{ψ_1} turns positive before h_{ψ_2}. Second, after the robots reaching their goal regions, they can still leave the goal region rather than remaining in the goal region. This is because our design of CBF adopts the guarding function $M_\psi(t)$ which increases over time and hence enhances the feasibility of the QP. The relaxation introduced by the guarding function enables the satisfaction of the connectivity constraint. As we could observe in Fig. 1c, CBFs h_{ψ_1} and h_{ψ_2} remain positive after the robots reaching their goal regions.

VII. Conclusion

In this paper, we studied the problem of control synthesis for CPSs under LTL constraints modeled by LTL without next operator. We focused on synthesizing the controller to satisfy the LTL constraint without explicitly computing the abstraction of the CPS. A CBF-based approach is used in this paper. We first constructed a sequence of LTL formulae corresponding to an accepting run on the DRA, and presented a design rule to design time-varying CBFs for the sequence of formulae. We introduced a function named guard function when designing CBFs, which enhances feasibility of CBF constraints. We showed that the positivity of the CBF implies the satisfaction of the LTL formula. Then we showed how to satisfy the set of formulae by guaranteeing the positivities of their CBFs. We formulated a QP to compute a controller that satisfies the LTL specification. A numerical case study is presented to illustrate the proposed approach. In future work, we aim to jointly consider the selection of accepting run and CBF-based control synthesis.

References

[1] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.

[2] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[3] S. Coogan, E. A. Gol, M. Arcak, and C. Belta, "Traffic network control from temporal logic specifications," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 162–172, 2015.

[4] 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, 2000.

[5] 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.

[6] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of markov decision processes with linear temporal logic constraints," *Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[7] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning for dynamical systems," in *the Proc. of Intl. Conf. on Decision and Control (CDC)*. IEEE, 2009, pp. 5997–6004.

[8] L. Niu and A. Clark, "Optimal secure control with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, 2019.

[9] E. M. Wolff, U. Topcu, and R. M. Murray, "Optimization-based trajectory generation with linear temporal logic specifications," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5319–5325.

[10] M. B. Horowitz, E. M. Wolff, and R. M. Murray, "A compositional approach to stochastic optimal control with co-safe temporal logic specifications," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1466–1473.

[11] I. Papusha, J. Fu, U. Topcu, and R. M. Murray, "Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 434–440.

[12] H. J. Kappen, "Linear theory for control of nonlinear stochastic systems," *Physical review letters*, vol. 95, no. 20, p. 200201, 2005.

[13] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[14] 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, 2017.

[15] 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, 2016.

[16] M. Srinivasan and S. Coogan, "Control of mobile robots using barrier functions under temporal logic specifications," *arXiv preprint arXiv:1908.04903*, 2019.

[17] 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)*. IEEE, 2018, pp. 1991–1996.

[18] 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, 2018.

[19] G. Yang, R. Tron, and C. Belta, "Continuous-time signal temporal logic planning with control barrier function," *arXiv preprint arXiv:1903.03860*, 2019.

[20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[21] 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, 2015.

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

[23] 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.

[24] W. M. Haddad, S. G. Nersesov, and L. Du, "Finite-time stability for time-varying nonlinear dynamical systems," in *2008 American control conference*. IEEE, 2008, pp. 4135–4139.

[25] T. Babiak, F. Blahoudek, M. Křetínskỳ, and J. Strejček, "Effective translation of ltl to deterministic rabin automata: Beyond the (F, G)-fragment," in *Automated Technology for Verification and Analysis*. Springer, 2013, pp. 24–39.

[26] 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*. IEEE, 2014, pp. 6271–6278.