Zonotope-based Symbolic Controller Synthesis for Linear Temporal Logic Specifications

Wei Ren, *Member, IEEE*, Raphaël M. Jungers, *Senior Member, IEEE*, and Dimos V. Dimarogonas, *Fellow, IEEE*

***Abstract*—This paper studies the controller synthesis problem for nonlinear control systems under linear temporal logic (LTL) specifications using zonotope techniques. A local-to-global control strategy is proposed for the desired specification expressed as an LTL formula. First, a novel approach is developed to divide the state space into finite zonotopes and constrained zonotopes, which are called *cells* and allowed to intersect with the neighbor cells. Second, from the intersection relation, a graph among all cells is generated to verify the realization of the accepting path for the LTL formula. The realization verification determines if there is a need for the control design, and also results in finite local LTL formulas. Third, once the accepting path is realized, a novel abstraction-based method is derived for the controller design. In particular, we only focus on the cells from the realization verification and approximate each cell thanks to properties of zonotopes. Based on local symbolic models and local LTL formulas, an iterative synthesis algorithm is proposed to design all local abstract controllers, whose existence and combination establish the global controller for the LTL formula. Finally, the proposed framework is illustrated via a path planning problem of mobile robots.**

***Index Terms*—Controller synthesis, nonlinear systems, symbolic control, linear temporal logic, zonotope-based covering.**

I. Introduction

Due to the resemblance to natural language and the existence of off-the-shelf algorithms for model checking [1], linear temporal logic (LTL) is an expressive language to specify high-level tasks for dynamical systems. Because of the complexities of tasks and dynamical systems, control synthesis problems usually cannot be solved directly. For this purpose, many approaches have been proposed, such as sampling/optimization-based approaches [2], game-based approaches [3], [4], and automata/abstraction-based approaches [5]. Among all these approaches, the abstraction-based approach allows us to solve control synthesis problems efficiently via techniques from supervisory control [6] or game theory [3]. Symbolic abstractions are a dynamical systems description where each symbolic state is a collection or aggregate of system states [7]. With an equivalence relation between the dynamical system and its symbolic abstraction, the controller is synthesized to be correct by design [8], and the formal verification may be unnecessary.

In the abstraction-based control approach, the bottom-up strategy is standard and has two steps: the abstraction construction to discretize dynamical systems, and the backward searching to find abstract controllers. Once an equivalence relation between the dynamical system and its abstraction is established, the abstract controller can be refined as a hybrid controller for the dynamical system [7]. Since the backward searching can be done by existing algorithmic tools [3], [6], the key of the bottom-up strategy is the abstraction construction to ensure certain equivalence relations. Many classes of dynamical systems admitting symbolic abstractions have been identified, including nonlinear systems [9]–[11], time-delay systems [12], [13], switched systems [14], networked control systems [15] and stochastic systems [16]. In terms of equivalence relations, (bi-)simulation relation [7], feedback refinement relation [17] and their variants/extensions [9], [18] have been widely studied. Recently many toolboxes [5], [19]–[21] have been developed for high-level control problems.

Despite these works via the abstraction-based control approach, the control synthesis for dynamical systems under LTL specifications is still challenging from the following two perspectives. Regarding dynamical systems and LTL formulas, general LTL formulas or GR(1) formulas [3] are only studied for dynamical systems with simple structures [5], [22], or only simple/specific LTL formulas like safety and reachability specifications are addressed for complex dynamical systems [19]–[21]. A direct question is: *how do we deal with the controller synthesis problem for complex dynamical systems under general LTL formulas*? Regarding the control synthesis, the design strategies are usually derived in a global way that the state space of dynamical systems needs to be considered as a whole [7], [17]. In this respect, although different equivalence relations are proposed, the abstraction construction and the controller synthesis are still time-consuming and computationally complex. Hence, how to address the controller synthesis problem in a local way needs further study.

Motivated by the above discussion, in this paper we propose a zonotope-based top-down approach to deal with the control synthesis problem for dynamical systems under LTL specifications. In particular, we propose a zonotope-based approach to verify the realization of LTL formulas, and then combine the realization verification and the abstraction-based control design to develop a local-to-global control strategy. To this end, our first contribution is the zonotope-based approach for the realization of LTL formulas. Different from many works [23]–[25] where zonotopes are used to represent or estimate reachable sets, we cover the state space via finite zonotopes and constrained zonotopes, which are called *cells*

This work was supported by the Fundamental Research Funds for the Central Universities under Grant DUT22RT(3)090, the National Natural Science Foundation of China under Grant 08120003, the H2020 ERC Consolidator Grants L2C (Grant 864017) and LEAFHOUND (Grant 864720), the CHIST-ERA 2018 project DRUID-NET, the Walloon Region and the Innoviris Foundation, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

W. Ren is with Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian 116024, China. R. M. Jungers is with ICTEAM institute, UCLouvain, 1348 Louvain-la-Neuve, Belgium. D. V. Dimarogonas is with Division of Decision and Control Systems, EECS, KTH Royal Institute of Technology, SE-10044, Stockholm, Sweden. Email: wei.ren@dlut.edu.cn, raphael.jungers@uclouvain.be, dimos@kth.se.

and allowed to intersect with neighbor cells. Based on the intersection relation among all cells, a graph is constructed and further applied to verify the realization of the LTL formula by searching an admissible path to realize the accepting path of the LTL formula. From the graph-based method, the realization verification determines a sequence of finite cells, which not only shows the necessity of the controller design, but also prepares for the controller synthesis afterwards.

Due to the zonotope-based approach and the realization verification, a direct outcome is a formal decomposition of the satisfaction of the global LTL formula into the satisfaction of finite local ones. Hence, the control synthesis problem is decomposed into finite local ones, which can be solved iteratively. Another outcome is a novel and local abstraction construction approach. In particular, each cell is approximated locally via its own properties, and thus the approximations of different cells are not necessarily the same. Furthermore, different local symbolic models are constructed such that different equivalence relations can be satisfied. The proposed approach is local such that different equivalence relations can be satisfied locally, which potentially reduces the conservatism caused by one equivalence relation and the computational complexity from the global approximation.

With the decomposition of the control synthesis problem and the construction of local symbolic models, a local-to-global control strategy is developed to design all local abstract controllers to ensure the satisfaction of the LTL formula, which is the second contribution of this paper. For each cell, its intersection with the previous cell is the initial region, whereas its intersection with the next cell is the target region. Hence, each cell has its local LTL formula, which is combined with the local symbolic model to design the local controller. From the local LTL formula of each cell, the local controller design strategy is implemented repeatedly, which is formulated into an iterative synthesis algorithm. In addition, using the derived equivalence relations, we further discuss the refinement of local abstract controllers and the relation between local and global controllers, which validates the satisfaction of the LTL formula via all designed local abstract controllers.

A preliminary version of this work has been presented in [26], and is expanded significantly here in several directions. First, since we aim to propose a local-to-global control strategy, here the realization verification is modified and the detailed decomposition is established to show how to decompose the LTL formula into finite local ones. Hence, the robust accepting path and its realization verification are presented to ensure the well-decomposedness of the LTL formula. Second, the controller design is only outlined in [26] while presented here in detail. In terms of the abstraction construction, we address both the case that the system is not incrementally stable and the case that the system is incrementally stable, while only the former one is studied in [26]. In terms of the controller design, the relation between the global and local control problems is derived, and an explicit local-to-global control strategy is established. Therefore, different equivalence relations can be mixed up in the current setting, which however is not the case in [26]. Finally, all formal proofs of the main results are included in this paper to substantiate the correctness and feasibility of the proposed control strategy.

The rest of this paper is organized as follows. Preliminaries are stated in Section III and the problem formulation is given in Section IV. Section V presents the covering mechanism, and Section VI shows the realization verification. The controller synthesis is derived in Section VII. Simulation results are given in Section VIII. Section IX concludes with future work.

II. Related Work

The control synthesis problem for LTL specifications has been considered extensively [5], [27]–[31]. In abstraction-based control, the bottom-up and top-down structures are two common control approaches, both of which are built on the partition of the state space. A coarse partition is computed in a top-down strategy to derive a discrete plan for the controller design in [28], [31], whereas a finer partition is used in a bottom-up strategy to construct an appropriate symbolic model such that certain equivalence relation is satisfied [9], [11], [17], [18], [32]. In these works, neighbor cells are adjacent but not overlapped. Hence, in order to deal with the controller synthesis problem, the state-space partition needs to be considered globally [14], [17] or semi-globally [27], [31]. In this respect, general LTL specifications are hard to be addressed efficiently and the computational complexity is still huge. In this work, we propose a zonotope-based covering mechanism for the state space, and allow for the intersection among neighbor cells such that local LTL specifications can be derived, which can further be resolved iteratively and efficiently via existing tools.

In the abstraction construction, the bottom-up strategy requires the symbolic abstraction to be constructed first such that a fully actuated model is derived. In this way, the entire state space is considered integrally and uniformly, while some assumptions like incremental stability properties are needed in [8], [9], [33], [34] for the well-constructedness. However, global and uniform discretization of high-dimensional dynamical systems usually results in complexity explosion. On the other hand, the top-down strategy allows to construct symbolic abstractions at different levels of granularity to deal with the planning and control problem sequentially. For instance, a coarse abstraction is constructed first to derive a high-level plan over regions of interest in the workspace given the temporal task [29], [31], [35], which is then combined with the abstraction refinement [31], [36] to design a low-level feedback controller to execute the derived high-level plan. The abstraction construction or refinement is still (semi-)global, and only one equivalence relation is allowed in these existing works. Due to the zonotope-based covering and the resulting intersection relation, here we propose a novel graph-based way for high-level planning and realization verification, and then construct the symbolic abstraction for each cell individually and locally, which hence allows different equivalence relations to be combined and further facilitates the controller design.

III. Notation and Preliminaries

Let $\mathbb{R} = (-\infty, +\infty)$, $\mathbb{R}^+ = [0, +\infty)$, $\mathbb{N} = \{0, 1, \ldots\}$ and $\mathbb{N}^+ = \{1, 2, \ldots\}$. $\mathbb{R}^n$ is the n-dimensional Euclidean space. For $x \in \mathbb{R}^n$, x_i is the i-th element of x; $\|x\|$ is the infinity norm of x. $\mathbb{B}(x,\varepsilon)=\{y\in\mathbb{R}^n:\|x-y\|\leq\varepsilon\}$ denotes the closed ball centered at $x\in\mathbb{R}^n$ with radius $\varepsilon\in\mathbb{R}^+$. For a finite set $\mathbb{A}\subset\mathbb{R}^n$, $|\mathbb{A}|$ is the cardinality of $\mathbb{A}$. For $\mathbb{A},\mathbb{B}\subset\mathbb{R}^n$, $\mathbb{A}$ is compact if it is closed and bounded; $\mathbb{B}\setminus\mathbb{A}=\{x:x\in\mathbb{B},x\notin\mathbb{A}\}$. For a compact set $\mathbb{A}\subset\mathbb{R}^n$ and $\varepsilon>0$, $\mathbb{A}^\circ$ is the interior of $\mathbb{A}$, and $\partial\mathbb{A}$ is the boundary of $\mathbb{A}$; $\mathbb{A}^\varepsilon=\{x\in\mathbb{A}:\mathbb{B}(x,\varepsilon)\subseteq\mathbb{A}\}$ is the ε-contraction of $\mathbb{A}$; $\mathbf{E}_\varepsilon(\mathbb{A})=\{y\in\mathbb{R}^n:\exists x\in\mathbb{A},\|y-x\|\leq\varepsilon\}$ is the ε-expansion of $\mathbb{A}$. A relation $\mathcal{R}\subseteq\mathbb{A}\times\mathbb{B}$ is identified with the map $\mathcal{R}:\mathbb{A}\to 2^{\mathbb{B}}$ defined by $b\in\mathcal{R}(a)$ if and only if $(a,b)\in\mathcal{R}$. The inverse relation of $\mathcal{R}$ is $\mathcal{R}^{-1}=\{(b,a)\in\mathbb{B}\times\mathbb{A}:(a,b)\in\mathcal{R}\}$. Id is the identity function. A function $\alpha:\mathbb{R}^+\to\mathbb{R}^+$ is of class $\mathcal{K}$ if it is continuous, strictly increasing and $\alpha(0)=0$; α is of class $\mathcal{K}_\infty$ if $\alpha\in\mathcal{K}$ and it is unbounded. A function $\beta:\mathbb{R}^+\times\mathbb{R}^+\to\mathbb{R}^+$ is of class $\mathcal{KL}$ if $\beta(s,t)\in\mathcal{K}$ for each fixed $t\geq 0$ and $\beta(s,t)$ decreases to zero as $t\to\infty$ for each fixed $s\geq 0$.

A set $\mathbf{Z}\subset\mathbb{R}^n$ is a *zonotope*, if there exists $(\mathbf{c},\mathbf{G})\in\mathbb{R}^n\times\mathbb{R}^{n\times n_g}$ such that $\mathbf{Z}=\{\mathbf{c}+\mathbf{G}\xi:\|\xi\|\leq 1\}$, where $\mathbf{c}\in\mathbb{R}^n$ is the center and $\mathbf{G}\in\mathbb{R}^{n\times n_g}$ is the generator matrix with each column as a generator. A set $\mathbf{Z}^{\mathrm{c}}\subset\mathbb{R}^n$ is a *constrained zonotope*, if there exists $(\mathbf{c},\mathbf{G},\mathbf{A},\mathbf{b})\in\mathbb{R}^n\times\mathbb{R}^{n\times n_g}\times\mathbb{R}^{n_c\times n_g}\times\mathbb{R}^{n_c}$ such that $\mathbf{Z}^{\mathrm{c}}=\{\mathbf{c}+\mathbf{G}\xi:\|\xi\|\leq 1,\mathbf{A}\xi=\mathbf{b}\}$, where $\mathbf{A}\xi=\mathbf{b}$ is the constraint condition. We use the notations $\mathbf{Z}=\{\mathbf{c},\mathbf{G}\}$ (generator representation or G-representation) for zonotopes, and $\mathbf{Z}^{\mathrm{c}}=\{\mathbf{c},\mathbf{G},\mathbf{A},\mathbf{b}\}$ (constrained generator representation or CG-representation) for constrained zonotopes. Given $\mathbf{Z}=\{\mathbf{c},\mathbf{G}\}$, if $\mathbf{G}$ is diagonal, orthogonal or invertible, then $\mathbf{Z}$ is reduced to a box, a hypercube or a parallelotope, respectively. From [23], $\mathbf{Z}^{\mathrm{c}}\subset\mathbb{R}^n$ is a constrained zonotope if and only if it is a convex polytope.

Lemma 1 ([23]): For every $\mathbf{Z}^{\mathrm{c}}=\{\mathbf{c},\mathbf{G},\mathbf{A},\mathbf{b}\}\subset\mathbb{R}^n$, $\mathbf{Z}^{\mathrm{c}}\neq\varnothing$ if and only if $\min\{\|\xi\|:\mathbf{A}\xi=\mathbf{b}\}\leq 1$; $z\in\mathbf{Z}^{\mathrm{c}}$ if and only if $\min\{\|\xi\|:\mathbf{G}\xi=z-\mathbf{c},\mathbf{A}\xi=\mathbf{b}\}\leq 1$.

A. Transition Systems

Definition 1 ([32]): A *transition system* is a sextuple $\mathbf{T}=(X,X_0,U,\Delta,Y,H)$, consisting of: (i) a set of states X; (ii) a set of initial states $X^0\subseteq X$; (iii) a set of inputs U; (iv) a transition relation $\Delta\subseteq X\times U\times X$; (v) a set of outputs Y; (vi) an output function $H:X\to Y$. $\mathbf{T}$ is said to be *metric* if the output set Y is equipped with a metric $\mathbf{d}:Y\times Y\to\mathbb{R}^+$, and *symbolic* if the sets X and U are finite or countable.

The transition $(x,u,x')\in\Delta$ is denoted by $x'\in\Delta(x,u)$, which means that the system can evolve from the state x to the state x' under the input u. An input $u\in U$ belongs to *the set of the enabled inputs* at the state x, denoted by $\mathsf{enab}(x)$, if $\Delta(x,u)\neq\varnothing$; see [8]. If $\mathsf{enab}(x)=\varnothing$, then x is said to be *blocking* (or deadlock), otherwise, x is said to be *non-blocking*. The transition system $\mathbf{T}$ is said to be *deterministic*, if for all $x\in X$ and all $u\in\mathsf{enab}(x)$, $\Delta(x,u)$ has exactly one element. In this case, let $x'=\Delta(x,u)$ with a slight abuse of notation.

Definition 2 ([32]): Let $\mathbf{T}_i=(X_i,X_i^0,U_i,\Delta_i,Y,H_i)$, $i=1,2$, be two transition systems with the same output set Y equipped with the metric $\mathbf{d}$. Given a precision $\varepsilon>0$, a relation $\mathcal{R}\subseteq X_1\times X_2$ is called an *ε-approximate bisimulation relation (ε-ABR)* between $\mathbf{T}_1$ and $\mathbf{T}_2$, denoted by $\mathbf{T}_1\simeq_\varepsilon\mathbf{T}_2$, if for all $(x_1,x_2)\in\mathcal{R}$: (i) $\mathbf{d}(H_1(x_1),H_2(x_2))\leq\varepsilon$; (ii) for each $x_1'\in\Delta_1(x_1,u_1)$ with $u_1\in\mathsf{enab}(x_1)$, there exists $x_2'\in\Delta_2(x_2,u_2)$ with $u_2\in\mathsf{enab}(x_2)$ such that $(x_1',x_2')\in\mathcal{R}$; (iii) for each $x_2'\in\Delta_2(x_2,u_2)$ with $u_2\in\mathsf{enab}(x_2)$, there exists $x_1'\in\Delta_1(x_1,u_1)$ with $u_1\in\mathsf{enab}(x_1)$ such that $(x_1',x_2')\in\mathcal{R}$.

Definition 3 ([17]): Let $\mathbf{T}_1$ and $\mathbf{T}_2$ be two transition systems $\mathbf{T}_i=(X_i,X_i^0,U_i,\Delta_i,Y_i,H_i)$ with $i\in\{1,2\}$, and $U_2\subseteq U_1$. A relation $\mathcal{F}\subseteq X_1\times X_2$ is a *feedback refinement relation (FRR)* from $\mathbf{T}_1$ to $\mathbf{T}_2$, denoted by $\mathbf{T}_1\preceq_{\mathcal{F}}\mathbf{T}_2$, if for all $(x_1,x_2)\in\mathcal{F}$: $U_2(x_2)\subseteq U_1(x_1)$; $\mathcal{F}(\Delta_1(x_1,u))\subseteq\Delta_2(x_2,u)$ for all $u\in U_2(x_2)$, where $U_i(x):=\{u\in U_i:u\in\mathsf{enab}(x)\}$.

Definition 4 ([8]): A *controller* for a transition system $\mathbf{T}=(X,X_0,U,\Delta,Y,H)$ is a map $\mathcal{C}:X\to 2^U$ with the domain defined as $\{x\in X:\mathcal{C}(x)\neq\varnothing\}$. It is *well-defined*, if $\mathcal{C}(x)\subseteq\mathsf{enab}(x)$ for all $x\in X$. The *controlled transition system* is denoted by the transition system $\mathbf{T}^{\mathrm{c}}=(X,X_0,U,\Delta_{\mathrm{c}},Y,H)$, where $x'\in\Delta_{\mathrm{c}}(x,u)$ if and only if $u\in\mathcal{C}(x)$ and $x'\in\Delta(x,u)$.

B. Linear Temporal Logic (LTL)

Let $\mathcal{AP}$ be a set of atomic propositions [1]. Based on atomic propositions (state labels $\alpha\in\mathcal{AP}$), Boolean connectors like negation $\neg$ and conjunction $\wedge$, and two temporal operators $\bigcirc$ ('next') and U ('until'), LTL is formed via the syntax below:

$$\varphi ::= \mathsf{true}\mid\alpha\mid\neg\varphi\mid\varphi_1\wedge\varphi_2\mid\bigcirc\varphi\mid\varphi_1\mathsf{U}\varphi_2, \tag{1}$$

where $\varphi,\varphi_1,\varphi_2$ are LTL formulas. The disjunction $\vee$, and temporal operators $\Diamond$ ('eventually') and $\Box$ ('always') can be derived as $\varphi_1\vee\varphi_2:=\neg(\neg\varphi_1\wedge\neg\varphi_2)$, $\Diamond\varphi:=\mathsf{true}\mathsf{U}\varphi$ and $\Box\varphi:=\neg\Diamond\neg\varphi$. Let $2^{\mathcal{AP}}$ be the power set of $\mathcal{AP}$.

Definition 5 ([4, Chapter 2.2.1]): A *Kripke structure* is of the form $(X,X_0,\Delta,\mathcal{L})$ with a state set X, an initial state set $X_0\subseteq X$, a transition relation $\Delta\subseteq X\times X$, and a labeling function $\mathcal{L}:X\to 2^{\mathcal{AP}}$.

An infinite *word* over $2^{\mathcal{AP}}$ is an infinite sequence $\sigma:=\sigma_0\sigma_1\ldots\in(2^{\mathcal{AP}})^\omega$, where ω denotes infinite repetition and $\sigma_k\in 2^{\mathcal{AP}}$, $k\in\mathbb{N}$. For each $i\in\mathbb{N}$, let $\sigma|_i:=\sigma_i\sigma_{i+1}\ldots$. The semantics of LTL is defined as follows.

- $\sigma\models\alpha$ if $\alpha\in\sigma_1$;
- $\sigma\models\neg\varphi$ if $\sigma\not\models\varphi$;
- $\sigma\models\varphi_1\wedge\varphi_2$ if $\sigma\models\varphi_1$ and $\sigma\models\varphi_2$;
- $\sigma\models\bigcirc\varphi$ if $\sigma|_2\models\varphi$;
- $\sigma\models\varphi_1\mathsf{U}\varphi_2$ if there exists $k\geq 0$ such that $\sigma|_i\models\varphi_1$ for all $0\leq i<k$ and $\sigma|_k\models\varphi_2$.

The union of infinite words satisfying φ is denoted as $\mathsf{Words}(\varphi)=\{\sigma\in(2^{\mathcal{AP}})^\omega:\sigma\models\varphi\}$. From [1, Theorem 5.41], any LTL formula φ can be translated into a nondeterministic Büchi automaton B over $2^{\mathcal{AP}}$, which is defined below.

Definition 6: A *nondeterministic Büchi automaton (NBA)* B over $2^{\mathcal{AP}}$ is a tuple $\mathsf{B}=(\mathcal{Q}_{\mathsf{B}},\mathcal{Q}_{\mathsf{B}}^0,2^{\mathcal{AP}},\Delta_{\mathsf{B}},\mathcal{F}_{\mathsf{B}})$, where $\mathcal{Q}_{\mathsf{B}}$ is a finite set of states, $\mathcal{Q}_{\mathsf{B}}^0\subseteq\mathcal{Q}_{\mathsf{B}}$ is the set of initial states, $2^{\mathcal{AP}}$ is the set of input alphabets, $\Delta_{\mathsf{B}}\subseteq\mathcal{Q}_{\mathsf{B}}\times 2^{\mathcal{AP}}\times 2^{\mathcal{Q}_{\mathsf{B}}}$ is a transition relation, and $\mathcal{F}_{\mathsf{B}}\subseteq\mathcal{Q}_{\mathsf{B}}$ is a set of accepting states.

An infinite run q of B over an infinite word $\sigma=\sigma_0\sigma_1\ldots\in(2^{\mathcal{AP}})^\omega$ is a sequence $\mathsf{q}=\mathsf{q}_0\mathsf{q}_1\ldots$ such that $\mathsf{q}_0\in\mathcal{Q}_{\mathsf{B}}^0$ and $(\mathsf{q}_k,\sigma_k,\mathsf{q}_{k+1})\in\Delta_{\mathsf{B}}$, $k\in\mathbb{N}$. An infinite run q is called *accepting*, if $\mathsf{Inf}(\mathsf{q})\cap\mathcal{F}_{\mathsf{B}}\neq\varnothing$, where $\mathsf{Inf}(\mathsf{q})$ is the set of states that appear in q infinitely often. The set of infinite words over $2^{\mathcal{AP}}$ resulting in an accepting run of B is called the accepted language of B and is denoted as $\mathfrak{L}(\mathsf{B})$. From [1, Theorem 5.41], the NBA B associated with an LTL formula φ does always exists such that B accepts all and only those infinite runs over $\mathcal{AP}$ that satisfy φ, i.e., $\mathfrak{L}(\mathsf{B}) = \mathsf{Words}(\varphi)$.

C. Nonlinear Control Systems

Consider the nonlinear control system

$$\Sigma : \quad \dot{x}(t) = f(x(t), u(t)), \tag{2}$$

where $x(t) \in \mathbb{X}$ is the system state, and $u(t) \in \mathbb{U}$ is the control input. $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ are assumed to be compact. The initial state set is denoted as $\mathbb{X}_0 \subset \mathbb{X}$, and is assumed to be a constrained zonotope. The controller is assumed to be piecewise continuous, and the function $f : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$ is assumed to be continuous and satisfy the local Lipschitz assumption for all $u \in \mathbb{U}$. For (2), a curve $\mathbf{x} : \mathbb{R}^+ \to \mathbb{X}$ is called a *trajectory*, if there exists a control input $u(t) \in \mathbb{U}$ such that $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), u(t))$ for all $t \in \mathbb{R}^+$. We denote by $\mathbf{x}(t, x, u)$ the state reached at the time $t \in \mathbb{R}^+$ under the input $u \in \mathbb{U}$ from the initial state $x \in \mathbb{X}$.

Using the sampling technique, we derive the time discretization of (2). Let $\tau > 0$ be the sampling period to be designed. The sampled-data system of (2) is denoted as a transition system $\mathbf{T}_\tau(\Sigma) := (X_1, X_1^0, U_1, \Delta_1, Y_1, H_1)$, where, $X_1 = \mathbb{X}$ is the state set; $X_1^0 = \mathbb{X}_0$ is the set of initial states; $U_1 = \{u \in \mathbb{U} : \mathbf{x}(\tau, x, u) \text{ is defined for all } x \in \mathbb{X}\}$ is the input set; Δ_1 is the transition relation: for $x \in X_1$ and $u \in U_1$, $x' = \Delta_1(x, u)$ if and only if $x' = \mathbf{x}(\tau, x, u)$; $Y_1 = X_1$ is the output set; and $H_1 = \mathrm{Id}$ is the output map. The system $\mathbf{T}_\tau(\Sigma)$ is non-blocking and deterministic. $\mathbf{T}_\tau(\Sigma)$ is metric if the output set Y_1 is equipped with the metric $\mathbf{d}(y, y') = \|y - y'\|$ for all $y, y' \in Y_1$. If the state space is limited to a subset $\mathbb{X} \subseteq \mathbb{X}$, then we use the notation $\mathbf{T}_\tau(\Sigma, \mathbb{X})$ to emphasize the local state space $\mathbb{X} \subseteq \mathbb{X}$.

IV. Problem Formulation

In the paper, we consider the system (2) and aim to design a controller such that the desired specification described by an LTL formula φ can be satisfied. For this purpose, we start with the high-level plan to satisfy the LTL formula φ.

In the state space $\mathbb{X} \subset \mathbb{R}^n$, the set of all the obstacles is denoted by $\mathbb{O} := \cup_{l \in \mathbb{K}} \mathbb{O}_l \subset \mathbb{X}$ with a finite index set $\mathbb{K} \subset \mathbb{N}^+$. We consider the existence of the set $\Pi \subseteq \mathcal{AP}$ of propositions for (2); see [29], [31], [35]. Each $\pi_i \in \Pi$ is associated with a subset $\mathbb{X}_i \subseteq \mathbb{X} \setminus \mathbb{O}$ such that $\pi_i = \mathtt{true}$ if $x \in \mathbb{X}_i$, where $i \in \{0, 1, \ldots, |\Pi|\}$ and $|\Pi|$ is finite. In particular, $\mathbb{X}_0$ is the initial state set. In order to show the relation between $\pi_i \in \Pi$ and $\mathbb{X}_i \subseteq \mathbb{X} \setminus \mathbb{O}$, we denote $\mathbb{R}(\pi_i) := \mathbb{X}_i$. All these subsets are called the *regions of interest*. Let each $\mathbb{X}_i$ be a constrained zonotope, which is a reasonable since any convex polytope can be represented as a constrained zonotope [23].

Based on the above, we can define a Kripke structure $\mathsf{S} := (\Pi, \Pi_0, \Delta_\mathsf{S}, \mathcal{L})$, where Π is defined above, $\Pi_0 \subseteq \Pi$, $\Delta_\mathsf{S} \subseteq \Pi \times \Pi$ is the transition relation and $\mathcal{L} : \Pi \to 2^{\mathcal{AP}}$ is the labeling function. We define an infinite *path* of S as an infinite sequence of states $\pi = \pi_0\pi_1 \ldots$ such that $(\pi_k, \pi_{k+1}) \in \Delta_\mathsf{S}$ for all $k \in \mathbb{N}$. A state sequence

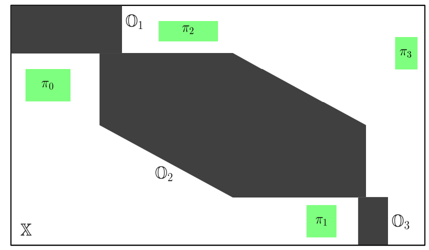

Fig. 1. The state space of the robot in Example 1. The four regions of interest are denoted as $\mathbb{R}(\pi_0), \mathbb{R}(\pi_1), \mathbb{R}(\pi_2), \mathbb{R}(\pi_3)$ with the propositions $\pi_0, \pi_1, \pi_2, \pi_3$, and the three obstacles are $\mathbb{O}_1, \mathbb{O}_2, \mathbb{O}_3$.

results in an infinite word over $2^{\mathcal{AP}}$, which is defined as $\mathsf{Word}(\pi) := \mathcal{L}(\pi_0)\mathcal{L}(\pi_1)\ldots$. Consider the LTL formula φ and a path π, we have $\mathsf{Word}(\pi) \models \varphi$ if $\mathsf{Word}(\pi) \in \mathsf{Words}(\varphi)$. Given the Kripke structure $\mathsf{S} = (\Pi, \Pi_0, \Delta_\mathsf{S}, \mathcal{L})$ and the NBA $\mathsf{B} = (\mathcal{Q}_\mathsf{B}, \mathcal{Q}_\mathsf{B}^0, 2^{\mathcal{AP}}, \Delta_\mathsf{B}, \mathcal{F}_\mathsf{B})$, the product Büchi automaton is defined as $\mathsf{P} := \mathsf{S} \times \mathsf{B} = (\mathcal{Q}_\mathsf{P}, \mathcal{Q}_\mathsf{P}^0, 2^{\mathcal{AP}}, \Delta_\mathsf{P}, \mathcal{F}_\mathsf{P})$, where $\mathcal{Q}_\mathsf{P} = \Pi \times \mathcal{Q}_\mathsf{B}$, $\mathcal{Q}_\mathsf{P}^0 = \Pi_0 \times \mathcal{Q}_\mathsf{B}^0$, $\mathcal{F}_\mathsf{P} = \Pi \times \mathcal{F}_\mathsf{B}$, and $\Delta_\mathsf{P} \subseteq \mathcal{Q}_\mathsf{P} \times \mathcal{Q}_\mathsf{P}$ defined as $((\pi_\mathsf{a}, \mathsf{q}_\mathsf{a}), (\pi_\mathsf{b}, \mathsf{q}_\mathsf{b})) \in \Delta_\mathsf{P}$ if and only if $\pi_\mathsf{b} = \Delta_\mathsf{S}(\pi_\mathsf{a})$ and $\mathsf{q}_\mathsf{b} = \Delta_\mathsf{B}(\mathsf{q}_\mathsf{a})$. Following [37, Algorithm 1] and [35, Algorithm 3], we can find a run of P defined as $\mathsf{p} := \mathsf{p}_0\mathsf{p}_1 \ldots = (\pi_0, \mathsf{q}_0)(\pi_1, \mathsf{q}_1) \ldots$ with $\pi = \pi_0\pi_1 \ldots$ such that $\mathsf{Word}(\pi) \models \varphi$. In this case, the derived path $\pi = \pi_0\pi_1 \ldots$ is called an *accepting path*. From any accepting path π, the path-based region set is defined as

$$\mathbb{R}(\pi) := \{\mathbb{R}(\pi_i) \subseteq \mathbb{X} \setminus \mathbb{O} : \pi_i \in \pi, i \in \mathbb{N}\}. \tag{3}$$

Example 1: Consider a robot which moves in a given state space $\mathbb{X} \subset \mathbb{R}^2$ and is initially placed in a certain region $\mathbb{X}_0 \subset \mathbb{X}$. The robot is to accomplish the task: "patrol between Regions A and B while visiting Region C not before Region B has been visited at least once", which can be expressed as an LTL formula $\varphi = \Box\Diamond\pi_1 \wedge \Box\Diamond\pi_2 \wedge \Diamond\pi_3 \wedge \neg\pi_3 \mathsf{U} \pi_2$. Let $\mathcal{AP} = \{\pi_0, \pi_1, \pi_2, \pi_3\}$. The regions of interest are $\mathbb{R}(\pi_0) = \mathbb{X}_0$ and $\mathbb{R}(\pi_1), \mathbb{R}(\pi_2), \mathbb{R}(\pi_3)$, which are respectively Regions A, B and C; see Fig. 1. We can derive an NBA $\mathsf{B} = (\mathcal{Q}_\mathsf{B}, \mathcal{Q}_\mathsf{B}^0, 2^{\mathcal{AP}}, \Delta_\mathsf{B}, \mathcal{F}_\mathsf{B})$ to be associated with φ. On the other hand, we can define a Kripke structure $\mathsf{S} = (\Pi, \Pi_0, \Delta_\mathsf{S}, \mathcal{L})$ by letting $\Pi = \mathcal{AP}$. We can derive the high-level plan via [35, Algorithm 3], and obtain an accepting path $\pi = \pi_0\pi_1\pi_2\pi_3(\pi_1\pi_2)^\omega$. This accepting path provides a guide for the controller synthesis afterwards. That is, from the accepting path π, we only need to deal with the following five pairs: $(\pi_0, \pi_1), (\pi_1, \pi_2), (\pi_2, \pi_3), (\pi_3, \pi_1)$, and (π_2, π_1). $\lhd$

Next, we need to confirm the realization of the derived accepting path and further design the controller such that the LTL specification is satisfied for the considered system. To this end, the following definition is introduced.

Definition 7: Consider the state space $\mathbb{X} \subset \mathbb{R}^n$, the obstacle set $\mathbb{O} \subseteq \mathbb{X}$, and the LTL formula φ with its accepting path π. If there exists a connected set $\mathbb{X} \subseteq \mathbb{X} \setminus \mathbb{O}$ such that $\mathbb{X} \cap \mathbb{R}(\pi_i) \neq \varnothing$ for all $\pi_i \in \pi$, then π is said to be *realized* in $\mathbb{X}$. The set $\mathbb{X}$ is called a *realization region* of π.

From Definition 7, the realization of π shows that there exists a subset in $\mathbb{X}$ such that the system (2) is not thwarted to achieve the LTL specification φ. Note that the realization of the accepting path is necessary. If the accepting path is not

Algorithm 1: Cover of the State Space

Input: $\mathbb{X} \subset \mathbb{R}^n, \varepsilon > 0$
Output: the set $\mathbb{Z}$ of zonotopes and constrained zonotopes

1 Generate finite zonotopes $\mathbf{Z}_i$ and constrained zonotopes $\mathbf{Z}^c_j$ to cover $\mathbb{X}$
2 Expand both $\mathbf{Z}_i$ and $\mathbf{Z}^c_j$ via the parameter $\epsilon > 0$ as in (4)
3 **return** $\mathbb{Z} = (\cup_{i=1}^N \mathbf{E}_\epsilon(\mathbf{Z}_i)) \cup (\cup_{j=1}^M \mathbf{E}_\epsilon(\mathbf{Z}^c_j))$

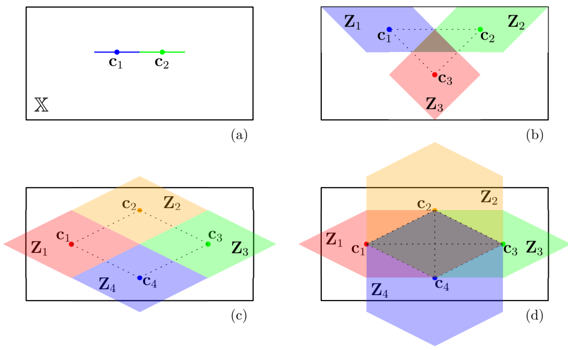

Fig. 2. Illustration of the generation of zonotopes in different cases. (a) $N = 2 < 3$: the generated zonotopes are two segments and thus not well-constructed. (b) $N = 3$: three zonotopes are generated and overlapped. (c) $N = 4$ and each center connects with 2 neighbor centers: four zonotopes are generated but not overlapped. (d) $N = 4$ and each center connects with 3 neighbor centers: four zonotopes are generated and overlapped.

realized, then it means that the LTL specification cannot be satisfied by any controller, which needs to be ruled out; see Fig. 1. Since the realization of the accepting path is verified first, the need of the controller design can be checked and the cost for the controller design can be reduced. For instance, the accepting path in Example 1 is not realized in $\mathbb{X}$, since there does not exist a connect subset $\mathbb{X} \subseteq \mathbb{X} \setminus \mathbb{O}$ such that $\mathbb{X} \cap \mathbb{R}(\pi_1) \neq \varnothing$ and $\mathbb{X} \cap \mathbb{R}(\pi_2) \neq \varnothing$ hold simultaneously.

Based on the above discussion, the problems to be studied in this paper are formulated as follows.

Problem 1. Given the system (2) and an LTL formula φ with its accepting path π, verify whether π is realized.

Problem 2. If π is realized, then design a controller such that the LTL formula φ is satisfied for the system (2).

V. Zonotope-Based Covering Approach

To deal with **Problems 1** and **2**, we first implement zonotope techniques to propose a novel cover of the state space in this section. To be specific, we summarize the covering strategy in Section V-A, then show the detailed generation of (constrained) zonotopes in Section V-B, and finally derive a graph from the proposed generation mechanism in Section V-C.

A. Summary of Covering Strategy

The covering strategy is presented in Algorithm 1 and has two steps. The first step is to generate finite zonotopes and constrained zonotopes to cover the state space $\mathbb{X} \subset \mathbb{R}^n$ (i.e., line 1 in Algorithm 1), and the detailed generation rules will be presented in the next subsection. Since all generated zonotopes and constrained zonotopes are not necessarily overlapped (see Fig. 2(c)), the second step is to implement the expansion operator to expand all generated zonotopes and constrained zonotopes (i.e., line 2 in Algorithm 1). Given any zonotope $\mathbf{Z} = \{\mathbf{c}, \mathbf{G}\}$ and any constrained zonotope $\mathbf{Z}^c = \{\mathbf{c}, \mathbf{G}, \mathbf{A}, \mathbf{b}\}$,

$$\begin{aligned} \mathbf{E}_\epsilon(\mathbf{Z}) &= \{\mathbf{c}, (1+\epsilon)\mathbf{G}\}, \\ \mathbf{E}_\epsilon(\mathbf{Z}^c) &= \{\mathbf{c}, (1+\epsilon)\mathbf{G}, \mathbf{A}, (1+\epsilon)\mathbf{b}\}, \end{aligned} \tag{4}$$

which are respectively treated as the ϵ-expansions of $\mathbf{Z}$ and $\mathbf{Z}^c$ with a slight abuse of notation. These expansion operations have explicit forms and ensures each (constrained) zonotope to overlap with its neighbor zonotopes and constrained zonotopes. Hence, Algorithm 1 produces the union of finite zonotopes and constrained zonotopes as in line 3. In the following, the generated zonotopes and constrained zonotopes are called the *cells* if no confusion arises.

Different from existing partition approaches [27], [29], the proposed approach allows all cells to intersect with their neighbor cells, and we emphasize here that the intersection relation will play an important role in the controller synthesis afterwards. In particular, due to the intersection relation, the proposed approach results in an undirected graph, which will be applied to verify the desired specification in Section VI, and the global specification can be decomposed into finite local ones such that local controllers can be designed individually to ensure the global specification in Section VII.

Algorithm 2: Zonotope Generation

Input: $\mathbb{X} \subset \mathbb{R}^n$
Output: finite zonotopes and constrained zonotopes

Choose $\{\mathbf{c}_i \in \mathbb{X} : i = 1, \ldots, N\}$ with $N > n$
Connect these points such that for each $\mathbf{c}_i \in \mathbb{R}^n$, there exists a full-rank matrix

$$\mathbf{G}_i = (\mathbf{c}_{i_1} - \mathbf{c}_i, \ldots, \mathbf{c}_{i_k} - \mathbf{c}_i), \quad i_k \geq n \tag{5}$$

where $i_l \in \{1, \ldots, N\} \setminus \{i\}$ and $l, k \in \mathbb{N}$
With the center $\mathbf{c}_i \in \mathbb{R}^n$ and the generator matrix $0.5\mathbf{G}_i$, construct the zonotope

$$\mathbf{Z}_i = \{\mathbf{c}_i + 0.5\mathbf{G}_i\xi : \|\xi\| \leq 1\} \tag{6}$$

if $\mathbb{X} \setminus (\cup_{i=1}^N \mathbf{Z}_i) = \varnothing$ **then**
  No need to construct constrained zonotopes
else
  Determine the intersection among $\partial\mathbf{Z}_i, \ldots, \partial\mathbf{Z}_N$ and $\partial\mathbb{X}$

$$\mathbb{H}_1 = \{x \in \mathbb{X} : x \in \partial\mathbf{Z}_i \cap \partial\mathbf{Z}_j \text{ or } x \in \partial\mathbf{Z}_i \cap \partial\mathbb{X}\} \tag{7}$$

  where $i, j \in \{1, \ldots, N\}$ and $i \neq j$
  Refine the set $\mathbb{H}_1$ into the following set

$$\mathbb{H} = \{x \in \mathbb{H}_1 : x \notin \mathbf{Z}_i^\circ\} \tag{8}$$

  Connect all elements in $\mathbb{H}$ to generate a finite set $\mathbb{S}$ of convex regions such that

$$\mathbb{X} \setminus (\cup_{i=1}^N \mathbf{Z}_i) \subseteq \mathbb{S} \text{ and } \mathbb{S} \cap (\cup_{i=1}^N \mathbf{Z}_i^\circ) = \varnothing \tag{9}$$

  Construct $M = |\mathbb{S}|$ constrained zonotopes $\{\mathbf{Z}^c_j : j = 1, \ldots, M\}$ such that

$$\cup_{j=1}^M \mathbf{Z}^c_j \supseteq \mathbb{S} \tag{10}$$

return $(\cup_{i=1}^N \mathbf{Z}_i) \cup (\cup_{j=1}^M \mathbf{Z}^c_j)$

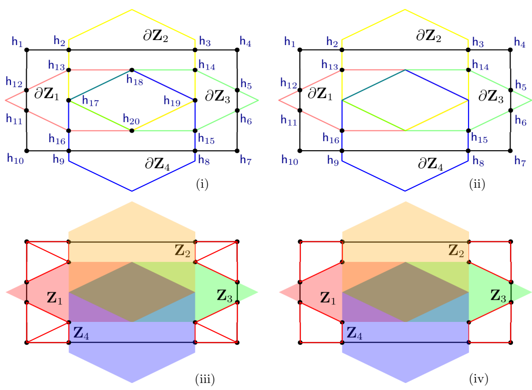

Fig. 3. Illustration of the generation of constrained zonotopes for the case in Fig. 2(d). (i) Illustration of line 7 in Algorithm 2. The boundaries of four zonotopes are lines in different colors, and $\mathbb{H}_1 = \{\mathsf{h}_1, \ldots, \mathsf{h}_{20}\}$. (ii) Illustration of line 8, and $\mathbb{H} = \{\mathsf{h}_1, \ldots, \mathsf{h}_{16}\}$. (iii)-(iv) Illustration of different ways to lines 9-10. In (iii), the set $\mathbb{S}$ has 8 triangle regions, which are used to construct $M = 8$ constrained zonotopes. In (iv), the set $\mathbb{S}$ has 4 regions, which are used to construct $M = 4$ constrained zonotopes.

B. Generation of Zonotopes and Constrained Zonotopes

In this subsection, we show how to generate zonotopes and constrained zonotopes, presented in Algorithm 2. The generation rule consists of two parts: the first part is to generate zonotopes (lines 1-5), and the second part is to generate constrained zonotopes (lines 6-10).

To begin with, the number of zonotopes to be generated is set *a priori* as $N \in \mathbb{N}$, and we choose N points $\mathbf{c}_i \in \mathbb{X}$ as the centers of zonotopes, where $i \in \{1, \ldots, N\}$. Here, we assume $N > n$, where $n \in \mathbb{N}$ is the state-space dimension. Since only the number N is constrained, there is no constraint on how to choose these points, which thus can be chosen arbitrarily. Second, we connect these centers such that each center is connected with at least $n \in \mathbb{N}$ neighbor centers. Therefore, for each center, these connections leads to at least $n \in \mathbb{N}$ vectors, which will be further used as the generators for each zonotope. Hence, the constraints on the number $N \in \mathbb{N}$ and the matrix $\mathbf{G}_i$ are to ensure the construction of zonotopes, which is illustrated in Fig. 2. Finally, with these centers and generators, we generate a zonotope $\mathbf{Z}_i$ as in line 3 of Algorithm 2. In (6), the coefficient 0.5 is to guarantee each generated zonotope to be as large as possible and the intersection region of neighbor zonotopes to be as small as possible.

The generation rule of zonotopes is illustrated in Fig. 2. Given a 2-dimensional space $\mathbb{X} \subset \mathbb{R}^2$, if $N = 2$ and we choose two centers $\mathbf{c}_1, \mathbf{c}_2$ as in Fig. 2(a), then the generated zonotopes are two one-dimensional segments, which cannot be used in the analysis afterwards, implying that the zonotopes are not well-constructed. If the matrix $\mathbf{G}_i$ is not full-rank, then similar cases may occur, where the zonotopes are not well-constructed. Therefore, the constraints on $N \in \mathbb{N}$ and the matrix $\mathbf{G}_i$ are necessary for the well-constructedness of all generated zonotopes. In addition, different choices of $N \in \mathbb{N}$ and $i_k \in \mathbb{N}$ in (5) have effects on the generated zonotopes and result in different zonotopes; see Figs. 2(b)-(d).

If the union of all generated zonotopes covers the state space, then there is no need for the generation of constrained zonotopes (lines 4-5); otherwise, the generation of constrained zonotopes is needed as in lines 7-10. The motivation of using constrained zonotopes lies in that constrained zonotopes are asymmetric [23], [38] and can thus be used to cover asymmetric regions. The generation rule of constrained zonotopes is presented below in detail. In line 7, we determine the intersection among boundaries of all generated zonotopes and the state space $\mathbb{X}$ to derive the set $\mathbb{H}_1$. In line 8, the set $\mathbb{H}_1$ is refined into the set $\mathbb{H}$ by ruling out the elements which are in both $\mathbb{H}_1$ and $\mathbf{Z}_i^\circ$ with certain $i \in \{1, \ldots, N\}$. In line 9, all elements are connected to generate finite convex regions to partition $\mathbb{X} \setminus (\cup_{i=1}^N \mathbf{Z}_i)$. In line 10, these convex regions are represented as constrained zonotopes. See Fig. 3 for the illustration of the generation of constrained zonotopes.

The generation of constrained zonotopes is based on basic operations of polytopes. In line 7, the vertices of all zonotopes are obtained via the transformation of zonotopes from the G-representation into the V-representation [39, Algorithm 2]. The V-representation determines the boundaries of all zonotopes and the set $\mathbb{H}_1$ is derived by verifying the intersection relation of all boundaries. In line 8, the refinement of $\mathbb{H}_1$ is based on Lemma 1. Based on the vertices in $\mathbb{H}$ and using the transformation from V-representation into Z-representation [39, Algorithm 1], constrained zonotopes are generated in lines 9-10 to partition the region $\mathbb{X} \setminus (\cup_{i=1}^N \mathbf{Z}_i)$. Note that other transformation approaches (see e.g., [23]) can be applied to show the reasonability of the generation mechanism.

The overall generation rule is summarized in Algorithm 2, which produces finite zonotopes and constrained zonotopes (line 11) to cover the state space. After the expansion operation (4), Algorithm 1 produces the set $\mathbb{Z}$. To simplify the following notations, all cells in $\mathbb{Z}$ are labeled via a finite symbol set $\mathbb{V} := \{\mathsf{v}_1, \ldots, \mathsf{v}_{N+M}\}$, where $\mathbb{R}(\mathsf{v}_i) = \mathbf{E}_\epsilon(\mathbf{Z}_i)$ with $i \in \{1, \ldots, N\}$ and $\mathbb{R}(\mathsf{v}_{N+j}) = \mathbf{E}_\epsilon(\mathbf{Z}_j^c)$ with $j \in \{1, \ldots, M\}$. Let $\mathcal{N} := \{1, \ldots, N+M\}$, and the cover of $\mathbb{X}$ is $\mathcal{P}(\mathbb{X}) = \mathbb{Z}$ with

$$\mathbb{Z} := \{\mathbb{Z}_k = \mathbb{R}(\mathsf{v}_k) : k \in \mathcal{N}\}. \tag{11}$$

Remark 1: In Algorithm 2, the number N and the matrix $\mathbf{G}_i$ are not fixed, which indicates the flexibility of the proposed covering strategy. Different choices of N and $\mathbf{G}_i$ have no effects on the feasibility of Algorithm 2. That is, for fixed N and $\mathbf{G}_i$, Algorithm 2 produces finite zonotopes and constrained zonotopes to cover the state space. However, the choices of N and $\mathbf{G}_i$ affect the generation of zonotopes (see Fig. 2), and further affect the computational complexity of Algorithm 2. In particular, lines 7-9 involve basic operations of polytopes, which can be achieved efficiently with tools from [39]. If N and $\mathbf{G}_i$ are such that the constrained zonotopes are not needed (i.e., line 5 in Algorithm 2), then the computational complexity of Algorithm 2 can be reduced. $\triangle$

Remark 2: As mentioned in Section V-A, each cell intersects with its neighbor cells. From Figs. 1(b)-1(d), the intersection comes partially and naturally from the generation rule, which is further enhanced via line 2 in Algorithm 1. Although other classes of polytopes [27], [30] have been applied to cover the state space, the properties of zonotopes and constrained zonotopes will be involved in the controller design afterwards, which is not the case in [27], [30]. $\triangle$

Algorithm 3: Adjacency Matrix

Input: $\mathbb{Z}$, $\mathbb{O} \subset \mathbb{X}$, $\varepsilon \in (0, \epsilon]$
Output: $A = [a_{ij}]$, $\mathbb{I} = \{\mathbb{I}_{ij}\}$
1 **for** $i = 1 : 1 : N + M - 1$ **do**
2   **if** $\mathbb{Z}_i^{\varepsilon} \cap \mathbf{O} \neq \mathbb{Z}_i^{\varepsilon}$ **then**
3   **for** $j = 1 : 1 : N + M$ **do**
4   **if** $\mathbb{Z}_j^{\varepsilon} \cap \mathbf{O} \neq \mathbb{Z}_j^{\varepsilon}$ **then**
5         $\Omega = \mathbb{Z}_i^{\varepsilon} \cap \mathbb{Z}_j^{\varepsilon}$
6   **if** $\Omega = \varnothing$ **then**
7           $a_{ij} = 0$
8   **else**
9   **if** $(\mathbb{Z}_i^{\varepsilon} \cup \mathbb{Z}_j^{\varepsilon}) \setminus (\Omega \cap \mathbf{O})$ *is connected* **then**
10             $a_{ij} = 1$
11             $\mathbb{I}_{ij} = \Omega \setminus \mathbf{O}$
12   **else**
13             $a_{ij} = 0$
14   **else**
15         $a_{ij} = 0$
16   **else**
17     $a_{ij} = 0$ for $j \in \mathcal{N}$
18 **return** $A = [a_{ij}]$, $\mathbb{I} = \{\mathbb{I}_{ij}\}$

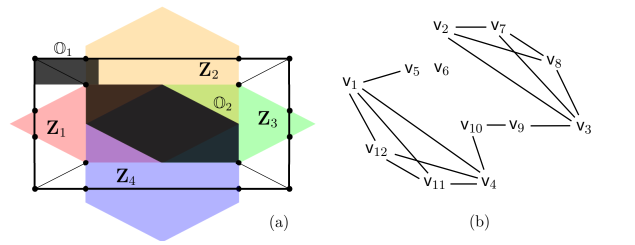

Fig. 4. Illustration of Algorithm 3 via Fig. 3(iii). (a) The cover of the state space $\mathbb{X} \subset \mathbb{R}^2$, and $\mathbb{O}_3$ in Fig. 1 is removed here. (b) The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{\mathsf{v}_1, \ldots, \mathsf{v}_{12}\}$ and $\mathcal{E}$ from the adjacency matrix A.

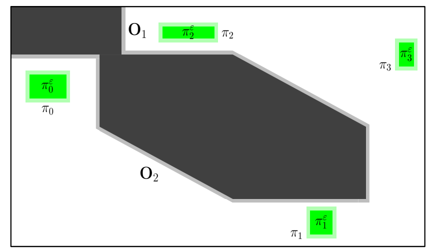

Fig. 5. Illustration of the state space in Fig. 4(a). For each $i \in \{0, 1, 2, 3\}$, the light green region is included in $\mathbb{R}(\pi_i)$ while not in $\mathbb{R}(\pi_i^{\varepsilon})$. For each $j \in \{1, 2\}$, $\mathbf{O}_j = \mathbf{E}_{\varepsilon}(\mathbb{O}_j)$ includes both dark and light grey regions.

C. Topological Graph

In this subsection, we follow the intersection relation among all cells to derive an undirected graph. For this purpose, some auxiliary notations are introduced. Given two regions $\mathbb{A}, \mathbb{B} \subset \mathbb{X}$, their intersection $\mathbb{A} \cap \mathbb{B}$ is *admissible*, if $\mathbb{A} \cup \mathbb{B} \setminus (\mathbb{A} \cap \mathbb{B} \cap \mathbb{O})$ is connected. Given an $\varepsilon \in (0, \epsilon]$, we denote $\mathbf{O} := \mathbf{E}_{\varepsilon}(\mathbb{O})$.

To derive the graph, the key is how to determine the existence of an edge between two neighbor cells, which can be achieved by verifying whether all intersection regions are admissible. For instance, if the intersection region between two neighbor cells belongs to the obstacle set, then this intersection region is not admissible and no edge is defined between these two neighbor cells. By verifying all intersection regions, we have all admissible intersection regions and the adjacency matrix, which is summarized in Algorithm 3. In Algorithm 3, the contraction and expansion operations in lines 2-17 are implemented to facilitate the controller design afterwards. These contraction and expansion operations can be removed if we only focus on the realization of the accepting path.

Lemma 2: Given the cover $\mathbb{Z}$ in (11) of the state space $\mathbb{X} \subseteq \mathbb{R}^n$, the adjacency matrix A from Algorithm 3 shows all edges among all cell pairs.

Lemma 2 is derived directly from Algorithm 3. Due to the finite number of all cells, the termination time of Algorithm 3 is finite. From the graph theory, the matrix A is further transformed into an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set is $\mathcal{V} = \mathbb{V}$ and the edge set is $\mathcal{E} \subseteq \mathbb{V} \times \mathbb{V}$ with $(\mathsf{v}_i, \mathsf{v}_j) \in \mathcal{E}$ if $a_{ij} = 1$. See Fig. 4 for the illustration of Algorithm 3 and the generated graph.

VI. Realization Verification

Based on the cover of the state space, **Problem 1** is solved via the verification of the accepting path of the LTL specification in this section. To be specific, the graph-based approach is presented in Section VI-A to verify the realization of the accepting path. The realization verification results in finite local LTL formulas in Section VI-B.

A. Graph-based Verification

To realize the accepting path π of the LTL formula φ, it suffices to show that any pair (π_i, π_{i+1}) from π can be connected. For this purpose and to facilitate the controller synthesis afterwards, we consider an auxiliary path π_{ε}, which is derived from the accepting path π and $\varepsilon > 0$ given in Algorithm 3. Given the accepting path $\pi = \pi_0\pi_1\pi_2\ldots$, the auxiliary path is defined as $\pi_{\varepsilon} := \pi_0^{\varepsilon}\pi_1^{\varepsilon}\pi_2^{\varepsilon}\ldots$ such that $\mathbb{R}(\pi_i^{\varepsilon}) = \mathbb{R}^{\varepsilon}(\pi_i)$ and $i \in \mathbb{N}$. That is, for each $i \in \mathbb{N}$, $\mathbb{R}(\pi_i^{\varepsilon})$ is the ε-contraction of $\mathbb{R}^{\varepsilon}(\pi_i)$. Due to the contraction operation, the auxiliary path π_{ε} is called a *robust accepting path* of φ.

Example 2: Let us revisit Example 1. For the LTL formula $\varphi = \Box\Diamond\pi_1 \wedge \Box\Diamond\pi_2 \wedge \Diamond\pi_3 \wedge \neg\pi_3 \mathsf{U} \pi_2$, the accepting path is $\pi = \pi_0\pi_1\pi_2\pi_3(\pi_1\pi_2)^{\omega}$. The robust accepting path is defined as $\pi_{\varepsilon} = \pi_0^{\varepsilon}\pi_1^{\varepsilon}\pi_2^{\varepsilon}\pi_3^{\varepsilon}(\pi_1^{\varepsilon}\pi_2^{\varepsilon})^{\omega}$. For each $i \in \{0, 1, 2, 3\}$, $\mathbb{R}(\pi_i^{\varepsilon}) = \mathbb{R}^{\varepsilon}(\pi_i)$, which is shown in Fig. 5. ◁

Lemma 3: Given the LTL formula φ with the accepting path π and $\varepsilon > 0$, if the robust accepting path π_{ε} is realized, then the accepting path π is realized.

Proof: From Definition 7, the realization of the accepting path π_{ε} means that there exists a connected set $\mathbb{X} \subseteq \mathbb{X} \setminus \mathbb{O}$ such that $\mathbb{X} \cap \mathbb{R}(\pi_i^{\varepsilon}) \neq \varnothing$ for all $\pi_i^{\varepsilon} \in \pi_{\varepsilon}$. From the relation between π and π_{ε}, $\mathbb{R}(\pi_{\varepsilon})$ is an ε-contraction of $\mathbb{R}(\pi)$, and thus $\mathbb{X} \cap \mathbb{R}(\pi_i) \neq \varnothing$. That is, for all $\pi_i \in \pi$, it holds $\mathbb{X} \cap \mathbb{R}(\pi_i) \neq \varnothing$, which implies the realization of the accepting path π. ■

Lemma 3 shows the relation between the realization of π and π_{ε}, and next we focus on the robust accepting path π_{ε}. To verify the realization of π_{ε}, we generalize the graph $\mathcal{G}$ by including all symbols π_i^{ε}, $i \in \{0, 1, \ldots, |\pi_{\varepsilon}|\}$. To show this, the vertex set is generalized as $\bar{\mathcal{V}} = \mathbb{V} \cup \{\pi_i^{\varepsilon} : i \in$

$\{0, 1, \ldots, |\pi_\varepsilon|\}\}$. Since $\mathbb{R}(\pi_\varepsilon)$ may intersect with some cells in $\mathbb{Z}$, these intersection relations can be checked and included into the generalized edge set $\bar{\mathcal{E}} \subseteq \bar{\mathcal{V}} \times \bar{\mathcal{V}}$. Hence, the generalized graph is $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$, and each path $\mathbb{p}$ in $\bar{\mathcal{G}}$ results in a sequence of finite cells denoted as $\mathbb{P} := \{\mathbb{P}_k = \mathbb{R}(\mathsf{v}_k) \in \mathbb{Z} : \mathsf{v}_k \in \mathbb{p}, k \in \mathbb{N}^+\}$. The intersection region set in $\mathbb{P}$ is denoted as $\mathbb{I} := \{\mathbb{I}_{ij} = \mathbb{P}_i \cap \mathbb{P}_j : \forall \mathbb{P}_i, \mathbb{P}_j \in \mathbb{P}\}$. From the graph $\bar{\mathcal{G}}$, the next theorem is derived to verify the realization of π_ε.

Theorem 1: Consider the state space $\mathbb{X} \subseteq \mathbb{R}^n$, the LTL formula φ with the accepting path π_ε and a given precision $\varepsilon > 0$. Let the generalized graph $\bar{\mathcal{G}}$ be defined as above. the robust accepting path π_ε is realized in $\mathbb{X}$, if and only if

(i) there exists a path $\mathbb{p}$ in $\bar{\mathcal{G}}$ such that π_ε is embedded, that is, there exits at least one sub-path in $\mathbb{p}$ such that for any $k \in \{0, 1, \ldots, |\pi_\varepsilon| - 1\}$, π^ε_k and π^ε_{k+1} are connected;

(ii) for each $\mathbb{P}_i \in \mathbb{P}$ with $\mathbb{P}$ from the path $\mathbb{p}$,

(ii-a) $\mathbb{P}^\varepsilon_i \setminus \mathbf{O}$ is a connected region;

(ii-b) otherwise, there exists a subregion $\mathbb{P}'_i \subset \mathbb{P}^\varepsilon_i \setminus \mathbf{O}$ such that $\mathbb{P}'_i \cup \mathbb{I}^\varepsilon_{(i-1)i} \cup \mathbb{I}^\varepsilon_{i(i+1)}$ is connected, and $\mathbb{P}'_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$ if $\mathbb{P}_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$.

Proof: See Appendix A. ■

In Theorem 1, item (i) implies that any pair $(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$ in π_ε is connected in a sub-path $\mathbb{p}_i \subseteq \mathbb{p}$. Item (ii) is to verify the connectedness property of each cell in $\mathbb{P}$ from $\mathbb{p}$. For each cell in $\mathbb{P}$, it either is itself connected or has a connected subregion to intersect the previous and next cells. From Theorem 1, the sequence of finite cells from the path $\mathbb{p}$ is denoted as

$$\mathbb{P} := \{\mathbb{P}_i = \mathbb{R}(\mathsf{v}_i) \in \mathbb{Z} : \mathsf{v}_i \in \mathbb{p} \subset \bar{\mathcal{V}}\}. \tag{12}$$

Similarly, $\mathbb{P}(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$ is the sequence of finite cells from the sub-path $\mathbb{p}_i$ for a pair $(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$. In particular, if there exist loops in π_ε, then there exists loops in $\mathbb{p}$ and each loop can be divided into two sub-sequences in $\mathbb{P}$. If a loop in π_ε appears finite/infinite times, then the corresponding two sub-sequences in $\mathbb{P}$ only need to be checked once via item (ii).

Theorem 1 offers a graph-based approach to verify the realization of the accepting path and thus solves **Problem 1**. Note that the path $\mathbb{p}$ is not necessarily unique. Similar to Algorithm 3, the contraction operation is applied to facilitate the controller design in Section VII, and can be reduced if only the realization is addressed.

Remark 3: The computational complexity of the realization verification essentially depends on the number of all cells. In terms of graph searching, the time for searching an admissible path depends on the number of all cells. In terms of graph generation, the existence of all edges lies in Algorithm 3 to verify the admissibility of all intersection regions, which has the computational complexity $\mathcal{O}((N+M)n^3)$; see [23], [39]. Therefore, with an increasing number of cells, more operations are needed and the computational complexity increases. We stress that the realization verification is a premise for the control design, and thus its computational complexity is not included in the following sections. △

B. Decomposition into Local LTL Formulas

Once the accepting path of the LTL formula can be realized, the obtained path $\mathbb{p}$ can be projected into a sequence of cells in (12). In this subsection, we formulate a local LTL formula for each cell in the derived sequence.

From Section V, any two neighbor cells overlap such that it is possible for the considered system to move between these two cells. Given the LTL formula φ, let $\mathbb{P}$ be the sequence of the cells derived from Theorem 1, and $\mathsf{L} := |\mathbb{P}|$. For each cell $\mathbb{P}^\varepsilon_i \in \mathbb{P}^\varepsilon$ with $i \in \mathbb{L} = \{0, \ldots, \mathsf{L}\}$, the following three types of local regions are derived.

(1) For the first cell $\mathbb{P}_0$, $\mathbb{P}^\varepsilon_0 \setminus \mathbf{O}$ is the local initial region in $\mathbb{P}_0$. For $i \in \mathbb{N}^+$, the local initial region is $(\mathbb{P}^\varepsilon_{i-1} \cap \mathbb{P}^\varepsilon_i) \setminus \mathbf{O}$.

(2) For each $i \in \mathbb{N}$, if $\mathbb{P}^\varepsilon_{i+1}$ does not exist, then there exists no local target region in $\mathbb{P}^\varepsilon_i$. If $\mathbb{P}^\varepsilon_{i+1}$ exists, then $\mathcal{T}_i := (\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i+1}) \setminus \mathbf{O}$ is the local target region in $\mathbb{P}^\varepsilon_i$.

(3) If $\mathbb{P}^\varepsilon_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$ and $\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i+1} \cap \mathbb{R}(\pi_\varepsilon) = \varnothing$, then all regions in $\mathbb{P}^\varepsilon_i \cap \mathbb{R}(\pi_\varepsilon)$ are called *local internal regions*, whose number is denoted as $\ell_i \in \mathbb{N}$ and which are denoted as $\mathcal{I}_{il}$ with $l \in \mathbb{L}_i := \{1, \ldots, \ell_i\}$.

Observe that the local target region of the current cell will be the local initial region of the next cell (if it exists). By verifying whether $\mathbb{P}^\varepsilon_i \cap \mathbb{R}(\pi_\varepsilon)$ is empty, we can establish all local internal regions in $\mathbb{P}^\varepsilon_i$. The local target and internal regions depend on the LTL formula φ and thus do not necessarily exist. If they are existent, then the local target region may intersect with local internal regions, that is, $\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i+1} \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$. In this case, the local target region is specified as $\mathcal{T}_i = (\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i+1} \cap \mathbb{R}(\pi_\varepsilon)) \setminus \mathbf{O}$, and accordingly, the local initial region can be specified as $(\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i-1} \cap \mathbb{R}(\pi_\varepsilon)) \setminus \mathbf{O}$. Based on these three types of local regions, the local LTL formula of each $\mathbb{P}_i \in \mathbb{P}$, which is denoted as $\bar{\varphi}^\varepsilon_i$, can be derived as follows.

Let the accepting path be π_ε. For any pair $(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$ with $\pi^\varepsilon_i, \pi^\varepsilon_{i+1} \in \pi_\varepsilon$ and $i \in \mathbb{N}$, its sub-path is denoted as $\mathbb{p}_i \subseteq \mathbb{p}$, which results in the subsequence $\mathbb{P}(\pi^\varepsilon_i, \pi^\varepsilon_{i+1}) \subseteq \mathbb{P}$. Define $\mathsf{L}_i := |\mathbb{P}(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})|$. For each cell $\mathbb{P}_{ij} \in \mathbb{P}(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$, where $j \in \mathbb{L}_i := \{1, \ldots, \mathsf{L}_i\}$, we denote by $\mathcal{T}_{ij}$ the local target region and by $\{\mathcal{I}_{ijl} : l \in \mathbb{L}_{ij}\}$ all local internal regions. The goal in the cell $\mathbb{P}_{ij}$ has the following three parts.

(a) If the system state moves into $\mathbb{P}^\varepsilon_{ij}$, then the system state is expected to stay in $\mathbb{P}^\varepsilon_{ij}$ while avoiding $\mathbf{O}$. We formulate this part into an LTL formula $\bar{\varphi}^\varepsilon_{ij1} := \Box\tilde{\varphi}^\varepsilon_{ij1}$ with $\mathbb{R}(\tilde{\varphi}^\varepsilon_{ij1}) = \mathbb{P}^\varepsilon_{ij} \setminus \mathbf{O}$.

(b) If the local target region $\mathcal{T}_{ij}$ exists, then in $\mathbb{P}^\varepsilon_{ij}$ the system state is expected to move into $\mathcal{T}_{ij}$ eventually. This part is formulated into an LTL formula $\bar{\varphi}^\varepsilon_{ij2} := \Diamond\tilde{\varphi}^\varepsilon_{ij2}$ with $\mathbb{R}(\tilde{\varphi}^\varepsilon_{ij2}) = \mathcal{T}_{ij}$.

(c) If there exist local internal regions in $\mathbb{P}^\varepsilon_{ij}$, then the system state is expected to visit all or some of these local internal regions. The following two cases are addressed.

- If $\mathsf{L}_i > 1$, then only the first and last cells may have local internal regions, i.e, $\mathbb{R}(\pi^\varepsilon_i)$ or $\mathbb{R}(\pi^\varepsilon_{i+1})$. In this case, an LTL formula in $\mathbb{P}^\varepsilon_{ij}$ (if it exists) is formulated as $\bar{\varphi}^\varepsilon_{ij3} := \Diamond\tilde{\varphi}^\varepsilon_{ij3}$, where $\mathbb{R}(\tilde{\varphi}^\varepsilon_{ij3}) \in \{\mathcal{I}_{ijl} : l \in \mathbb{L}_{ij}\}$. In particular, in the last cell of $\mathbb{P}(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$, $\bar{\varphi}^\varepsilon_{ij3}$ can be either $\Diamond\tilde{\varphi}^\varepsilon_{ij3}$ or $\Box\tilde{\varphi}^\varepsilon_{ij3}$, which to be chosen depends on the LTL formula φ_ε. If $(\pi^\varepsilon_i, \pi^\varepsilon_{i+1})$ is the last pair in π_ε and the system state is expected to stay in $\mathbb{R}(\pi^\varepsilon_{i+1})$, then $\bar{\varphi}^\varepsilon_{ij3} = \Box\tilde{\varphi}^\varepsilon_{ij3}$; otherwise, $\bar{\varphi}^\varepsilon_{ij3} = \Diamond\tilde{\varphi}^\varepsilon_{ij3}$.

- If $\mathsf{L}_i = 1$, then the LTL formula $\bar{\varphi}^{\varepsilon}_{ij3}$ is either $\Diamond(\hat{\varphi}^{\varepsilon}_{ij3} \wedge \Diamond\check{\varphi}^{\varepsilon}_{ij3})$ or $\Diamond(\hat{\varphi}^{\varepsilon}_{ij3} \wedge \Box\check{\varphi}^{\varepsilon}_{ij3})$, where $\mathbb{R}(\hat{\varphi}^{\varepsilon}_{ij3}), \mathbb{R}(\check{\varphi}^{\varepsilon}_{ij3}) \in \{\mathcal{J}_{ijl} : l \in \mathbb{L}_{ij}\}$. Similar to the case $\mathsf{L}_i > 1$, the LTL formula φ_ε determines the exact form of $\bar{\varphi}^{\varepsilon}_{ij3}$. For instance, if $(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$ is the last pair in π_ε and the system state is expected to stay in $\mathbb{R}(\pi^{\varepsilon}_{i+1})$, then $\bar{\varphi}^{\varepsilon}_{ij3} = \Diamond(\hat{\varphi}^{\varepsilon}_{ij3} \wedge \Box\check{\varphi}^{\varepsilon}_{ij3})$.

Based on the above three parts, we have $\bar{\varphi}^{\varepsilon}_{ij} := \bar{\varphi}^{\varepsilon}_{ij1} \wedge \bar{\varphi}^{\varepsilon}_{ij2} \wedge \bar{\varphi}^{\varepsilon}_{ij3}$. From the above three types of local regions, we can see that $\bar{\varphi}^{\varepsilon}_{i1}$ always exists, whereas the existence of $\bar{\varphi}^{\varepsilon}_{i2}$ and $\bar{\varphi}^{\varepsilon}_{i3}$ depends on the existence of local target and internal regions. If all expansion and contraction operators in the above analysis are removed, then we can follow the same mechanism to derive the local LTL formula $\bar{\varphi}_{ij}$ for the cell $\mathbb{P}_{ij}$.

Theorem 1 shows the realization of the pair $(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$ in π_ε via the sequence $\mathbb{P}(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$. To ensure the considered system moves from $\mathbb{R}(\pi^{\varepsilon}_i)$ to $\mathbb{R}(\pi^{\varepsilon}_{i+1})$, the considered system needs to satisfy all local LTL formulas $\bar{\varphi}^{\varepsilon}_{ij}$ sequentially, which results in the following local LTL formula for $\mathbb{P}(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$:

$$\bar{\varphi}^{\varepsilon}(\mathbb{p}_i) := \Diamond(\bar{\varphi}^{\varepsilon}_{i1} \wedge \Diamond(\bar{\varphi}^{\varepsilon}_{i2} \wedge \ldots \wedge \Diamond\bar{\varphi}^{\varepsilon}_{i\mathsf{L}_i})), \tag{13}$$

where, the order of all local LTL formulas $\bar{\varphi}^{\varepsilon}_{ij}$ is based on the sub-path $\mathbb{p}_i$. For any two successive sequences $\mathbb{P}(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$ and $\mathbb{P}(\pi^{\varepsilon}_{i+1}, \pi^{\varepsilon}_{i+2})$, we notice that the last cell in $\mathbb{P}(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$ is the first cell in $\mathbb{P}(\pi^{\varepsilon}_{i+1}, \pi^{\varepsilon}_{i+2})$. In this way, the local LTL formula (13) can be constructed iteratively.

In addition, if there exists a loop in π_ε, then we can consider the following two cases. The first case is that the loop appears only one time. In this case, the loop can be divided into two parts, and a local LTL formula in the form (13) can be derived for each part. For instance, if there exists a loop between π^{ε}_i and π^{ε}_{i+1}, then the local LTL formulas $\bar{\varphi}^{\varepsilon}(\mathbb{p}_i)$ for $\mathbb{P}(\pi^{\varepsilon}_i, \pi^{\varepsilon}_{i+1})$ and $\bar{\varphi}^{\varepsilon}(\mathbb{p}_{i+1})$ for $\mathbb{P}(\pi^{\varepsilon}_{i+1}, \pi^{\varepsilon}_i)$ can be derived, and for this loop, the local LTL formula is defined as $\Diamond(\bar{\varphi}^{\varepsilon}(\mathbb{p}_i) \wedge \Diamond\bar{\varphi}^{\varepsilon}(\mathbb{p}_{i+1}))$. The second case is that the loop appears infinitely. Similar to the first case, the loop is divided into two parts and the local LTL formulas for these two parts can be derived. The the local LTL formula for this loop is defined as $\Box(\Diamond(\bar{\varphi}^{\varepsilon}(\mathbb{p}_i) \wedge \Diamond\bar{\varphi}^{\varepsilon}(\mathbb{p}_{i+1})))$. From the above analysis and the path $\mathbb{p}$, the local LTL formulas for all $\mathbb{P}_i \in \mathbb{P}$ can be combined into the following form:

$$\begin{aligned}&\Diamond(\bar{\varphi}^{\varepsilon}_0 \wedge \Diamond(\bar{\varphi}^{\varepsilon}_1 \wedge \ldots \wedge \Box(\Diamond(\bar{\varphi}^{\varepsilon}_j \wedge \Diamond(\bar{\varphi}^{\varepsilon}_{j+1} \wedge \ldots \wedge \Diamond\bar{\varphi}^{\varepsilon}_{j+k}))) \\ &\wedge \ldots \wedge \Diamond\bar{\varphi}^{\varepsilon}_{\mathsf{L}})),\end{aligned} \tag{14}$$

where, $\bar{\varphi}^{\varepsilon}_i$ is the local LTL formula corresponding to each cell $\mathbb{P}_i \in \mathbb{P}$, $i \in \mathbb{L} = \{0, \ldots, \mathsf{L}\}$ and $\mathsf{L} = |\mathbb{P}|$. Note that (14) is based on the robust accepting path π_ε. For the accepting path π, we can follow the above mechanism to establish all local LTL formulas, which are denoted as $\bar{\varphi}_i$ and whose combination is of the following form:

$$\begin{aligned}&\Diamond(\bar{\varphi}_0 \wedge \Diamond(\bar{\varphi}_1 \wedge \ldots \wedge \Box(\Diamond(\bar{\varphi}_j \wedge \Diamond(\bar{\varphi}_{j+1} \wedge \ldots \wedge \Diamond\bar{\varphi}_{j+k}))) \\ &\wedge \ldots \wedge \Diamond\bar{\varphi}_{\mathsf{L}})).\end{aligned} \tag{15}$$

Each φ_i corresponds to $\bar{\varphi}^{\varepsilon}_i$. Furthermore, the following proposition is derived directly.

Proposition 1: Consider the LTL formula φ and $\varepsilon > 0$. Let the decompositions (14) and (15) be derived as above. φ is satisfied if all $\bar{\varphi}_i$ in (14) are satisfied. Each φ_i is satisfied if the corresponding $\bar{\varphi}^{\varepsilon}_i$ are satisfied.

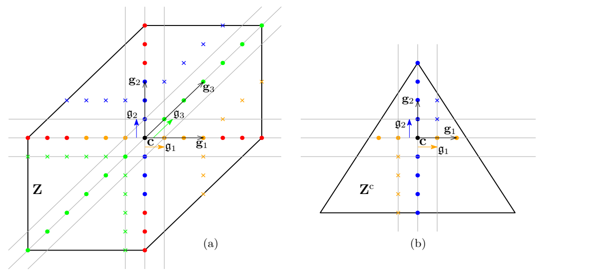

Fig. 6. Illustration of the proposed approximation approach. (a) Approximation of the zonotope $\mathbf{Z} = \{\mathbf{c}, \mathbf{G}\} \subset \mathbb{R}^2$ with $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$. (b) Approximation of the constrained zonotope $\mathbf{Z}^{\mathrm{c}} = \{\mathbf{c}, \mathbf{G}, \mathbf{A}, \mathbf{b}\} \subset \mathbb{R}^2$ with $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2)$. The dots belong to the set $\mathbb{F}_l$ or $\mathbb{F}^{\mathrm{c}}_l$, whereas the crosses belong to the set $\overline{\mathbb{F}}_l$ or $\overline{\mathbb{F}}^{\mathrm{c}}_l$.

With Proposition 1, the controller synthesis problem (i.e., **Problem 2**) is decomposed into finite local ones, which will be discussed in detail in the next section.

VII. Abstraction-Based Controller Synthesis

After solving **Problem 1**, we propose an abstraction-based local-to-global control strategy to deal with **Problem 2** in this section. To be specific, we take advantage of the properties of zonotopes to propose a novel approximation of each cell in $\mathbb{P}$ in Section VII-A and to construct the local symbolic abstraction in Section VII-B. The local-to-global abstract control strategy is proposed in Section VII-C.

A. Approximation of State and Input Sets

Note that each cell in $\mathbb{P}$ is either a zonotope or a constrained zonotope. Without loss of generality, we focus on the approximation of any zonotope $\mathbf{Z} \subset X_1$ and constrained zonotope $\mathbf{Z}^{\mathrm{c}} \subset X_1$ in this subsection.

1) Approximation of Zonotopes: Different from the classic quantization-based technique [9], we approximate the zonotope $\mathbf{Z} \subset X_1$ via the properties of zonotopes. Let $\mathbf{Z} = \{\mathbf{c}, \mathbf{G}\}$ with $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_\ell)$ and $\boldsymbol{\ell} := \{1, \ldots, \ell\}$, where $\ell \in \mathbb{N}$.

First, each $\mathbf{g}_l$ with $l \in \boldsymbol{\ell}$ is partitioned uniformly. Let $\mathsf{N}_l \in \mathbb{N}$ be the partition number, and define $\mathbf{G}_{\mathsf{b}} := (\mathfrak{g}_1, \ldots, \mathfrak{g}_\ell) = (\mathsf{N}_1^{-1}\mathbf{g}_1, \ldots, \mathsf{N}_\ell^{-1}\mathbf{g}_\ell)$ as the basic generator matrix. Thus, $(\mathbf{g}_1, \ldots, \mathbf{g}_\ell) = (\mathsf{N}_1\mathfrak{g}_1, \ldots, \mathsf{N}_\ell\mathfrak{g}_\ell)$. Since $\mathfrak{g}_l$ is the basis of $\mathbf{g}_l$, we derive a finite number of points consisting of the set below

$$\mathbb{F}_l := \{\mathbf{c} \pm \mathfrak{g}_l, \ldots, \mathbf{c} \pm \mathsf{N}_l\mathfrak{g}_l, \ldots, \mathbf{c} \pm \mathsf{M}_l\mathfrak{g}_l\}, \tag{16}$$

where $\mathsf{M}_l \geq \mathsf{N}_l$ is the largest integer such that $\mathbf{c} \pm \mathsf{M}_l\mathfrak{g}_l \in \mathbf{Z}$; see the red points in Fig. 6(a). In (16), M_l depends on other generators $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_{l-1}, \mathfrak{g}_{l+1}, \ldots, \mathfrak{g}_\ell\}$, and N_l is constrained by a parameter $\mu > 0$. That is, $\max_{l \in \boldsymbol{\ell}}\{\|\mathfrak{g}_l\|\} \leq \mu$ is imposed to constrain the choice of $\{\mathsf{N}_l : l \in \boldsymbol{\ell}\}$. $\mu > 0$ depends on the desired equivalence relation; see Section VII-B. In addition, the choice of the set $\{\mathsf{N}_l : l \in \boldsymbol{\ell}\}$ affects the set $\{\mathsf{M}_l : l \in \boldsymbol{\ell}\}$.

Second, all points in $\mathbb{F}_l$ are the bases to generate other points in $\mathbf{Z}$. Given a $\mathsf{p}_{lj} \in \mathbb{F}_l$, $0 < j \leq \mathsf{M}_l$, other points are generated in the following way: we first take $\mathsf{p}_{lj} \in \mathbb{F}_l$ as the basis point and any generator from $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_\ell\}$ as the direction, and then generate all the points leading to the following set

$$\bar{\mathbb{F}}_l := \bigcup_{0<j\leq \bar{\mathsf{N}}_l} \bigcup_{k\in\boldsymbol{\ell}, k\neq l} \{\mathsf{p}_{lj} \pm \mathfrak{g}_k, \ldots, \mathsf{p}_{lj} \pm \mathsf{N}_k\mathfrak{g}_k\} \cap \mathbf{Z}. \quad (17)$$

This generation mechanism is terminated until the generated point is not in $\mathbf{Z}$. Since $\mathbb{F}_l$ is finite, this generation mechanism can be implemented recursively and terminated in finite steps.

Finally, combining all generated points, the zonotope $\mathbf{Z}$ is approximated by the following set:

$$\mathcal{A}(\mathbf{Z}) := \{\mathbf{c}\} \cup (\cup_{l\in\boldsymbol{\ell}}(\mathbb{F}_l \cup \bar{\mathbb{F}}_l)). \quad (18)$$

Remark 4: We examine the above approximation in detail, and compare the proposed approach with the existing ones.

(i) The proposed approach is based on the properties of zonotopes. That is, the center and all generators of $\mathbf{Z}$ are applied in the approximation, which implies that each zonotope can have its own approximation. However, the classic quantization-based techniques [9], [31], [40] are to approximate the whole state space uniformly.

(ii) The proposed approach can be simplified by choosing n linearly independent generators from $\{\mathbf{g}_1, \ldots, \mathbf{g}_\ell\}$. Note that $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_\ell)$ is full-rank and $\ell \geq n$. Hence, we can choose n linear independent generators $\{\mathbf{g}_{l_1}, \ldots, \mathbf{g}_{l_n}\} \subseteq \{\mathbf{g}_1, \ldots, \mathbf{g}_\ell\}$ with $\{l_1, \ldots, l_n\} \subseteq \boldsymbol{\ell}$ and derive a novel basic generator matrix $\mathbf{G}'_{\mathrm{b}}$. The following approximation can be proceeded similarly. For instance, we can choose any two generators in Fig. 6(a). This simplification reduces the number of abstract states in (18), and will potentially reduce the transition number in the symbolic abstraction and further affect the computational complexity. If all generators are orthogonal, then the proposed approach is reduced to those in many existing works [16], [32]. In this aspect, the proposed approximation approach is more general.

(iii) Each generator has its own partition number and hence has its own partition. Here we present two special cases to unify the partition of all generators. The first case is to select basic generators with the same length for all $\{\mathbf{g}_1, \ldots, \mathbf{g}_\ell\}$ such that the partition is unified. Let the length be $\mathsf{h} > 0$. Hence, $\mathfrak{g}_l := \mathsf{h}\mathbf{g}_l/\|\mathbf{g}_l\|$, $\mathsf{N}_l := \lfloor \|\mathbf{g}_l\|/\|\mathfrak{g}_l\| \rfloor$ with the ceiling operator $\lfloor\cdot\rfloor$, and the set $\mathbb{F}_l$ in (16) can be defined similarly. The second case is to set the same partition number for all $\{\mathbf{g}_1, \ldots, \mathbf{g}_\ell\}$. That is, $\mathsf{N}_l = \mathsf{N} \in \mathbb{N}$ for all $l \in \boldsymbol{\ell}$. Hence, the set $\mathbb{F}_l$ in (16) can be defined similarly. After defining the set $\mathbb{F}_l$, the following approximation is proceeded similarly. △

2) Approximation of Constrained Zonotopes: Following the above techniques, any constrained zonotope $\mathbf{Z}^{\mathrm{c}} = \{\mathbf{c}, \mathbf{G}, \mathbf{A}, \mathbf{b}\}$ can be approximated similarly. The only difference lies in the constraint condition $\mathbf{A}\xi = \mathbf{b}$, which has effects on M_l and further on the set $\mathbb{F}_l$ as in (16). In addition, the set $\bar{\mathbb{F}}_l$ in (17) is modified to be

$$\bar{\mathbb{F}}^{\mathrm{c}}_l := \bigcup_{0<j\leq \bar{\mathsf{N}}_l} \bigcup_{k\in\boldsymbol{\ell}, k\neq l} \{\mathsf{p}_{lj} \pm \mathfrak{g}_k, \ldots, \mathsf{p}_{lj} \pm \mathsf{M}_k\mathfrak{g}_k\} \cap \mathbf{Z}^{\mathrm{c}}. \quad (19)$$

Therefore, $\mathbf{Z}^{\mathrm{c}}$ is approximated via the following set

$$\mathcal{A}(\mathbf{Z}^{\mathrm{c}}) := \{\mathbf{c}\} \cup (\cup_{l\in\boldsymbol{\ell}}(\mathbb{F}^{\mathrm{c}}_l \cup \bar{\mathbb{F}}^{\mathrm{c}}_l)). \quad (20)$$

Remark 5: Besides the proposed approach, other approaches like tiling and zonotopal subdivision techniques [41], [42] can be applied to the approximation of zonotopes. These techniques take centrally symmetric advantages of zonotopes to partition zonotopes into sub-zonotopes. For instance, the connection of all generated points leads to a line arrangement, which determines the zonotope division [43]. However, these techniques may result in huge constructional and computational complexities. How to apply these techniques to approximate constrained zonotopes is unknown, since constrained zonotopes are not necessarily centrally symmetric. △

With the approximation of (constrained) zonotopes, we introduce the norm $\|\cdot\|_{\mathbf{G}}$ based on the generator matrix $\mathbf{G}$ and the vector projection. Given any $\mathbf{v} \in \mathbb{R}^n$, $\|\mathbf{v}\|_{\mathbf{G}} := \max_{l\in\boldsymbol{\ell}}\{\mathbf{v}\cdot\mathbf{g}_l/\|\mathbf{g}_l\|\}$, where $\mathbf{v}\cdot\mathbf{g}_l/\|\mathbf{g}_l\|$ is the scalar projection of $\mathbf{v}$ onto $\mathbf{g}_l$. If all generators are orthogonal, then this norm is reduced to the infinity norm. With the norm $\|\cdot\|_{\mathbf{G}}$, we can see that for any $x \in \mathbf{Z}$ (or $x \in \mathbf{Z}^{\mathrm{c}}$), there exists $q \in \mathcal{A}(\mathbf{Z})$ (or $q \in \mathcal{A}(\mathbf{Z}^{\mathrm{c}})$) such that $\|x - q\|_{\mathbf{G}} \leq 0.5\max_{l\in\boldsymbol{\ell}}\{\|\mathfrak{g}_l\|\}$.

3) Approximation of Input Set: For a zonotope $\mathbf{Z} \subset X_1$, its input set is defined as $U_1(\mathbf{Z}) := \cup_{x\in\mathbf{Z}} \mathsf{enab}(x) \subseteq U_1$, and approximated as follows. Given any $q \in \mathcal{A}(\mathbf{Z})$, the reachable set of $\mathbf{T}_\tau(\Sigma, \mathbf{Z})$ from q is $\mathsf{Reach}(\tau, q) := \{x' \in \mathbf{Z} : x' = \mathbf{x}(\tau, q, u), u \in U_1(\mathbf{Z})\}$, which is well-defined due to $U_1(\mathbf{Z})$. Given any $\eta \in \mathbb{R}^+$, consider the set $\$_\eta(\tau, q) := \{v \in \mathcal{A}(\mathbf{Z}) :$ there exists $z \in \mathsf{Reach}(\tau, q)$ such that $\|v - z\|_{\mathbf{G}} \leq 0.5\eta\}$, which is a countable set. Note that η is constrained by the desired precision; see Section VII-B. A function $\psi : \$_\eta(\tau, q) \to U_1(\mathbf{Z})$ is defined: for any $v \in \$_\eta(\tau, q)$, there exists $u_1 = \psi(v) \in U_1(\mathbf{Z})$ such that $\|v - \mathbf{x}(\tau, q, u_1)\|_{\mathbf{G}} \leq 0.5\eta$. Let $U_2(q) := \psi(\$_\eta(\tau, q))$, which captures all inputs that can be applied at $q \in \mathcal{A}(\mathbf{Z})$. The set $U_2(q)$ is countable since $U_2(q)$ is the image of the function ψ on $\$_\eta(\tau, q)$. Hence, $U_1(\mathbf{Z})$ is approximated via the following countable set:

$$U_2(\mathbf{Z}) := \bigcup_{q\in\mathcal{A}(\mathbf{Z})} U_2(q). \quad (21)$$

That is, the set $U_2(\mathbf{Z})$ approximates the set $U_1(\mathbf{Z})$ in the following way: given any $q \in \mathcal{A}(\mathbf{Z})$ and any $u_1 \in U_1(\mathbf{Z})$, there exists $u_2 \in U_2(q)$ such that $\|\mathbf{x}(\tau, q, u_1) - \mathbf{x}(\tau, q, u_2)\|_{\mathbf{G}} \leq \eta$.

The above approximation can be applied similarly to constrained zonotopes, which is omitted here. From (4) and (11), all cells in $\mathbb{Z}$ are (constrained) zonotopes, and hence the above approximation can also be applied to any cell in $\mathbb{Z}$.

B. Abstraction Construction

With the approximation in the previous subsection, the symbolic abstraction is constructed in this subsection. For each cell $\mathbb{P}_i \in \mathbb{P}$, $i \in \mathbb{L}$, the symbolic abstraction for the system $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ is constructed as a transition system $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) = (X_{2i}, X^0_{2i}, U_{2i}, \Delta_{2i}, Y_{2i}, H_{2i})$, where,

- the set of states is $X_{2i} = \mathcal{A}(\mathbb{P}_i)$;
- the set of initial states is $X^0_{2i} = \mathcal{A}(\mathbb{P}^i_0)$ with $\mathbb{P}^i_0 \subseteq \mathbb{P}_i$;
- the set of inputs is $U_{2i} = U_2(\mathbb{P}_i)$;

- the transition relation is given as follows: for $q_1, q_2 \in X_{2i}$ and $u \in U_{2i}$, $q_2 \in \Delta_{2i}(q_1, u)$ if and only if

$$q_2 \in \{\bar{q} \in X_{2i} : \|\mathbf{x}(\tau, q_1, u) - \bar{q}\|_{\mathbf{G}} \leq (1 + e^{L_i\tau})\varepsilon\}, \quad (22)$$

  where $L_i > 0$ is the local Lipschitz constant of f in $\mathbb{P}_i$;
- the set of outputs is $Y_{2i} = X_{2i}$;
- the output map is $H_{2i} = \mathrm{Id}$.

The parameters $\mu_i, \eta_i > 0$ are from the approximation of $\mathbb{P}_i \in \mathbb{P}$. $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ is metric when Y_{2i} is equipped with the metric $\mathbf{d}(y', y) = \|y - y'\|$ for all $y, y' \in Y_{2i}$. $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ is nondeterministic from the transition relation (22). Next, we show the FRR from $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ to $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$.

Theorem 2: Consider the systems $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ with the parameters $\tau, \mu_i, \eta_i \in \mathbb{R}^+$. Given a precision $\varepsilon \in \mathbb{R}^+$, if the relation $\mathcal{F} : X_{1i} \to X_{2i}$ is defined as

$$\mathcal{F}(x) = \{q \in X_{2i} : \|x - q\|_{\mathbf{G}} \leq \varepsilon, x \in X_{1i}\}, \quad (23)$$

then $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) \preceq_{\mathcal{F}} \mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$.

Proof: See Appendix B. ■

Since the state space is constrained into $\mathbb{P}_i \in \mathbb{P}$, the symbolic abstraction is local and can be different for different cells in $\mathbb{P}$, which allows for the flexibility in the abstraction construction. If the system Σ satisfies the stability condition as in [8], then a special case of the symbolic abstraction can be derived. To be specific, the system Σ is *incrementally globally asymptotically stable (δ-GAS)* in $\mathbb{P}_i$, if there exists $\beta_i \in \mathcal{KL}$ such that for all $x_1, x_2 \in \mathbb{P}_i, u \in U_1(\mathbb{P}_i)$ and $t \in \mathbb{R}^+$,

$$\|\mathbf{x}(t, x_1, u) - \mathbf{x}(t, x_2, u)\|_{\mathbf{G}} \leq \beta_i(\|x_1 - x_2\|_{\mathbf{G}}, t). \quad (24)$$

Note that the proposed norm in Section VII-A does not affect the incremental stability due to the norm equivalence. In this case, the symbolic abstraction of $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ is constructed as $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) = (\bar{X}_{2i}, \bar{X}^0_{2i}, \bar{U}_{2i}, \bar{\Delta}_{2i}, \bar{Y}_{2i}, \bar{H}_{2i})$,

- the set of states is $\bar{X}_{2i} = \mathcal{A}(\mathbb{P}_i)$;
- the set of initial states is $\bar{X}^0_{2i} = \mathcal{A}(\mathbb{P}^i_0)$ with $\mathbb{P}^i_0 \subseteq \mathbb{P}_i$;
- the set of inputs is $\bar{U}_{2i} = U_2(\mathbb{P}_i)$;
- the transition relation is given as follows: for $q_1, q_2 \in \bar{X}_{2i}$ and $u \in \bar{U}_{2i}$, $q_2 \in \bar{\Delta}_{2i}(q_1, u)$ if and only if

$$q_2 \in \{\bar{q} \in \bar{X}_{2i} : \|\mathbf{x}(\tau, q_1, u) - \bar{q}\|_{\mathbf{G}} \leq 0.5\mu_i\}; \quad (25)$$

- the set of outputs is $\bar{Y}_{2i} = \bar{X}_{2i}$;
- the output map is $\bar{H}_{2i} = \mathrm{Id}$.

Similar to Theorem 2, the following theorem establishes the ε-ABR between $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$.

Theorem 3: Consider the δ-GAS system $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ and its symbolic model $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ with the parameters $\tau, \mu_i, \eta_i \in \mathbb{R}^+$. Given a precision $\varepsilon > 0$, if

$$\beta_i(\varepsilon, \tau) + \mu_i + 0.5\eta_i \leq \varepsilon, \quad (26)$$

where β_i is given in (24), then $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) \simeq_\varepsilon \mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$.

Proof: See Appendix C. ■

In the construction of $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, the difference is the transition relation. The transition relation in $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ is non-deterministic and the FRR defines which transition to be chosen, whereas the transition relation in $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ is deterministic and the ε-ABR constrains the choices of the parameters via (26). From Theorems 2-3, the proposed construction approach can be applied to general nonlinear systems, which does not necessarily need to be δ-GAS. In addition, the global δ-GAS assumption in [8], [9], [34] is relaxed into the local version (24), which implies that the proposed approach is less conservative.

For each cell $\mathbb{P}_i \in \mathbb{P}$, $i \in \mathbb{L}$, the symbolic abstraction can be constructed as either $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ or $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$. That is, different cells have their own symbolic models such that either the FRR or the ε-ABR is satisfied. To unify the notations for all cells, we use $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ to denote the symbolic abstraction. All parameters μ_i, η_i with $i \in \mathbb{L}$ are combined as $\boldsymbol{\mu} := (\mu_0, \ldots, \mu_{\mathsf{L}})$ and $\boldsymbol{\eta} := (\eta_0, \ldots, \eta_{\mathsf{L}})$. For the set $\mathbb{P}$, its symbolic abstraction is denoted as $\mathcal{T}_{\tau,\boldsymbol{\mu},\boldsymbol{\eta}}(\Sigma, \mathbb{P})$, which is the union of all $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$.

C. Local-to-Global Controller Strategy

Using the abstraction-based approach, the controller strategy is derived in this subsection. For this purpose, we first show the relations between global and local controllers.

Theorem 4: Consider the symbolic model $\mathcal{T}_{\tau,\boldsymbol{\mu},\boldsymbol{\eta}}(\Sigma, \mathbb{P})$ and the LTL formula φ. If for all $i \in \mathbb{L}$, there exists a local abstract controller $\mathcal{C}^i_{\mathsf{a}}$ such that the local LTL formula $\bar{\varphi}^\varepsilon_i$ is satisfied for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, then all local abstract controllers are combined such that φ is satisfied for $\mathcal{T}_{\tau,\boldsymbol{\mu},\boldsymbol{\eta}}(\Sigma, \mathbb{P})$.

Proof: From Proposition 1, the satisfaction of the LTL formula φ equals to the satisfaction of all local LTL formulas $\bar{\varphi}^\varepsilon_i$, $i \in \mathbb{L}$. Since the local abstract controller $\mathcal{C}^i_{\mathsf{a}}$ ensures the satisfaction of the local LTL formula $\bar{\varphi}^\varepsilon_i$ for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, all local abstract controllers can be combined as the following global abstract controller

$$\mathcal{C}_{\mathsf{a}}(q) := \mathcal{C}^i_{\mathsf{a}}(q), \quad \forall q \in \mathcal{A}(\mathbb{P}_i), i \in \mathbb{L}, \quad (27)$$

which further guarantees the satisfaction of φ for the system $\mathcal{T}_{\tau,\boldsymbol{\mu},\boldsymbol{\eta}}(\Sigma, \mathbb{P})$ and thus completes the proof. ■

Theorem 4 shows how to construct the global abstract controller (27) from all local abstract controllers. From Theorem 4, we can derive the similar result to bridge the controllers for the LTL formulas φ and $\bar{\varphi}_i$. That is, if $\mathcal{C}_i$ is the local controller such that $\bar{\varphi}_i$ is satisfied for $\mathcal{T}_\tau(\Sigma, \mathbb{P}_i)$, then the global controller for φ is constructed as

$$\mathcal{C}(x) := \mathcal{C}_i(x), \quad \forall x \in \mathbb{P}_i, i \in \mathbb{L}, \quad (28)$$

which shows the relation between global and local controllers. In the following, we establish the relation between the controllers for the LTL formulas $\bar{\varphi}^\varepsilon_i$ and $\bar{\varphi}_i$.

Theorem 5: Consider the systems $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$. Let $\varepsilon > 0$ be a given precision and the local LTL formulas be $\bar{\varphi}_i$ and $\bar{\varphi}^\varepsilon_i$.

(1) Let $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) = \mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_\tau(\Sigma, \mathbb{P}) \preceq_{\mathcal{F}} \mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$. If there exists a local abstract controller $\mathcal{C}^i_{\mathsf{a}} : \mathcal{A}(\mathbb{P}^\varepsilon_i) \to 2^{U_2(\mathbb{P}^\varepsilon_i)}$ such that $\bar{\varphi}^\varepsilon_i$ is satisfied for $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, then there exists a local controller

$$\mathcal{C}_i(x) := \mathcal{C}^i_{\mathsf{a}}(\mathcal{F}(x)), \quad x \in \mathbb{P}_i \quad (29)$$

such that $\bar{\varphi}_i$ is satisfied for $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$.

(2) Let $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i) = \overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_\tau(\Sigma, \mathbb{P}) \simeq_\varepsilon \overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ with the ε-ABR $\mathcal{R}$. If there exists a local

Algorithm 4: Local Abstract Controller ($\texttt{LAC}$)

Input: $\tau, \varepsilon \in \mathbb{R}^+, \mathbb{P}_i \in \mathbb{P}, \mathbf{O} \subset \mathbb{X}, \mathbb{R}(\pi_\varepsilon) \subset \mathbb{X}$
Output: the local abstract controller $\mathcal{C}^i_{\mathsf{a}}$
1 Construct the local symbolic abstraction $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ to guarantee either the FRR or the ε-ABR
2 **if** $i = 0$ **then**
3   Set $\mathbb{P}^\varepsilon_0 \setminus \mathbf{O}$ as the local initial region
4 **else**
5   Set $(\mathbb{P}^\varepsilon_{i-1} \cap \mathbb{P}^\varepsilon_i) \setminus \mathbf{O}$ as the local initial region
6 **for** any element $\mathbb{R}_j(\pi_\varepsilon)$ in $\mathbb{R}(\pi_\varepsilon)$ **do**
7   **if** $\mathbb{P}^\varepsilon_i \cap \mathbb{R}_j(\pi_\varepsilon) \neq \varnothing$ **then**
8   Set $(\mathbb{P}^\varepsilon_i \cap \mathbb{R}_j(\pi_\varepsilon)) \setminus \mathbf{O}$ as the local internal regions
9   **else**
10   No local internal regions
11 **if** $\mathbb{P}_{i+1}$ exists **then**
12   Set $(\mathbb{P}^\varepsilon_i \cap \mathbb{P}^\varepsilon_{i+1}) \setminus \mathbf{O}$ as the local target region
13 **else**
14   No local target region
15 Generate the local LTL formula $\bar{\varphi}^\varepsilon_i$ based on π_ε
16 Design the controller $\mathcal{C}^i_{\mathsf{a}}$ for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}^\varepsilon_i)$ with $\bar{\varphi}^\varepsilon_i$
17 **return** the local abstract controller $\mathcal{C}^i_{\mathsf{a}}$

abstract controller $\mathcal{C}^i_{\mathsf{a}} : \mathcal{A}(\mathbb{P}^\varepsilon_i) \to 2^{U_2(\mathbb{P}^\varepsilon_i)}$ such that $\bar{\varphi}^\varepsilon_i$ is satisfied for $\overline{\mathbf{T}}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, then the local controller

$$\mathcal{C}_i(x) := \mathcal{C}^i_{\mathsf{a}}(\mathcal{R}(x)), \quad x \in \mathbb{P}_i \tag{30}$$

is such that $\bar{\varphi}_i$ is satisfied for $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$.

Proof: Since we focus on the local state space $\mathbb{P}_i$ instead of the whole state space $\mathbb{X}$, the local abstract controller $\mathcal{C}^i_{\mathsf{a}}$ is defined on $\mathcal{A}(\mathbb{P}^\varepsilon_i)$ such that the controllers (29)-(30) are well-defined on $\mathbb{P}_i$. That is, for any $x \in \mathbb{P}_i$, $\mathcal{F}(x) \in \mathcal{A}(\mathbb{P}^\varepsilon_i)$ holds from the FRR, and $\mathcal{R}(x) \in \mathcal{A}(\mathbb{P}^\varepsilon_i)$ holds from the ε-ABR. Next, we can follow [17, Theorem VI.3] and [33, Theorem 5.1] respectively to show that, under the controllers (29)-(30), the local LTL formula $\bar{\varphi}_i$ is satisfied for $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$. ■

From Theorems 4-5, the essence of the controller synthesis is to design all local abstract controllers. Since each local symbolic abstraction has its own local LTL formula and all local LTL formulas are independent with each other, existing algorithms like the fixed-point algorithm [21] and the dynamic programming algorithm [44] can be applied locally to derive local abstract controllers. Therefore, the design of all local abstract controllers is addressed in the following.

For each cell $\mathbb{P}_i$, $i \in \mathbb{L}$, the local abstract controller design is presented in Algorithm 4. First, the local symbolic abstraction $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ is constructed in line 1 to guarantee certain equivalence relation (see also Theorems 2-3). Second, the local LTL formula $\bar{\varphi}^\varepsilon_i$ is formulated in lines 2-15 for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$. In particular, lines 2-5 show the local initial region in two different cases, lines 6-10 determine all local internal regions in $\mathbb{P}_i$, and lines 11-14 provide the local target region in two different cases. All these regions have been explained in detail in Section VI-B, and are used to construct the local LTL formula $\bar{\varphi}^\varepsilon_i$ in line 15 based on the requirements of the accepting path π_ε. Finally, with the local symbolic model $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}^\varepsilon_i)$ and its local LTL formula $\bar{\varphi}^\varepsilon_i$, the local abstract controller is designed in line 16 such that $\bar{\varphi}^\varepsilon_i$ is satisfied for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}^\varepsilon_i)$. Here, we emphasize that the local abstract controller is designed for $\mathcal{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}^\varepsilon_i)$ such that the abstract controller refinement in Theorem 5 can be implemented well.

Algorithm 5: Abstract Controller Design

Input: $\tau, \varepsilon \in \mathbb{R}^+, \mathbb{X}_0 \subseteq \mathbb{X}, \mathbb{R}(\pi_\varepsilon) \subset \mathbb{X}, \mathbf{O} \subset \mathbb{X}$
Output: the union of all local abstract controllers
Initialize: $\mathcal{C}_{\mathsf{a}} = \mathcal{C}^i_{\mathsf{a}} = \varnothing$
1 Cover the state space $\mathbb{X}$ via Algorithm 1
2 Generate the graph $\mathcal{G}$ via Algorithm 3
3 Generalize the graph $\mathcal{G}$ to $\bar{\mathcal{G}}$ by including $\mathbb{X}_0$ and $\mathbb{R}(\pi_\varepsilon)$
4 **if** the path $\mathbb{p}$ in Theorem 1 does not exist **then**
5   Stop and **return** Null
6 **else**
7   Derive the cell sequence $\mathbb{P}$ from the path $\mathbb{p}$
8   **for** all $\mathbb{P}_i \in \mathbb{P}$ **do**
9   Design the local abstract controller $\mathcal{C}^i_{\mathsf{a}} = \texttt{LAC}(\tau, \mathbb{P}_i, \mathbf{O}, \mathbb{R}(\pi_\varepsilon))$ via Algorithm 4
10         $\mathcal{C}_{\mathsf{a}} = \mathcal{C}_{\mathsf{a}} \cup \mathcal{C}^i_{\mathsf{a}}$
11   **return** $\mathcal{C}_{\mathsf{a}}$

Since any two successive cells in $\mathbb{P}$ overlap, the local target region in $\mathbb{P}_i$ will be the local initial region in $\mathbb{P}_{i+1}$, which is shown via lines 5 and 12 in Algorithm 4. Therefore, Algorithm 4 can be implemented iteratively to derive all local abstract controllers. In this respect, the whole control strategy is summarized in Algorithm 5 to solve **Problems 1** and **2**. More precisely, if the path $\mathbb{p}$ from lines 1-4 does not exist, then Algorithm 5 terminates and returns a signal Null to show the unrealization of the accepting path for the LTL specification; otherwise, all local abstract controllers are designed in lines 7-10. Therefore, the following theorem is established directly.

Theorem 6: Consider the system Σ, the obstacle set $\mathbb{O} \subset \mathbb{X}$, and the LTL formula φ with the accepting path π. The realization of π and the satisfaction of φ can be verified from the result of Algorithm 5. That is, Algorithm 5 returns either the signal Null to show the unrealization of π or a sequence of local abstract controllers to guarantee the satisfaction of φ.

Proof: Follows by Theorems 1, 4 and 5. ■

Theorem 6 shows that the satisfaction of the LTL formula φ can be determined via Algorithm 5, which thus solves **Problems 1** and **2**. In particular, all local abstract controllers can be combined as (27), which can be refined as a global controller for the original system via (28)-(30).

VIII. Numerical Example

To illustrate the derived results, a numerical example is presented below. All computations are performed on a desk computer with Intel Core i9-10900K CPU@3.70GHz and 64GB RAM. Consider an autonomous vehicle whose dynamics is of the bicycle model in [45, Example 2.8]. To be specific, the system dynamics Σ of the vehicle is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos(\alpha + x_3)\cos(\alpha)^{-1} \\ u_1 \sin(\alpha + x_3)\cos(\alpha)^{-1} \\ u_1 \tan(u_2) \end{bmatrix}, \tag{31}$$

where $\alpha := \arctan(0.5\tan(u_2))$, $x := (x_1, x_2, x_3) \in \mathbb{R}^3$ is the vehicle state, and $u = (u_1, u_2) \in \mathbb{R}^2$ is the control input. Here, $\mathbf{p} = (x_1, x_2) \in \mathbb{R}^2$ is the vehicle position, $x_3 \in \mathbb{R}$ is the

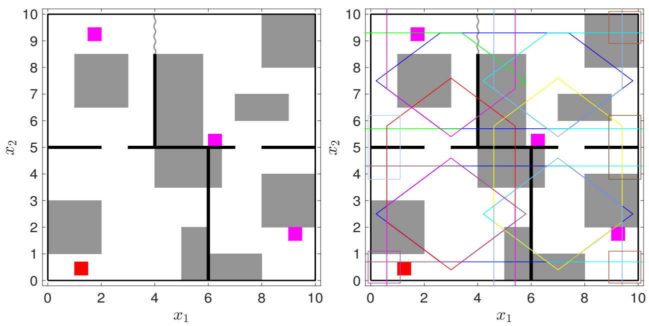

Fig. 7. Illustration of the state space and its cover in the 2-D plane. (Left) The state space in the 2-D plane. The grey regions are obstacles, the red region is the initial region, the black lines are the walls among different rooms, and the grey wave line is the door. The magenta regions are the regions of interest. (Right) The covering of the state space in the 2-D plane.

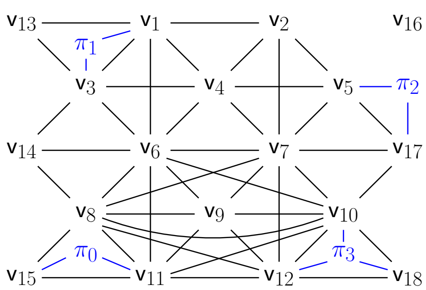

Fig. 8. Illustration of the generated graph. $\mathcal{V} = \{\mathsf{v}_i : i = 1, \ldots, 18\}$ is the vertex set. $\Pi = \{\pi_0, \pi_1, \pi_2, \pi_3\}$ is the symbol set of all regions of interest.

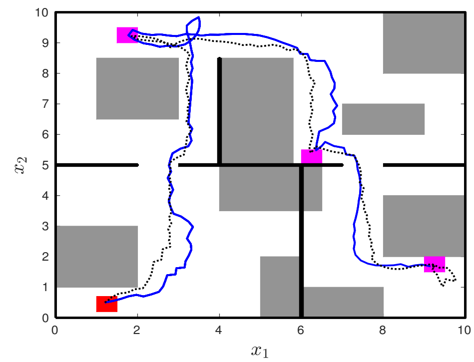

Fig. 9. The state trajectories of the vehicle in the 2-D position space. The blue solid trajectory is generated via the proposed approach, while the black dotted one is based on the global approach.

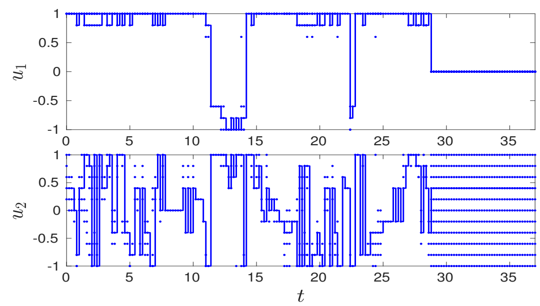

Fig. 10. The control inputs for the satisfaction of the LTL formula φ. The blue dots are the feasible control inputs at each time instant, and the blue lines are the control trajectories to generate the blue position trajectory in Fig. 9.

orientation of the vehicle in the 2-dimensional plane, $u_1 \in \mathbb{R}$ is the rear wheel velocity, and $u_2 \in \mathbb{R}$ is the steering angle. The control input is assumed in $\mathbb{U} = [-1, 1] \times [-1, 1]$.

1) Problem Setup: For the vehicle, the state space is $\mathbb{X} = [0, 10] \times [0, 10] \times [-\boldsymbol{\pi}, \boldsymbol{\pi}]$. In the 2-dimensional plane, the state space consists of 4 rooms, and there exists a door between Rooms 2 and 3 (i.e., the grey wave line in Fig. 7). Initially, the vehicle is placed in the charge station $\mathbb{X}_0 = [1, 1.5] \times [0.2, 0.7] \times [-\boldsymbol{\pi}, \boldsymbol{\pi}]$ of Room 1 (i.e., the red region in Fig. 7) and the velocity is zero. The vehicle is to accomplish the following task: "go to the region $\mathbf{S}_1 = [1.5, 2] \times [9, 9.5] \times [-\boldsymbol{\pi}, \boldsymbol{\pi}]$ in Room 2, then visit the region $\mathbf{S}_2 = [9.5, 10] \times [6, 6.5] \times [-\boldsymbol{\pi}, \boldsymbol{\pi}]$ in Room 3 or $\mathbf{S}_3 = [8.5, 9] \times [0.5, 1] \times [-\boldsymbol{\pi}, \boldsymbol{\pi}]$ in Room 4, and finally stay in the region $\mathbf{S}_3$ while avoiding the obstacle regions in $\mathbb{O}$ (i.e., the black regions in Fig. 7)". This task can be represented formally into the following LTL formula:

$$\varphi = (\neg(\pi_2 \vee \pi_3) \mathsf{U} \pi_1) \wedge \Diamond(\pi_2 \vee \pi_3) \wedge \Box \pi_3, \tag{32}$$

where $\mathbb{R}(\pi_1) = \mathbf{S}_1$, $\mathbb{R}(\pi_2) = \mathbf{S}_2$ and $\mathbb{R}(\pi_3) = \mathbf{S}_3$. For the LTL formula φ, we can implement [35, Algorithm 3] to derive an accepting path $\pi = \pi_0\pi_1\pi_2(\pi_3)^\omega$, where $\mathbb{R}(\pi_0) = \mathbb{X}_0$.

2) Realization of the Accepting Path: To illustrate the realization of the accepting path π, we choose $\epsilon = 0.2$ and the state space is divided into 18 overlapping zonotopes, which is shown in Fig. 7. Due to the existence of the door, we consider two cases to show the necessity of the realization verification.

The first case is that the door between Rooms 2 and 3 is open. In this case, we apply Algorithm 3 to generate the graph, which is shown in Fig. 8. We search the graph in Fig. 8 to find a path to realize the LTL formula. We follow Theorem 1 and the shortest path search algorithm to find a path $\mathbb{p} := \pi_0\mathsf{v}_{11}\mathsf{v}_6\mathsf{v}_1\pi_1\mathsf{v}_2\mathsf{v}_5\pi_2\mathsf{v}_7(\pi_3)^\omega$. In particular, for all $\mathsf{v}_k \in \mathbb{p}$, $\mathbb{R}(\mathsf{v}_k) \setminus \mathbb{O} \neq \varnothing$, which shows the validity of item (ii) of Theorem 1. With the region $\mathbb{P}$ from the path $\mathbb{p}$, we follow Section VI-B to construct all the local LTL formulas. For instance, in the region $\mathbb{R}(\mathsf{v}_{11})$, the local LTL formula is $\bar{\varphi}_1 = \bar{\varphi}_{11} \wedge \bar{\varphi}_{12}$ with $\bar{\varphi}_{11} = \Box(x \in \mathbb{R}(\mathsf{v}_{11}))$ and $\bar{\varphi}_{12} = \Diamond(x \in \mathbb{R}(\mathsf{v}_{11}) \cap \mathbb{R}(\mathsf{v}_6))$. Note that no local internal regions exist in $\mathbb{R}(\mathsf{v}_{11})$ and thus $\bar{\varphi}_{13}$ does not exist. Similarly, we derive the local LTL formulas for all cells in $\mathbb{P}$, and combine them into $\bar{\varphi}_0 \wedge \bar{\varphi}_1 \wedge \ldots \wedge \bar{\varphi}_6$.

The second case is that the door is closed. From Fig. 8, the path $\mathbb{p}$ in the first case is available to validate item (i) of Theorem 1. However, the cell $\mathbb{R}(\mathsf{v}_1)$ is not connected and no subregion in $\mathbb{R}(\mathsf{v}_1)$ exists such that item (ii-b) of Theorem 1 is guaranteed. Hence, the accepting path π is not realized and there is no need for the controller design in this case.

From these two cases, we can see that the output of Algorithm 5 is Null in the second case. Hence, there is no need for the controller design, and the non-existence of the controller comes from the unrealization of π. Different from the classic abstraction-based approach [8], [9], [13], [17] without the realization verification, the proposed approach reveals the real reason for the non-existence of the controller. Due to the realization verification, the costs of the abstraction construction and controller design can be avoided. From these two cases, we show the necessity of the realization verification.

3) Controller Design: In the following, we only consider the first case and design the controller such that the LTL formula φ is satisfied. Let $\tau = 0.2$ and $\varepsilon = \epsilon = 0.2$. In order

TABLE I
COMPARISON OF TRANSITION NUMBERS AND RUN TIMES

	Transition numbers	$\mathsf{t}_{\mathsf{abs}}$ (s)	$\mathsf{t}_{\mathsf{con}}$ (s)
$\mathbb{R}(\mathsf{v}_{11})$	$3.38112 \cdot 10^7$	102.30	44.13
$\mathbb{R}(\mathsf{v}_6)$	$5.34214 \cdot 10^7$	291.91	99.10
$\mathbb{R}(\mathsf{v}_1)$	$2.18706 \cdot 10^7$	79.82	56.44
$\mathbb{R}(\mathsf{v}_2)$	$3.35502 \cdot 10^7$	107.57	49.14
$\mathbb{R}(\mathsf{v}_7)$	$3.57037 \cdot 10^7$	134.82	48.06
$\mathbb{R}(\mathsf{v}_{10})$	$3.69582 \cdot 10^7$	134.18	54.00
Global approach	$2.17682 \cdot 10^8$	1425.18	6015.38

to compare with the global approach, we follow the toolbox SCOTS [21] to approximate the state space and to construct the symbolic models $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$ for $i \in \{1, \ldots, 6\}$. Let $\eta_i = 0.2$, and the input space $\mathbb{U}$ is approximated into $U_2 := \{(0.2j, 0.2k) \in \mathbb{R}^2 : j, k \in \{-5, \ldots, 5\}\}$. Let $\mu_1 = \mu_2 = \mu_4 = 0.16, \mu_3 = \mu_5 = 0.18$ and $\mu_6 = 0.15$. Hence, all local symbolic models are constructed and all local abstract controllers can be designed via Algorithm 4 to satisfy the local LTL formulas. Fig. 9 shows the position trajectories of the vehicle via the proposed and global approaches respectively, and thus implies that the global LTL formula is satisfied. In particular, $\mu = 0.15$ for the global approach; otherwise, the LTL formula cannot be satisfied. The control inputs are given in Fig. 10, and the vehicle stays in $\mathbf{S}_3$ since 28.8 seconds.

The comparison between the proposed approach and the global approach (e.g., [9], [17]) is shown in Table I, where $\mathsf{t}_{\mathsf{abs}}$ and $\mathsf{t}_{\mathsf{con}}$ are respectively the computation times (in seconds) of the abstraction construction and control synthesis. We can see significant decreases in the computation times and transition numbers due to our proposed approach. Since each cell is a subset of the state space and different cells are allowed to have their own partition parameters, the number of transition relations in each cell is much smaller than that in the whole state space. Note that φ in (32) is not co-safe, and thus we only need to construct a fine local symbolic model for $\mathbb{R}(\mathsf{v}_{10})$ instead of for the whole state space, which results in the decrease of the computation time for the controller synthesis. For the co-safe LTL formula $\varphi' = (\neg(\varphi_2 \vee \varphi_3)\mathsf{U}\varphi_1) \wedge \Diamond(\varphi_2 \vee \varphi_3) \wedge \Diamond\varphi_3$, the global approach can be applied with the relaxed parameter $\mu = \varepsilon$. In this case, the computation times are still much smaller since we only focus on six cells.

IX. CONCLUSION

In this paper we studied the controller synthesis problem for nonlinear control systems with linear temporal logical specifications. A zonotope-based covering of the state space was proposed, which allows the intersection between neighbour cells and results in a graph among all cells. Further, a graph-based method was developed to verify the realization of the accepting paths for LTL specifications, which results in finite local LTL formulas via decomposition techniques. To ensure the satisfaction of LTL formulas, we discussed the relations between local and global controllers, and proposed a local-to-global control synthesis strategy. Future work will be devoted to the extension of the proposed approach to the case of multi-robot systems and the case where other temporal logic tasks like signal temporal logic are involved.

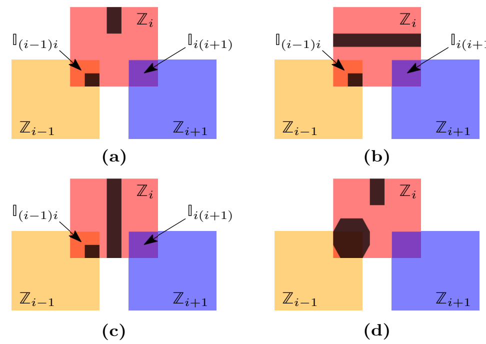

Fig. 11. Illustration of different cases in the proof of Theorem 1. The black regions are obstacles. **(a)** $\mathbb{P}_i \setminus \mathbf{O}$ is connected and item (ii-a) is satisfied. **(b)** $\mathbb{P}'_i$ exists and item (ii-b) is satisfied. **(c)** $\mathbb{P}_i$ is not connected and $\mathbb{P}'_i$ does not exist, which imply that item (ii-b) is not satisfied. **(d)** The intersection region $\mathbb{P}_i \cap \mathbb{P}_{i-1}$ is not admissible and the edge between $\mathbb{P}_i$ and $\mathbb{P}_{i-1}$ does not exist, which shows that item (i) is not satisfied.

APPENDIX A
PROOF OF THEOREM 1

First, from Definition 7, the realization of the robust accepting path π_ε implies there exists a connected region $\mathbb{X} \subset \mathbb{X} \setminus \mathbb{O}$ intersecting with all components in $\mathbb{R}(\pi_\varepsilon)$. The existence of the connected region $\mathbb{X}$ implies the existence of the path $\mathbb{p}$ in $\bar{\mathcal{G}}$ such that π_ε is realized, which shows the validity of item (i). To show item (ii), we consider two cases. The first case is that the region $\mathbb{P}$ from the path $\mathbb{p}$ is connected; see Fig. 11(a). In this case, it readily follows that item (ii-a) holds. The second case is that the region $\mathbb{P}$ is not connected; see Fig. 11(b). In this case, $\mathbb{X} \subset \mathbb{P}$, and for each $\mathbb{P}_i \in \mathbb{P}$, $\mathbb{X} \cap \mathbb{P}_i$ is connected, which implies the existence of $\mathbb{P}'_i \subset \mathbb{P}^\varepsilon_i \setminus \mathbf{O}$ in item (ii-b) and further $\mathbb{P}'_i \cup \mathbb{I}^\varepsilon_{(i-1)i} \cup \mathbb{I}^\varepsilon_{i(i+1)}$ is connected. Since $\mathbb{X}$ intersects with all components in $\mathbb{R}(\pi_\varepsilon)$, we have $\mathbb{P}'_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$ if $\mathbb{P}^\varepsilon_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$. Therefore, items (i)-(ii) are satisfied. In particular, if item (ii-b) does not hold, then the case in Fig. 11(c) may exist, which contradicts with the realization of π_ε.

Second, we consider the realization of π_ε from items (i)-(ii). If item (i) is satisfied, then for each pair $(\pi^\varepsilon_k, \pi^\varepsilon_{k+1})$ from the accepting path π_ε, there exists a sub-path in $\mathbb{p}$ such that π^ε_k and π^ε_{k+1} are connected, which in turn shows the case in Fig. 11(d) is not feasible. In the following, we show the existence of the connected region $\mathbb{X}$ in the region $\mathbb{P}$ from the path $\mathbb{p}$. From item (ii-a), if $\mathbb{P}^\varepsilon_i \setminus \mathbf{O}$ is connected for each component $\mathbb{P}_i \in \mathbb{P}$ (see Fig. 11(a)), then the region $\mathbb{P}$ is connected and we can set $\mathbb{X} = \mathbb{P}^\varepsilon \setminus \mathbf{O}$ directly. Otherwise, from item (ii-b), there exists a connected subregion $\mathbb{P}'_i \subset \mathbb{P}_i \setminus \mathbf{O}$ intersecting with both $\mathbb{I}^\varepsilon_{(i-1)i}$ and $\mathbb{I}^\varepsilon_{i(i+1)}$; see Fig. 11(b). In addition, if $\mathbb{P}^\varepsilon_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$, then $\mathbb{P}'_i \cap \mathbb{R}(\pi_\varepsilon) \neq \varnothing$. Hence, there exists a connected subregion in $\mathbb{P} \setminus \mathbf{O}$ intersecting with all components in $\mathbb{R}(\pi_\varepsilon)$, which further implies the realization of π_ε.

APPENDIX B
PROOF OF THEOREM 2

From the definitions of $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, one has $U_2(\mathbb{P}_i) \subseteq U_1(\mathbb{P}_i)$. Let $(x_1, q_1) \in \mathcal{F}$ with $x_1 \in \mathbb{P}_i$ and

$q_1 \in X_{2i}$, and $\|x_1 - q_1\|_{\mathbf{G}} \leq \varepsilon$ holds from (23). For each $u \in U_2(q_1)$, we obtain $u \in U_2(q_1) \subseteq U_2(\mathbb{P}_i) \subseteq U_1(\mathbb{P}_i)$, and $\Delta_{2i}(q_1, u) \neq \varnothing$ holds from the definition of $U_2(q_1)$. If $\Delta_1(x_1, u) = \varnothing$, then $u \notin U_1(x_1)$, which further implies $u \notin U_1$ and thus results in a contradiction. Hence, $\Delta_1(x_1, u) \neq \varnothing$ and $u \in U_1(x_1)$. We conclude that $U_2(q_1) \subseteq U_1(x_1)$.

Let $q_1, q_2 \in X_{2i}$ and $u \in U_2(q_1)$. Since $(x_1, q_1) \in \mathcal{F}$, $\|x_1 - q_1\|_{\mathbf{G}} \leq \varepsilon$. If $\|\Delta_1(x_1, u) - q_2\|_{\mathbf{G}} \leq \varepsilon$, then there exists $x_2 := \mathbf{x}(\tau, x_1, u) \in \mathbb{P}_i$ such that $\|x_2 - q_2\|_{\mathbf{G}} \leq \varepsilon$ from the approximation of $\mathbb{P}_i$. That is, $(x_2, q_2) \in \mathcal{F}$. Since f in (2) satisfies the local Lipschitz condition, one has $\|\mathbf{x}(\tau, x_1, u) - \mathbf{x}(\tau, q_1, u)\|_{\mathbf{G}} \leq e^{L_i\tau}\|x_1 - q_1\|_{\mathbf{G}} \leq e^{L_i\tau}\varepsilon$ with the local Lipschitz constant $L_i > 0$. Hence,

$$\begin{aligned}\|q_2 - \mathbf{x}(\tau, q_1, u)\|_{\mathbf{G}} &\leq \|\mathbf{x}(\tau, x_1, u) - \mathbf{x}(\tau, q_1, u)\|_{\mathbf{G}} \\ &\quad + \|\mathbf{x}(\tau, x_1, u) - q_2\|_{\mathbf{G}} \\ &\leq e^{L_i\tau}\|x_1 - q_1\|_{\mathbf{G}} + \varepsilon \\ &\leq (1 + e^{L_i\tau})\varepsilon,\end{aligned}$$

which implies $q_2 \in \Delta_{2i}(q_1, u)$ from the construction of $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$. Therefore, the proof is completed.

Appendix C
Proof of Theorem 3

We show that the relation $\mathcal{R} := \{(x, q) \in X_{1i} \times X_{2i} : \|x - q\|_{\mathbf{G}} \leq \varepsilon\}$ is an ε-ABR between $\mathbf{T}_\tau(\Sigma, \mathbb{P}_i)$ and $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$. The relation $\mathcal{R}$ implies that $\mathcal{R}(X_{1i}) = X_{2i}$, and further $\mathbb{P}_i \subseteq \cup_{q \in X_{2i}}\{y \in \mathbb{R}^n : \|y - q\|_{\mathbf{G}} \leq 0.5\mu_i\}$ from the geometrical consideration. For any $x_1 \in \mathbb{P}_i$, there exists $q_1 \in X_{2i}$ such that $\|x_1 - q_1\|_{\mathbf{G}} \leq 0.5\mu_i \leq \varepsilon$ from (25).

Consider $(x_1, q_1) \in \mathcal{R}$ with $x_1 \in \mathbb{P}_i$ and $u_1 \in U_1(\mathbb{P}_i)$ such that $x_2 = \Delta_1(x_1, u_1) \in \mathbb{P}_i$. Let $\bar{x} := \mathbf{x}(\tau, q_1, u_1) \in \mathbb{P}_i$. If $\max_{l \in \boldsymbol{\ell}}\{\|g_l\|\} \leq \mu_i$, then there exists $v \in \mathcal{A}(\mathbb{P}_i)$ such that

$$\|\bar{x} - v\|_{\mathbf{G}} = \|\mathbf{x}(\tau, q_1, u_1) - v\|_{\mathbf{G}} \leq 0.5\mu_i, \tag{C.1}$$

which implies that $\bar{x} \in \mathsf{Reach}(\tau, q_1)$ and $v \in \mathbb{S}_\eta(\tau, q_1)$. Given any $u_2 \in U_2(\mathbb{P}_i)$ such that $u_2 = \psi(v)$, there exists $w := \mathbf{x}(\tau, q_1, u_2) \in \mathbb{P}_i$ such that

$$\|w - v\|_{\mathbf{G}} = \|\mathbf{x}(\tau, q_1, u_2) - v\|_{\mathbf{G}} \leq 0.5\eta_i. \tag{C.2}$$

Since $\mathbb{P}_i \subseteq \cup_{q \in X_{2i}}\{y \in \mathbb{R}^n : \|y - q\|_{\mathbf{G}} \leq 0.5\mu_i\}$, there exists $q_2 \in X_2$ such that

$$\|q_2 - w\|_{\mathbf{G}} = \|q_2 - \mathbf{x}(\tau, q_1, u_2)\|_{\mathbf{G}} \leq 0.5\mu_i. \tag{C.3}$$

Let $q_2 = \Delta_2(q_1, u_2)$. Since the system Σ satisfies (24), we yield from (C.1)-(C.3) that

$$\begin{aligned}\|x_2 - q_2\|_{\mathbf{G}} &\leq \|\Delta_1(x_1, u_1) - \Delta_2(q_1, u_2)\|_{\mathbf{G}} \\ &\leq \|\Delta_1(x_1, u_1) - \bar{x}\|_{\mathbf{G}} + \|\bar{x} - \Delta_2(q_1, u_2)\|_{\mathbf{G}} \\ &= \|\mathbf{x}(\tau, x_1, u_1) - \mathbf{x}(\tau, q_1, u_1)\|_{\mathbf{G}} \\ &\quad + \|\bar{x} - \Delta_2(q_1, u_2)\|_{\mathbf{G}} \\ &\leq \beta(\|x_1 - q_1\|_{\mathbf{G}}, \tau) \\ &\quad + \|\bar{x} - v + v - w + w - \Delta_2(q_1, u_2)\|_{\mathbf{G}} \\ &\leq \beta(\varepsilon, \tau) + 0.5\eta_i + \mu_i.\end{aligned}$$

Therefore, we conclude from (26) that $(x_2, q_2) \in \mathcal{R}$.

Consider any $(x_1, q_1) \in \mathcal{R}$ with $x_1 \in \mathbb{P}_i$. Choose any $u_2 \in U_2(\mathbb{P}_i)$ such that $q_2 = \Delta_2(q_1, u_2) \in \mathbb{P}_i$. From the construction of $\mathbf{T}_{\tau,\mu_i,\eta_i}(\Sigma, \mathbb{P}_i)$, $\|q_2 - \mathbf{x}(t, q_1, u_2)\|_{\mathbf{G}} \leq 0.5\mu_i$. Pick $u_1 = u_2$ and let $x_2 = \Delta_1(x_1, u_1) \in \mathbb{P}_i$. From (24), we have

$$\begin{aligned}\|x_2 - q_2\|_{\mathbf{G}} &= \|x_2 - \mathbf{x}(t, q_1, u_2) + \mathbf{x}(t, q_1, u_2) - q_2\|_{\mathbf{G}} \\ &\leq \|x_2 - \mathbf{x}(t, q_1, u_2)\|_{\mathbf{G}} + 0.5\mu_i \\ &\leq \beta(\varepsilon, \tau) + 0.5\mu_i,\end{aligned}$$

which shows $(x_2, q_2) \in \mathcal{R}$ from (26), and completes the proof.

References

[1] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT press, 2008.

[2] E. M. Wolff, U. Topcu, and R. M. Murray, "Optimization-based trajectory generation with linear temporal logic specifications," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2014, pp. 5319–5325.

[3] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of reactive (1) designs," *Journal of Computer and System Sciences*, vol. 78, no. 3, pp. 911–938, 2012.

[4] E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem *et al.*, *Handbook of Model Checking*. Springer, 2018, vol. 10.

[5] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, "TuLiP: A software toolbox for receding horizon temporal logic planning," in *Int. Conf. Hybrid Syst.: Comput. Control*, 2011, pp. 313–314.

[6] P. J. Antsaklis, M. Lemmon, and J. A. Stiver, "Hybrid system modeling and event identification," *Technical Report of the ISIS Group at the University of Notre Dame ISIS-93-002*, vol. 93, pp. 366–392, 1993.

[7] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer Science & Business Media, 2009.

[8] A. Girard, "Controller synthesis for safety and reachability via approximate bisimulation," *Automatica*, vol. 48, no. 5, pp. 947–953, 2012.

[9] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[10] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Automat. Contr.*, vol. 57, no. 7, pp. 1804–1809, 2012.

[11] W. Ren and D. V. Dimarogonas, "Logarithmic quantization based symbolic abstractions for nonlinear control systems," in *European Control Conference*. IEEE, 2019, pp. 1312–1317.

[12] G. Pola, P. Pepe, M. D. Di Benedetto, and P. Tabuada, "Symbolic models for nonlinear time-delay systems using approximate bisimulations," *Systems & Control Letters*, vol. 59, no. 6, pp. 365–373, 2010.

[13] W. Ren and D. V. Dimarogonas, "Symbolic abstractions for nonlinear control systems via feedback refinement relation," *Automatica*, vol. 114, p. 108828, 2020.

[14] A. Girard, G. Gössler, and S. Mouelhi, "Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models," *IEEE Trans. Automat. Contr.*, vol. 61, no. 6, pp. 1537–1549, 2016.

[15] M. Zamani, M. Mazo Jr, M. Khaled, and A. Abate, "Symbolic abstractions of networked control systems," *IEEE Trans. Control. Netw. Syst.*, vol. 5, no. 4, pp. 1622–1634, 2017.

[16] M. Zamani, A. Abate, and A. Girard, "Symbolic models for stochastic switched systems: A discretization and a discretization-free approach," *Automatica*, vol. 55, pp. 183–196, 2015.

[17] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Trans. Automat. Contr.*, vol. 62, no. 4, pp. 1781–1796, 2017.

[18] E. S. Kim, M. Arcak, and S. A. Seshia, "Symbolic control design for monotone systems with directed specifications," *Automatica*, vol. 83, pp. 10–19, 2017.

[19] M. Mazo, A. Davitian, and P. Tabuada, "Pessoa: A tool for embedded controller synthesis," in *Int. Conf. Comput. Aided Verification*. Springer, 2010, pp. 566–569.

[20] S. Mouelhi, A. Girard, and G. Gössler, "CoSyMA: A tool for controller synthesis using multi-scale abstractions," in *Int. Conf. Hybrid Syst.: Comput. Control*, 2013, pp. 83–88.

[21] M. Rungger and M. Zamani, "SCOTS: A tool for the synthesis of symbolic controllers," in *Int. Conf. Hybrid Syst.: Comput. Control*. ACM, 2016, pp. 99–104.

[22] J. Tumova, B. Yordanov, C. Belta, I. Černá, and J. Barnat, "A symbolic approach to controlling piecewise affine systems," in *Proc. IEEE Conf. Decis. Control.* IEEE, 2010, pp. 4230–4235.

[23] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.

[24] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *Int. Conf. Hybrid Syst.: Comput. Control*. Springer, 2005, pp. 291–305.

[25] I. M. Mitchell, J. Budzis, and A. Bolyachevets, "Invariant, viability and discriminating kernel under-approximation via zonotope scaling," in *Int. Conf. Hybrid Syst., Comput. Control*, 2019, pp. 268–269.

[26] W. Ren, J. Calbert, and R. Jungers, "Zonotope-based controller synthesis for LTL specifications," in *IEEE Conf. Decis. Control*. IEEE, 2021, pp. 580–585.

[27] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, 2007.

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

[29] 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.

[30] E. A. Gol and C. Belta, "Time-constrained temporal logic control of multi-affine systems," *Nonlinear Anal. Hybrid Syst.*, vol. 10, pp. 21–33, 2013.

[31] P.-J. Meyer and D. V. Dimarogonas, "Hierarchical decomposition of LTL synthesis problem for nonlinear control systems," *IEEE Trans. Automat. Contr.*, vol. 64, no. 11, pp. 4676–4683, 2019.

[32] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Automat. Contr.*, vol. 5, no. 52, pp. 782–798, 2007.

[33] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Trans. Automat. Contr.*, vol. 53, no. 6, pp. 1406–1418, 2008.

[34] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Trans. Automat. Contr.*, vol. 55, no. 1, pp. 116–126, 2010.

[35] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local LTL specifications," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 218–235, 2015.

[36] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck, "Lazy abstraction-based control for safety specifications," in *IEEE Conf. Decis. Control*. IEEE, 2018, pp. 4902–4907.

[37] L. Lindemann, J. Nowak, L. Schönbächler, M. Guo, J. Tumova, and D. V. Dimarogonas, "Coupled multi-robot systems under linear temporal logic and signal temporal logic tasks," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 858–865, 2019.

[38] B. S. Rego, G. V. Raffo, J. K. Scott, and D. M. Raimondo, "Guaranteed methods based on constrained zonotopes for set-valued state estimation of nonlinear discrete-time systems," *Automatica*, vol. 111, p. 108614, 2020.

[39] N. Kochdumper and M. Althoff, "Representation of polytopes as polynomial zonotopes," *arXiv preprint arXiv:1910.07271*, 2019.

[40] W. Ren and D. V. Dimarogonas, "Dynamic quantization based symbolic abstractions for nonlinear control systems," in *IEEE Conf. Decis. Control*. IEEE, 2019, pp. 4343–4348.

[41] P. McMullen, "Space tiling zonotopes," *Mathematika*, vol. 22, no. 2, pp. 202–211, 1975.

[42] B. Kabi, "Synthesizing invariants: A constraint programming approach based on zonotopic abstraction," Ph.D. dissertation, Institut Polytechnique de Paris, 2020.

[43] J. Richter-Gebert, "Line arrangements and zonotopal tilings: A little printer exercise," *Hyper-Space*, vol. 2, pp. 8–17, 1993.

[44] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena scientific Belmont, MA, 2005, vol. 1, no. 3.

[45] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.

PLACE PHOTO HERE

Wei Ren received his B.Sci. degree from Hubei University, China, and his Ph.D. degree from the University of Science and Technology of China, China, in 2011 and 2018. He was a visiting student at the University of Melbourne, Australia, from 2017 to 2018. From 2018 to 2022, he was a postdoctoral fellow at KTH Royal Institute of Technology, Stockholm, Sweden and Université Catholique de Louvain, Belgium. He currently is a Professor at Dalian University of Technology, Dalian, China. His research interests include Multi-Agent Systems, Networked Control, Formal Methods, and Hybrid Systems.

PLACE PHOTO HERE

Raphaël M. Jungers is an FNRS Professor at UCLouvain, Belgium. His main interests lie in the fields of Computer Science, Graph Theory, Optimization and Control. He received a Ph.D. in Mathematical Engineering from UCLouvain (2008), and a M.Sc. in Applied Mathematics, both from the EcoleCentrale Paris, (2004), and from UCLouvain (2005). He has held various invited positions, at the Université Libre de Bruxelles (2008-2009), at the Laboratory for Information and Decision Systems of the Massachusetts Institute of Technology (2009-2010), at the University of L'Aquila (2011, 2013, 2016), at the University of California Los Angeles (2016-2017), and at the University of Oxford (2022-2023). He is an FNRS, BAEF, and Fulbright fellow. He has been an Editor at large for IEEE CDC, Associate Editor for the IEEE CSS Conference Editorial Board, and the journals NAHS, Systems and Control Letters, and IEEE Transactions on Automatic Control. He was the recipient of the IBM Belgium 2009 award and a finalist of the ERCIM Cor Baaten award 2011. He was the co-recipient of the SICON best paper award 2013-2014.

PLACE PHOTO HERE

Dimos V. Dimarogonas was born in Athens, Greece, in 1978. He received the Diploma in Electrical and Computer Engineering in 2001 and the Ph.D. in Mechanical Engineering in 2007, both from National Technical University of Athens (NTUA), Greece. Between 2007 and 2010, he held postdoctoral positions at the KTH Royal Institute, Stockholm Sweden, and the Laboratory for Information and Decision Systems (LIDS), MIT, Boston USA. He is currently a Professor and Head at the Division of Decision and Control, KTH Royal Institute of Technology, Stockholm, Sweden. His current research interests include Multi-Agent Systems, Hybrid Systems and Control, Robot Navigation and Networked Control. He serves in the Editorial Board of Automatica, and the IEEE Transactions on Control of Network Systems. He is an IEEE Fellow.