Energy-aware networked control systems under temporal logic specifications

Kazumune Hashimoto, Shuichi Adachi, and Dimos V. Dimarogonas

***Abstract*— In recent years, event and self-triggered control have been proposed as energy-aware control strategies to expand the life-time of battery powered devices in Networked Control Systems (NCSs). In contrast to the previous works in which their control objective is to achieve stability, this paper presents a novel energy-aware control scheme for achieving *high level specifications*, or more specifically, *temporal logic specifications*. Inspired by the standard hierarchical strategy that has been proposed in the field of formal control synthesis paradigm, we propose a new abstraction procedure for jointly synthesizing control and communication strategies, such that the communication reduction in NCSs and the satisfaction of the temporal logic specifications are guaranteed. The benefits of the proposal are illustrated through a numerical example.**

I. Introduction

The increased popularity of introducing Networked Control Systems (NCSs) where sensors, actuators, and controllers are spatially distributed over communication network, have recently suggested the use of *event-triggered* and *self-triggered control* [1], [2]. In the control strategies, sensor measurements are transmitted to the controller only when they are needed based on the violation of prescribed control performances. Applying these strategies is useful, since they have the potential to save energy expenditures by alleviating the communication load for NCSs. Some experimental validations have also been provided, e.g., in [3].

In this paper, we present a new aperiodic formulation for NCSs, which allows them to achieve not only the communication reduction, but also the satisfaction of *high level (temporal) specifications*. To illustrate the proposal more in detail, note first that the control objective in the previous aperiodic formulations is only *stabilization* or *tracking to a given reference*. However, considering the fact that the introduction of NCSs allows us to construct more complex, advanced control architectures as exemplified by automonous robots, traffic systems and so on, achieving the above control objectives may not be sufficient. For instance, consider *cloud robotics* [4], where autonomous robots such as UAVs and warehouse robots interact with the remote operators for supporting various motion tasks and missions. Although their motion tasks given by operators may be to achieve only stability (e.g., "go to the region A"), it is sometimes required to achieve more complex tasks involving *temporal* ones, such as such as sequential tasks (e.g., "visit the region A, *and then* B, *and then* C"), recurrence tasks with obstacle avoidance (e.g.,"survey the region A, B, C, D (in any order) *infinitely often*, while *always* avoiding the region E"), and other complex tasks with temporal properties (e.g., "visit A, B, C, D, and make sure to avoid D *unless* A is visited"). Therefore, it may be of great importance to design the event (self)-triggred strategies for accomodating such high level specifications.

The goal of this paper is to jointly synthesize a *communication strategy* such that the communication reduction is achieved for NCSs, and a *control strategy* such that the above high level missions can be achieved. In particular, we consider that the specifications are expressed by Linear Temporal Logic (LTL) formula [5], and we employ the hierarchical approach that has been proposed in field of formal control synthesis paradigm, see, e.g., [6]–[8]. First, we construct a finite transition system that represents an *abstracted* behavior of the original control system. The transition system is constructed through *reachability analysis*, which evaluates if the state can be steered between each pair of the regions of interest. One of the main contributions of this paper is to provide a new reachability analysis such that the communication strategies can be designed after this procedure. The proposed approach mainly consists of the two steps; *trajectory generation* and *tube generation*. In the trajectory generation, a nominal state trajectory is generated by implementing existing strategies, such as sampling-based algorithms [9]. The trajectory is utilized as a reference that the actual trajectory should follow to achieve the entrance to the target region. In the tube generation, we seek to generate a sequence of tubes (polytopes) among the regions of interest. Since the generated tubes represent safety margins to guarantee reachability, we will make use of them as criteria to analyze reachability. Based on the reachability analysis, we next propose a self-triggered strategy that the controller iteratively assigns communication times by evaluating the self-triggered condition. The self-triggered condition is derived by utilizing tubes obtained by the reachability analysis, and the communication is triggered only when the state trajectory is about to violate the satisfaction of the formula.

The rest of the paper is organized as follows. We describe some preliminaries and the problem formulation in Section II and III, respectively. In Section IV, reachability analysis and an algorithm to obtain a finite transition system are given.

Kazumune Hashimoto is with Department of Applied Physics and Physico-Informatics, Keio University, Yokohama, Japan. His work is supported by Grant-in-Aid for JSPS Research Fellow (Grant Number: 17J05743).

Shuichi Adachi are with Department of Applied Physics and Physico-Informatics, Keio University, Yokohama, Japan.

Dimos V. Dimarogonas is with the ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. His work was supported by the Swedish Research Council (VR), Knut och Alice Wallenberg foundation (KAW), and the H2020 ERC Starting Grant BUCOPHSYS.

In Section V, we propose the implementation algorithm involving both high and low level strategies. In Section VI, a simulation example is given to validate the effectiveness of the proposed approach. We finally conclude in Section VII.

Notations. Let $\mathbb{R}_+$, $\mathbb{N}$, $\mathbb{N}_+$, $\mathbb{N}_{a:b}$ be the positive real, non-negative, positive integers, and the set of integers in the interval $[a, b]$, respectively. For vectors $v_1, \ldots, v_N$, denote by $\mathrm{co}\{v_1, \ldots, v_N\}$ their *convex hull*. For two given sets $A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^n$, denote by $A \oplus B$ the *Minkowski sum* $A \oplus B = \{z \in \mathbb{R}^n \mid \exists x \in A, y \in B : z = x + y\}$ and by $A \ominus B$ the *Pontryagin difference* $A \ominus B = \{x \in \mathbb{R}^n \mid x + y \in A,\ \forall y \in B\}$. We denote the collection of N sets $\mathcal{X}_1, \ldots, \mathcal{X}_N \subset \mathbb{R}^n$ as $\mathcal{X}_{1:N} = \{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$.

II. Preliminaries

In this section, we provide some preliminaries on *finite transition systems* and *Linear Temporal Logic (LTL)* formulas. A finite transition system is a tuple $\mathcal{T} = (S, s_{init}, \delta, \Pi, g)$, where S is a set of states, $s_{init} \in S$ is an initial state, $\delta \subseteq S \times S$ is a transition relation, Π is a set of atomic propositions, and $g : S \to 2^{\Pi}$ is a labeling function. A path of $\mathcal{T}$ is an infinite sequence of states $s_{seq} = s_0 s_1 s_2 \cdots$, such that $s_0 = s_{init}$, $(s_i, s_{i+1}) \in \delta$, $\forall i \in \mathbb{N}$. A trace of a path $s_{seq} = s_0 s_1 s_2 \cdots$ is given by $\mathrm{trace}(s_{seq}) = g(s_0) g(s_1) g(s_2) \cdots$. Linear Temporal Logic (LTL) is defined over a set of atomic propositions Π and is useful to express various task specifications involving temporal properties. The LTL formulas is given according to the following grammer: $\phi ::= \mathrm{true} \mid \pi \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \phi_1 \mathsf{U} \phi_2$, where $\pi \in \Pi$ is the atomic proposition, $\wedge$ (and), $\neg$ (negation) are the boolean operators, and U (until) is the temporal operator. We can also derive other useful temporal operators, such as $\Box$ (always), $\Diamond$ (eventually), and $\Longrightarrow$ (implication). For the LTL semantics, see Chapter 5 in [5] for details. For a given path s_{seq} of $\mathcal{T}$, we denote $\mathrm{trace}(s_{seq}) \models \phi$ if and only if $\mathrm{trace}(s_{seq})$ satisfies ϕ. The path s_{seq} is called accepting if and only if $\mathrm{trace}(s_{seq}) \models \phi$.

III. Problem formulation

A. Plant dynamics

We consider a Networked Control System illustrated in Fig. 1, where the plant and the controller are connected over communication channels. The controller system is responsible for both implementing a high level planner that generates a symbolic plan for a given LTL specification ϕ, and a low level planner that generates feedback controllers to operate the plant. How these planners are designed will be formally given later in this paper. The dynamics of the plant are given by the following Linear-Time-Invariant (LTI) systems:

$$x_{k+1} = A x_k + B u_k + w_k, \tag{1}$$

for $k \in \mathbb{N}$, where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control variable, and $w_k \in \mathbb{R}^n$ is the additive disturbance. We assume that the control and the disturbance variables are constrained in the sets $\mathcal{U}$ and $\mathcal{W}$, i.e.,

$$u_k \in \mathcal{U}, \quad w_k \in \mathcal{W} \tag{2}$$

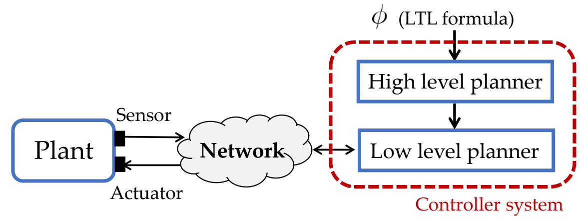


Fig. 1. Networked Control System.

for all $k \in \mathbb{N}$. Here, $\mathcal{U}$ and $\mathcal{W}$ are both polytopic sets containing the origin in their interiors. Regarding the state, we assume $x_k \in \mathcal{X}$, $\forall k \in \mathbb{N}$, where $\mathcal{X}$ is a polygonal set that can be either a convex or non-convex region. The set $\mathcal{X}$ represents the *free space*, in which the state is allowed to move. Inside $\mathcal{X}$, we assume that there exist in total $N_I \in \mathbb{N}_+$ number of polytopic regions $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{N_I} \subset \mathcal{X}$, which represent the *regions of interest* in $\mathcal{X}$. These regions are assumed to be disjoint, i.e., $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, $\forall i, j \in \mathbb{N}_{1:N_I}$ with $i \neq j$. Moreover, let $x_{cent,i} \in \mathcal{R}_i$, $i \in \mathbb{N}_{1:N_I}$ denote the Chebyshev center [10] of the polytope $\mathcal{R}_i$. The Chebyshev center is the center of the maximum ball that is included in $\mathcal{R}_i$ and is obtained by solving a linear program (for details, see Section 5.4.5 in [10]). For simplicity, the initial state is assumed to be inside one of the regions of interest, i.e., $x_0 \in \mathcal{R}_{init}$ with $\mathcal{R}_{init} \in \mathcal{R}$, where $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{N_I}\}$.

Let π_i, $i \in \mathbb{N}_{1:N_I}$ be the atomic proposition assigned to the region $\mathcal{R}_i$. Namely, π_i holds true if and only if $x \in \mathcal{R}_i$. Also, denote by π_0 an atomic proposition associated to the regions of *non-interest*, i.e., π_0 holds true if and only if $x \in \mathcal{X} \backslash (\cup_{i=1}^{N_I} \mathcal{R}_i)$. The atomic proposition π_0 represents a *dummy symbol*, which will not be used to describe the task specification. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_{N_I}\}$ and $h_X : \mathbb{R}^n \to 2^{\Pi}$ be the mapping that maps the state information to the corresponding atomic proposition, i.e.,

$$h_X(x) = \begin{cases} \pi_i, & \text{if} \quad x \in \mathcal{R}_i, \quad i \in \mathbb{N}_{1:N_I}, & (3) \\ \pi_0, & \text{if} \quad x \in \mathcal{X}_{\backslash \mathcal{R}}, & (4) \end{cases}$$

where $\mathcal{X}_{\backslash \mathcal{R}} = \mathcal{X} \backslash (\cup_{i=1}^{N_I} \mathcal{R}_i)$.

B. Satisfaction relation over the state trajectory

Denote by $\mathbf{x} = x_0 x_1 x_2 \cdots$ a state trajectory in accordance to (1), with $x_k \in \mathcal{X}$, $u_k \in \mathcal{U}$, $w_k \in \mathcal{W}$, $\forall k \in \mathbb{N}$. We next define the satisfaction of the formula ϕ by the state trajectory $\mathbf{x}$. To this end, we define the *trajectory of interest* as follows:

Definition 1: Given a state trajectory $\mathbf{x} = x_0 x_1 x_2 \cdots$, the *trajectory of interest* $\mathbf{x}_I$ corresponding to $\mathbf{x}$ is defined as $\mathbf{x}_I = x_{k_0} x_{k_1} x_{k_2} \cdots$ with $k_j < k_{j+1}$, $\forall j \in \mathbb{N}$, such that: $h_X(x_{k_j}) \in \Pi$, $\forall j \in \mathbb{N}$, and $h_X(x_k) = \pi_0$, $\forall k \in (k_j, k_{j+1})$, $\forall j \in \mathbb{N}$. $\Box$

In other words, the trajectory of interest is given by eliminating all states that belong to the regions of non-interest. The trace of the state trajectory is generated based on the trajectory of interest defined as follows:

Definition 2: Given a state trajectory $\mathbf{x} = x_0 x_1 x_2 \cdots$, the *trace* of $\mathbf{x}$ is given by $\mathrm{trace}(\mathbf{x}) = \rho_0 \rho_1 \rho_2 \cdots$, which is generated over the corresponding trajectory of interest $\mathbf{x}_I = x_{k_0}x_{k_1}x_{k_2}\cdots$, satisfying the following rules for all $\ell \in \mathbb{N}$, $i \in \mathbb{N}$:

1) $\rho_0 = h_X(x_{k_0})$;
2) If $\rho_\ell = h_X(x_{k_i})$ and there exists $j > i$ such that $h_X(x_{k_i}) = h_X(x_{k_{i+1}}) = \cdots = h_X(x_{k_j})$, $h_X(x_{k_j}) \neq h_X(x_{k_{j+1}})$, then $\rho_{\ell+1} = h_X(x_{k_{j+1}})$.
3) If $\rho_\ell = h_X(x_{k_i})$, and $h_X(x_{k_j}) = h_X(x_{k_i})$, $\forall j \geq i$, then $\rho_m = \rho_\ell$, $\forall m \geq \ell$. □

For example, assume that $x_0 \in \mathcal{R}_1$, $x_1, x_2 \in \mathcal{X}_{\setminus\mathcal{R}}$, $x_3, x_4 \in \mathcal{R}_2$, $x_5 \in \mathcal{X}_{\setminus\mathcal{R}}$, $x_6, x_7 \in \mathcal{X}_3 \cdots$ (i.e., the trajectory traverses $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3 \cdots$). The trajectory of interest is given by $x_0x_3x_4x_6x_7\cdots$. The trace of the trajectory according to Definition 2 is $\rho = \rho_0\rho_1\rho_2\cdots$ with $\rho_0 = \pi_1$, $\rho_1 = \pi_2$, $\rho_2 = \pi_3 \cdots$.

Definition 3: Given a state trajectory $\mathbf{x} = x_0x_1x_2\cdots$, we say that $\mathbf{x}$ satisfies the formula ϕ if and only if the corresponding trace according to Definition 2 satisfies ϕ, i.e., $\pi_{seq} = \mathrm{trace}(\mathbf{x}) \models \phi$.

C. Communication strategy

During online implementation, the plant needs to receive suitable control signals from the networked controller for achieving the LTL specification ϕ. In this paper, we employ a *self-triggered strategy* [1], which is known to be one of the most useful energy-aware communication protocols. To provide the overview, suppose that at $k \in \mathbb{N}$ the plant transmits the state information x_k to the controller. Then, the controller not only computes suitable control signals to be applied, but also the next communication time $k' > k$. Once k' is determined, a set of control actions from k to $k'-1$, say $u_k, \ldots, u_{k'-1}$, are transmitted in packets to the plant, and these are applied in an open-loop fashion. Thus, no communication occurs until the next communication time k'. Moreover, as the controllers are given in an open-loop fashion, the plant does not need to measure the state information until k'. As a consequence, both sensor and communication systems do not need to be used for all times until the next communication time, and, therefore, energy savings of battery powered devices can be achieved.

D. Problem formulation

The goal of this paper is to design control and self-triggered strategies, such that the resulting state trajectory satisfies a given LTL formula. That is:

Problem 1: For a given LTL specification ϕ, design both control and self-triggered strategies, such that the resulting state trajectory of (1) satisfies ϕ. □

IV. Constructing transition system

As a first step to solve Problem 1, we construct a finite transition system that represents an *abstracted* model to describe the behavior of the control systems in (1). Specifically, we aim at obtaining $\mathcal{T} = (S, s_{init}, \delta, \Pi, g)$, where $S = \{s_1, \ldots, s_{N_I}\}$ is a set of symbolic states, $s_{init} \in S$ is an initial state, $\delta \subseteq S \times S$ is a transition relation, $\Pi = \{\pi_1, \ldots, \pi_{N_I}\}$ is a set of atomic propositions, and $g : S \to 2^{\Pi}$ is a labeling function. Roughly speaking, each symbol $s_i \in S$ indicates the region of interest $\mathcal{R}_i$ (i.e., the region having the same index i). To relate each symbol to the corresponding region of interest, let $\Gamma : S \to \mathcal{R}$ be the mapping given by

$$\Gamma(s_i) = \mathcal{R}_i, \quad \forall i \in \mathbb{N}_{1:N_I}. \tag{5}$$

Conversely, let $\Gamma^{-1} : \mathcal{R} \to S$ be the mapping from each region of interest to the corresponding symbolic state. The symbol $s_{init} \in S$ represents the symbolic state associated with $\mathcal{R}_{init}$, i.e., $s_{init} = \Gamma^{-1}(\mathcal{R}_{init})$. The labeling function $g(s_i)$ outputs the atomic proposition assigned to $\mathcal{R}_i$, i.e., $g(s_i) = \pi_i$. The transition relation $(s_i, s_j) \in \delta$ indicates that every $x \in \mathcal{R}_i$ can be steered to $\mathcal{R}_j$ in finite time. A more formal definition of δ is provided below.

A. Definition of reachability

To characterize δ in the transition system, let us analyze the reachability among the regions of interest. To this end, consider a pair $(\mathcal{R}_i, \mathcal{R}_j) \in \mathcal{R} \times \mathcal{R}$ with $i \neq j$. For notational simplicity, let $\mathcal{X}_{ij} \subset \mathcal{X}$ be given by

$$\mathcal{X}_{ij} = \mathcal{X} \backslash \bigcup_{n \in \mathbb{N}_{\setminus ij}} \mathcal{R}_n, \tag{6}$$

where $\mathbb{N}_{\setminus ij} = \{1, \ldots, N_I\} \backslash \{i, j\}$. That is, $\mathcal{X}_{ij}$ represents the free space that we exclude all regions of interest other than $\mathcal{R}_i$ and $\mathcal{R}_j$. Note that $\mathcal{X}_{ij}$ is a polygonal set that can be a non-convex region. Whether the transition is allowed in $\mathcal{T}$, from $s_i = \Gamma^{-1}(\mathcal{R}_i)$ to $s_j = \Gamma^{-1}(\mathcal{R}_j)$ (i.e., $(s_i, s_j) \in \delta$), is determined according to the following notion of *reachability*:

Definition 4 (Reachability): The state is *reachable* from $\mathcal{R}_i$ to $\mathcal{R}_j$ ($i \neq j$), which we denote by $(s_i, s_j) \in \delta$, if there exists a finite $L \in \mathbb{N}_+$ such that the following holds: for any $x_0 \in \mathcal{R}_i$ and the disturbance sequence $w_0, w_1, \ldots, w_{L-1} \in \mathcal{W}$, there exist $u_0, u_1, \ldots, u_{L-1} \in \mathcal{U}$ such that the resulting state trajectory $x_1, \ldots, x_L$ in accordance with (1) satisfies

(C.1) $x_L \in \mathcal{R}_j$,
(C.2) $x_\ell \in \mathcal{X}_{ij}$, $\forall \ell \in \mathbb{N}_{0:L}$,
(C.3) If $x_{\ell'} \in \mathcal{R}_j$ for some $\ell' \in \mathbb{N}_{1:L}$, then $x_\ell \notin \mathcal{R}_i$, $\forall \ell \in \mathbb{N}_{\ell':L}$. □

Based on Definition 4, reachability holds from $\mathcal{R}_i$ to $\mathcal{R}_j$ if there exists a controller such that any state in $\mathcal{R}_i$ can be steered to $\mathcal{R}_j$ in finite time. Moreover, we require by (C.2) that the state needs to avoid any other region of interest apart from $\mathcal{R}_i$ and $\mathcal{R}_j$. Also, (C.3) implies that once the state enters $\mathcal{R}_j$ it must not enter $\mathcal{R}_i$ afterwards. The conditions (C.2), (C.3) are essentially required to guarantee that the *trace* of the state trajectory satisfies the following property:

Proposition 1: For every $x_0 \in \mathcal{R}_i$, the trace of the state trajectory $x_0, x_1, \ldots, x_L$ satisfying (C.1) – (C.3) is $\pi_i\pi_j$.

Proposition 1 implies that the trace of the state trajectory satisfying (C.1) – (C.3), which is generated according to the rules in Definition 2, is consistent with the trace for the trasition from s_i to s_j (i.e., $g(s_i)g(s_j)$). This consistency is important, since otherwise the accepting trace over $\mathcal{T}$ might not lead to the acceptance of the trace over the actual state trajectory. For example, the state trajectory $x_0x_1x_2$ $(L = 2)$

satisfying $x_0 \in \mathcal{R}_i, x_1 \in \mathcal{R}_m, x_2 \in \mathcal{R}_j$ with $m \neq i, j$, which does *not* satisfy the condition (C.2), yields the corresponding trace according to Definition 2 as $\pi_i\pi_m\pi_j (\neq \pi_i\pi_j)$. In this case, the formula such as $\phi = \Box\neg\pi_m$ may be satisfied by $\pi_i\pi_j$ (the trace of $\mathcal{T}$), while, on the other hand, not be satisfied by $\pi_i\pi_m\pi_j$ (the trace of the trajectory). As another example, consider $x_0x_1x_2x_3$ ($L = 3$) satisfying $x_0 \in \mathcal{R}_i$, $x_1 \in \mathcal{R}_j$, $x_2 \in \mathcal{R}_i$ $x_3 \in \mathcal{R}_j$ (i.e., the trajectory traverses $\mathcal{R}_i\mathcal{R}_j\mathcal{R}_i\mathcal{R}_j$). In this case, the condition (C.3) is violated since the state enters $\mathcal{R}_i$ after reaching $\mathcal{R}_j$. The corresponding trace here is $\pi_i\pi_j\pi_i\pi_j (\neq \pi_i\pi_j)$. Since the accepting path of $\mathcal{T}$ satisfying ϕ is designed by the high level planner (see Section V), and the resulting state trajectory is generated based on the accepting path, the above inconsistencies could lead to the violation of ϕ by the state trajectory.

Note also that our definition of reachability differs from the ones presented in [6], which, in contrast to (C.2), requires that once the state enters $\mathcal{R}_j$ it must remain there afterwards (i.e., if $x_{\ell'} \in \mathcal{R}_j$ for some $\ell' \in \mathbb{N}_{1:L}$ then $x_\ell \in \mathcal{R}_j$, $\forall \ell \in \mathbb{N}_{\ell':L}$). Indeed, such condition is more restrictive than (C.2) and is not required in this paper since we allow the states to enter the regions of non-interest while moving from $\mathcal{R}_i$ to $\mathcal{R}_j$. For example, consider $x_0x_1x_2x_3x_4$ satisfying $x_0 \in \mathcal{R}_i$, $x_1 \in \mathcal{X}_{\setminus\mathcal{R}}$, $x_2 \in \mathcal{R}_j$, $x_3 \in \mathcal{X}_{\setminus\mathcal{R}}$, $x_4 \in \mathcal{R}_j$. The corresponding trajectory of interest according to Definition 1 is $x_0x_2x_4$ and so the resulting trace according to Definition 2 is $\pi_i\pi_j$. That is, while the state traverses the regions of non-interest *after* reaching $\mathcal{R}_j$, the corresponding trace is still given by $\pi_i\pi_j$ (as long as the terminal state belongs to $\mathcal{R}_j$) since we eliminate all states in $\mathcal{X}_{\setminus\mathcal{R}}$ when generating the trace.

B. Proposed algorithm to analyze reachability

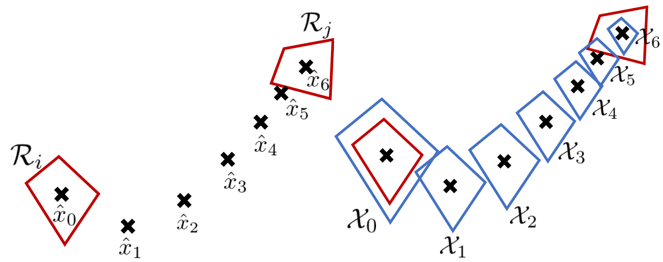
This section provides a way to analyze reachability from $\mathcal{R}_i$ to $\mathcal{R}_j$. The proposed algorithm consists of the two steps; *trajectory generation* and *tube generation*. The following subsections are provided to described these steps in detail.

1) Trajectory generation: In the trajectory generation step, we generate a sample state trajectory from the *nominal* system of (1), starting from a given $\hat{x}_0 \in \mathcal{R}_i$ to the desired target set $\mathcal{R}_j$. Specifically, we produce a state trajectory $\hat{x}_0, \hat{x}_1, \dots \hat{x}_L$, and the corresponding control $\hat{u}_0, \hat{u}_1, \dots \hat{u}_{L-1} \in \mathcal{U}$ for some $L \in \mathbb{N}_+$, such that $\hat{x}_0 = x_{cent,i}$ (recall that $x_{cent,i}$ represents the Chebyshev center of $\mathcal{R}_i$),

$$\hat{x}_{k+1} = A\hat{x}_k + B\hat{u}_k \in \mathcal{X}_{ij}, \quad \forall k \in \mathbb{N}_{1:L-1}, \tag{7}$$

and $\hat{x}_L \in \mathcal{R}_j$. The illustration of the trajectory is shown in Fig. 2(a). Roughly speaking, the trajectory provides a *reference* that the actual states should follow to move from $\mathcal{R}_i$ to $\mathcal{R}_j$. Notice that the nominal trajectory must remain in $\mathcal{X}_{ij}$, which aims to fulfill the condition (C.2) in Definition 4.

So far, numerous techniques have been proposed to generate the above trajectory. Popular ones are the sampling-based algorithms, such as RRT [9], RRT* [11] and their variants [12], which are known to be powerful techniques to find feasible trajectories regardless of the convexity of the polygonal set $\mathcal{X}_{ij}$. Alternative methods are the optimization-based approaches, such as those by solving Mixed Integer Linear Programming (MILP) [13] or other variants to relax the computational complexity [14]. In view of the many different techniques, how the trajectory generation scheme should be applied is beyond the scope of this paper; we can utilize any of the above techniques to obtain the nominal trajectory for the reachability analysis.



(a) Trajectory generation (b) Tube generation

Fig. 2. Illustration of the generated trajectory (black cross marks) and tubes (regions covered by blue lines).

2) Tube generation: In the tube generation step, we solve a convex program to generate a sequence of tubes (polytopes) based on the obtained nominal trajectory. The illustration of the generated tubes is shown in Fig. 2(b). Specifically, we aim to generate a sequence of tubes $\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_L \subset \mathcal{X}$, which are designed as

$$\mathcal{X}_\ell = \hat{x}_\ell \oplus \varepsilon_\ell \mathcal{Z}_{ij} \tag{8}$$

for $\ell \in \mathbb{N}_{0:L}$, where $\varepsilon_\ell > 0$ is the scalar decision variable and $\mathcal{Z}_{ij} = \mathrm{co}\{z_1, \dots, z_p\} \subset \mathbb{R}^n$ denotes a given polytope containing the origin in the interior. The set $\mathcal{Z}_{ij}$ can be arbitrarily chosen if $0 \in \mathcal{Z}_{ij}$ and is not necessary to be an invariant set. A natural choice may be to characterize as $\mathcal{Z}_{ij} = -x_{cent,i} \oplus \mathcal{R}_i$, i.e., select $\mathcal{Z}_{ij}$ as having the same shape as $\mathcal{R}_i$, and the Chebyshev center is the origin. From (8), each $\mathcal{X}_\ell$ is a polytope whose center is a nominal state $\hat{x}_\ell$ and the size (volume) is characterized by the scalar ε_ℓ. Alternatively, each $\mathcal{X}_\ell$ can be expressed by a convex hull $\mathcal{X}_\ell = \mathrm{co}\{x_{\ell,1}, \dots, x_{\ell,p}\}$, where $x_{\ell,s} = \hat{x}_\ell + \varepsilon_\ell z_s$, $s \in \mathbb{N}_{1:p}$. Let $\bar{\varepsilon}_\ell \in \mathbb{R}$, $\ell \in \mathbb{N}_{0:L}$ be given by

$$\bar{\varepsilon}_\ell = \max\{\varepsilon_\ell > 0 \mid \hat{x}_\ell \oplus \varepsilon_\ell \mathcal{Z}_{ij} \subseteq \mathcal{X}_{ij}\}. \tag{9}$$

That is, $\bar{\varepsilon}_\ell$ denotes the maximum value of ε_ℓ such that the corresponding set $\hat{x}_\ell \oplus \varepsilon_\ell \mathcal{Z}_{ij}$ belongs to $\mathcal{X}_{ij}$. Note that this can be easily computed by searching the maximum value of ε_ℓ with the property that the intersection between the polytope $\hat{x}_\ell \oplus \varepsilon_\ell \mathcal{Z}_{ij}$ and the polygonal set $\mathcal{X}_{ij}$ is empty.

Using the above notations, we propose the following optimization problem:

Problem 2: For given $\hat{x}_0, \dots, \hat{x}_L$, $\bar{\varepsilon}_0, \dots, \bar{\varepsilon}_L$ and $\mathcal{Z}_{ij}$, find $\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_L$ and $\mathcal{U}_0$, $\mathcal{U}_1$, …, $\mathcal{U}_L$ with $\mathcal{U}_\ell = \mathrm{co}\{u_{\ell,1}, \dots, u_{\ell,p}\}$, by solving the following problem:

$$\max_{\varepsilon_{0:L},\, \mathcal{U}_{0:L}} \ \varepsilon_0 \tag{10}$$

subject to

$$\begin{cases} \varepsilon_\ell \le \bar{\varepsilon}_\ell, \quad \forall \ell \in \mathbb{N}_{0:L} & (11) \\ \mathcal{X}_L \subseteq \mathcal{R}_j, & (12) \\ Ax_{\ell,s} + Bu_{\ell,s} \in \mathcal{X}_{\ell+1} \ominus \mathcal{W}, & (13) \\ u_{\ell,s} \in \mathcal{U}, & (14) \end{cases}$$

where (13), (14) must hold for all $\ell \in \mathbb{N}_{0:L-1},\ s \in \mathbb{N}_{1:p}$. □

The constraint (11) indicates that the scalar ε_ℓ must be upper bounded by $\bar{\varepsilon}_\ell$, and is utilized to guarantee that the generated tubes remain inside $\mathcal{X}_{ij}$. The constraint (12) indicates that the terminal tube must belong to $\mathcal{R}_j$, which guarantees that the state at the final time step belongs to $\mathcal{R}_j$. The constraint (13) indicates the condition to relate the neighboring tubes; it guarantess that all vertices of the tube $\mathcal{X}_\ell$ can move to the next one $\mathcal{X}_{\ell+1}$ that is constrained by the disturbance set $\mathcal{W}$. Overall, the problem aims at finding a sequence of tubes satisfying all the constraints as described above, by maximizing the volume of $\mathcal{X}_0$ (i.e., the scalar ε_0). Here, maximizing the variable ε_0 is motivated by the following result:

Theorem 1: Suppose that Problem 2 has a solution, providing the optimal tubes and the control sets denoted as $\mathcal{X}_0, \ldots, \mathcal{X}_L$, and $\mathcal{U}_0, \ldots, \mathcal{U}_{L-1}$, respectively. Then, reachability holds from $\mathcal{R}_i$ to $\mathcal{R}_j$, if:

(D.1) $\mathcal{R}_i \subseteq \mathcal{X}_0$;

(D.2) if $\mathcal{X}_{\ell'} \cap \mathcal{R}_j \neq \emptyset$ for $\ell' \in \mathbb{N}_{1:L}$, then $\mathcal{X}_\ell \cap \mathcal{R}_i = \emptyset$, $\forall \ell \in \mathbb{N}_{\ell':L}$.

Theorem 1 indicates that reachability holds from $\mathcal{R}_i$ to $\mathcal{R}_j$, if the computed initial tube $\mathcal{X}_0$ is *large enough* to satisfy $\mathcal{R}_i \subseteq \mathcal{X}_0$, and once some tube intersects $\mathcal{R}_j$, it does not intersect $\mathcal{R}_i$ afterwards. The latter condition is required in order to satisfy (C.3) in Definition 4, which aims to exclude the case when the state moves back to $\mathcal{R}_i$ after the entrance to $\mathcal{R}_j$.

For the notational use in the next section, let L_{x_0}, L_x, L_{u_0}, $L_u : \mathcal{R} \times \mathcal{R} \to 2^{\mathcal{X}}$ be the mappings given by

$$L_{x_0}(\mathcal{R}_i, \mathcal{R}_j) = \mathcal{X}_{0:L}, \quad L_x(\mathcal{R}_i, \mathcal{R}_j) = \mathcal{X}_{1:L} \tag{15}$$

$$L_{u_0}(\mathcal{R}_i, \mathcal{R}_j) = \mathcal{U}_{0:L}, \quad L_u(\mathcal{R}_i, \mathcal{R}_j) = \mathcal{U}_{1:L}, \tag{16}$$

i.e., the above mappings indicate the optimal sequence of tubes and the control sets as the solution to Problem 2 for a given pair of two regions of interest.

3) Summary of the algorithm: From Theorem 1, reachability from $\mathcal{R}_i$ to $\mathcal{R}_j$ can be evaluated by solving Problem 2 and checking if (D.1) and (D.2) hold. Therefore, by executing the trajectory generation step and the tube generation step for every pair $(\mathcal{R}_i, \mathcal{R}_j) \in \mathcal{R} \times \mathcal{R}$ $(i \neq j)$, we can characterize the transition relation δ and then construct $\mathcal{T}$.

V. Implementation

Based on the transition system $\mathcal{T}$ given in the previous section, we now present an online, controller/communication synthesis algorithm. Following the hierarchical-based approach, the proposed algorithm consists of high and low level planning. In particular, we present control and communication strategies in the low level planner, by utilizing the generated tubes obtained in the previous section.

A. High level planning

In the high level planning, the controller produces an infinite sequence of the regions of interest that the state should follow to satisfy the formula ϕ. Since the reachability among the regions of interest can be captured by the transition system $\mathcal{T}$, this can be done by finding a path $s_{seq} \in (2^S)^\omega$ of $\mathcal{T}$, such that $\mathrm{trace}(s_{seq}) \models \phi$ holds. Although there exist several methodologies to achieve this, this paper adopts an automata-based model checking algorithm; in this paper, we only describe the overview of the approach and refer the reader to Chapter 5 in [5] for a more detailed explanation. First, we construct a Büchi automaton associated with the formula ϕ by using off-the-shelf translation tools, such as LTL2BA [15]. Since the automaton can be viewed as a graph consisting nodes and edges, we can combine it with $\mathcal{T}$ and find an accepting path by implementing graph search methodologies. In particular, we construct a product automaton between the Büchi automaton and $\mathcal{T}$, and find the strongly connected components through the depth-first search method. If a certain cost is assigned to each transition of $\mathcal{T}$ (e.g., the time steps to achieve reachability, distance to the obstacles, etc.), we can find the *optimal* path such that the total cost is minimized while satisfying ϕ. For various formulations and approaches, see e.g., [16], [17].

Now, denote by $s^*_{seq} = s^*_0 s^*_1 s^*_2 \cdots$ with $s^*_0 = s_{init}$ the accepting path obtained by the high level planner. We can then obtain the corresponding sequence of regions of interest that is projected from s^*_{seq}, i.e.,

$$\mathcal{R}^*_{seq} = \mathcal{R}^*_0 \mathcal{R}^*_1 \mathcal{R}^*_2 \cdots, \tag{17}$$

where $\mathcal{R}^*_i = \Gamma(s^*_i)$, $\forall i \in \mathbb{N}$. Note that since $s^*_0 = s_{init}$, we have $\mathcal{R}^*_0 = \mathcal{R}_{init}$. As described above, $\mathcal{R}^*_{seq}$ represents the infinite sequence of the regions of interest that the state trajectory should traverse to satisfy ϕ.

B. Low level planning

In the low level planning, the controller designs a sequence of control inputs as well as the communication times based on the regions of interest obtained in (17). First, let $\mathcal{X}^*_0 \mathcal{X}^*_1 \mathcal{X}^*_2 \cdots, \mathcal{U}^*_0 \mathcal{U}^*_1 \mathcal{U}^*_2 \cdots$ denote the sequence of the tubes and the control sets produced by

$$\underbrace{L_{x_0}(\mathcal{R}^*_0, \mathcal{R}^*_1)}_{\mathcal{X}^*_0 \cdots \mathcal{X}^*_{L_1}} \underbrace{L_x(\mathcal{R}^*_1, \mathcal{R}^*_2)}_{\mathcal{X}^*_{L_1+1} \cdots \mathcal{X}^*_{L_2}} \underbrace{L_x(\mathcal{R}^*_2, \mathcal{R}^*_3)}_{\mathcal{X}^*_{L_2+1} \cdots \mathcal{X}^*_{L_3}} \cdots, \tag{18}$$

$$\underbrace{L_{u_0}(\mathcal{R}^*_0, \mathcal{R}^*_1)}_{\mathcal{U}^*_0 \cdots \mathcal{U}^*_{L_1}} \underbrace{L_u(\mathcal{R}^*_1, \mathcal{R}^*_2)}_{\mathcal{U}^*_{L_1+1} \cdots \mathcal{U}^*_{L_2}} \underbrace{L_u(\mathcal{R}^*_2, \mathcal{R}^*_3)}_{\mathcal{U}^*_{L_2+1} \cdots \mathcal{U}^*_{L_3}} \cdots, \tag{19}$$

where for the notational simplicity we denote $L_{x_0}(\mathcal{R}^*_0, \mathcal{R}^*_1) = \mathcal{X}^*_0\ \mathcal{X}^*_2 \cdots \mathcal{X}^*_{L_1}$, $L_{u_0}(\mathcal{R}^*_0, \mathcal{R}^*_1) = \mathcal{U}^*_0\ \mathcal{U}^*_2 \cdots \mathcal{U}^*_{L_1}$, and

$$L_x(\mathcal{R}^*_i, \mathcal{R}^*_{i+1}) = \mathcal{X}^*_{L_i+1} \cdots \mathcal{X}^*_{L_{i+1}}, \quad \forall i \in \mathbb{N}_+, \tag{20}$$

$$L_u(\mathcal{R}^*_i, \mathcal{R}^*_{i+1}) = \mathcal{U}^*_{L_i+1} \cdots \mathcal{U}^*_{L_{i+1}}, \quad \forall i \in \mathbb{N}_+, \tag{21}$$

with indices $L_i \in \mathbb{N}_+$, $i \in \mathbb{N}$ appropriately chosen to line up the sequences: $\mathcal{X}_0^*\mathcal{X}_1^*\mathcal{X}_2^*\cdots$, $\mathcal{U}_0^*\mathcal{U}_1^*\mathcal{U}_2^*\cdots$. Note that we have $\mathcal{R}_0^* = \mathcal{R}_{init} \subseteq \mathcal{X}_0^*$ since $\mathcal{X}_0^*$ represents the initial tube satisfying (D.1). Also, it holds from (12) that $\mathcal{X}_{L_i}^* \subseteq \mathcal{R}_i^*$, $\forall i \in \mathbb{N}$. Thus, $\mathcal{X}_0^*\mathcal{X}_1^*\mathcal{X}_2^*\cdots$ indicates an infinite sequence of tubes that the state should follow to satisfy the formula ϕ. In other words, if the trajectory starting from $x_0 \in \mathcal{R}_{init}(\subseteq \mathcal{X}_0^*)$ is forced to remain in the tubes, i.e., $x_k \in \mathcal{X}_k^*$, $\forall k \in \mathbb{N}$, the trajectory passes through all regions of interest $\mathcal{R}_0^*\mathcal{R}_1^*\mathcal{R}_2^*\cdots$, and thus the satisfaction of ϕ is achieved. Since $\mathcal{X}_k^*$ and $\mathcal{U}_k^*$ for each $k \in \mathbb{N}$ are represented as a convex hull from (8), we can denote them as

$$\mathcal{X}_k^* = \mathrm{co}\{x_{1,k}^*, \ldots, x_{p_k,k}^*\}, \tag{22}$$

$$\mathcal{U}_k^* = \mathrm{co}\{u_{1,k}^*, \ldots, u_{p_k,k}^*\} \tag{23}$$

for all $k \in \mathbb{N}$, where $p_k \in \mathbb{N}_+$ denotes the number of vertices of $\mathcal{X}_k^*$. Note that p_k may be time-varying since the choice of $\mathcal{Z}_{ij}$ in (8) depends on the pair $(\mathcal{R}_i, \mathcal{R}_j)$. The proposed control and communication strategies are given by the following algorithm:

Algorithm 1 (Low level control/communication strategy):

1) Initialization: Set $k = 0$.
2) The plant transmits x_k to the controller.
3) *(Compute control inputs)*: For a given $H \in \mathbb{N}_+$, the controller computes $u_k, u_{k+1}, \ldots, u_{k+H-1} \in \mathcal{U}$ according to the following procedure: set $\hat{x}_k = x_k$ and for all $\ell \in \mathbb{N}_{0:H-1}$,

$$u_{k+\ell} = \sum_{s=1}^{p_{k+\ell}} \lambda_{s,k+\ell} u_{s,k+\ell}^* \tag{24}$$

$$\hat{x}_{k+\ell+1} = A\hat{x}_{k+\ell} + Bu_{k+\ell}, \tag{25}$$

where $\lambda_{s,k+\ell} \in [0,1]$, $s \in \mathbb{N}_{1:p_{k+\ell}}$ are such that $\hat{x}_{k+\ell} = \sum_{s=1}^{p_{k+\ell}} \lambda_{s,k+\ell} x_{s,k+\ell}^*$ with $\sum_{s=1}^{p_{k+\ell}} \lambda_{s,k+\ell} = 1$.
4) *(Compute the next communication time)*: The controller computes the next communication time $k + \ell_k^*$, where $\ell_k^* \in \mathbb{N}_{0:H}$ is determined as

$$\ell_k^* = \max\{\ell' \in \mathbb{N}_{0:H} | \hat{x}_{k+\ell} \oplus \mathcal{W}_{k+\ell} \subseteq \mathcal{X}_{k+\ell}^*, \forall \ell \in \mathbb{N}_{0:\ell'}\}, \tag{26}$$

where $\mathcal{W}_{k+\ell} = \sum_{j=1}^{\ell} A^{j-1}\mathcal{W}$.
5) The controller transmits $u_k, \ldots, u_{k+\ell_k^*-1}$ to the plant. The plant applies the obtained control sequence, and it sends the new state information $x_{k+\ell_k^*}$ to the controller.
6) Set $k \leftarrow k + \ell_k^*$ and go back to step (2). □

As shown in the above algorithm, for each communication time the controller computes the sequence of control actions until the given prediction horizon H, and the next communication time when the plant should again transmit the state information to the controller. The control sequence is firstly obtained by using vertex interporations of the generated tubes according to (24), (25). Applying such control strategy is motivated by following property:

Lemma 1: Suppose that $x_k \in \mathcal{X}_k$ and the control sequence $u_k, u_{k+1}, \ldots, u_{k+H}$ is computed according to (24), (25). Then, we have $\hat{x}_{k+\ell} \in \mathcal{X}_{k+\ell}^* \ominus \mathcal{W}$, $\forall \ell \in \mathbb{N}_{1:H}$.

Lemma 1 implies that by applying the control sequence according to (24), all predictive states remain inside the corresponding tubes that are tightened by $\mathcal{W}$.

Once the controller computes the above control sequence, it proceeds to the computation of the next communication time $k+\ell_k^*$, where ℓ_k^* is given according to (26). As shown in the algorithm, the communication strategy is given in a self-triggered manner since the controller iteratively computes the next communication time for each update time. The following lemma illustrates that actual states are guaranteed to remain inside the tubes until the next communication time:

Lemma 2: Suppose that $x_k \in \mathcal{X}_k^*$ and the control inputs $u_k, u_{k+1}, \ldots, u_{k+\ell_k^*-1}$ are applied at the plant from k to $k + \ell_k^* - 1$. Then, it follows that $x_{k+\ell} \in \mathcal{X}_{k+\ell}^*$ for all $\ell \in \mathbb{N}_{1:\ell_k^*}$. This holds for any disturbance sequence $w_k, w_{k+1}, \ldots, w_{k+\ell_k^*-1} \in \mathcal{W}$. □

Proof: First, observe that the actual state at $k + \ell$, $\ell \in \mathbb{N}_{1:\ell_k^*}$ by applying $u_k, \ldots, u_{k+\ell-1}$ is given by

$$x_{k+\ell} = \hat{x}_{k+\ell} + \sum_{j=1}^{\ell} A^{j-1} w_{k+\ell}. \tag{27}$$

That is, the set $\hat{x}_{k+\ell} \oplus \mathcal{W}_{k+\ell}$, $\mathcal{W}_{k+\ell} = \sum_{j=1}^{\ell} A^{j-1}\mathcal{W}$ indicates the set of *all* actual states that can be reached at $k+\ell$ under the influence of disturbances. From (26) it follows that $\hat{x}_{k+\ell} \oplus \mathcal{W}_{k+\ell} \subseteq \mathcal{X}_{k+\ell}^*$, $\forall \ell \in \mathbb{N}_{1:\ell_k^*}$, which means that, due to (27), $x_{k+\ell} \in \mathcal{X}_{k+\ell}^*$, $\forall w_k, w_{k+1}, \ldots, w_{k+\ell-1} \in \mathcal{W}$. Since this holds for all $\ell \in \mathbb{N}_{1:\ell_k^*}$, the proof is complete. ■

From Lemma 2, we can ensure that $x_{k+\ell} \in \mathcal{X}_{k+\ell}^*$ for all $\ell \in \mathbb{N}_{1:\ell_k^*}$, meaning that applying the control sequence $u_k, \ldots, u_{k+\ell_k^*-1}$ in an open-loop fashion will not violate the formula ϕ. For the remaining time steps $k + \ell$ with $\ell \in \mathbb{N}_{\ell_k^*+1:H}$, however, it is no longer guaranteed that we have $\hat{x}_{k+\ell} \oplus \mathcal{W}_{k+\ell} \subseteq \mathcal{X}_{k+\ell}^*$. This means that the actual states $x_{k+\ell}$ for $\ell \in \mathbb{N}_{\ell_k^*+1:H}$ are possible to leave the set $\mathcal{X}_{k+\ell}^*$, leading to the violation of ϕ. Thus, $k + \ell_k^*$ represents the maximal future time step that the actual states are guaranteed to remain in the tubes to satisfy ϕ, as well as that the next state measurement should be communicated at that point to update the control inputs.

Remark 1 (On the positive inter-transmission times): From Lemma 1 and $\mathcal{W}_1 = \mathcal{W}$, it follows that $x_k \in \mathcal{X}_k \implies \hat{x}_{k+\ell} \oplus \mathcal{W}_1 \subseteq \mathcal{X}_{k+\ell}^*$. Thus, $\hat{x}_{k+\ell} \oplus \mathcal{W}_\ell \subseteq \mathcal{X}_{k+\ell}^*$ for all $\ell = 0, 1$, and, therefore, it holds that $\ell_k^* \geq 1$ (i.e., the inter-transmission time step is always positive). This property is important, since if we *were* to obtain $\ell_k^* = 0$, the controller would not provide a suitable next communication time according to (26). Note that we cannot show $\ell_k^* \geq 2$ since we have $\mathcal{W}_\ell \supseteq \mathcal{W}$ for all $\ell \geq 2$ and thus we cannot ensure that the condition $\hat{x}(k+\ell) \oplus \mathcal{W}_\ell \subseteq \mathcal{X}^*(k+\ell)$ holds.

C. Summary of the implementation

From Lemma 2, it holds that $x_k \in \mathcal{X}_k^* \implies x_{k+\ell} \in \mathcal{X}_{k+\ell}^*, \forall \ell \in \mathbb{N}_{1:\ell_k^*}$. Thus, starting from $x_0 \in \mathcal{R}_{init} \subseteq \mathcal{X}_0^*$, we inductively obtain $x_k \in \mathcal{X}_k^*$, $\forall k \in \mathbb{N}$. Thus, from (18) and $\mathcal{X}_{L_i}^* \subseteq \mathcal{R}_i^*$, $\forall i \in \mathbb{N}$, the actual state traverses all the regions of interest $\mathcal{R}_0^*\mathcal{R}_1^*\mathcal{R}_2^*\cdots$. Moreover, each tube sequence from $\mathcal{R}_i^*$ to $\mathcal{R}_{i+1}^*$ ($i \in \mathbb{N}$) has been obtained by solving Problem 2, and the resulting state trajectory satisfies (C.1)–(C.3) in Definition 4. In other words, the trace of the state trajectory while moving from $\mathcal{R}_i^*$ to $\mathcal{R}_{i+1}^*$ is $g(s_i^*)g(s_{i+1}^*)$ for all $i \in \mathbb{N}$ (see Proposition 1), which leads to the fact that the trace of the overall state trajectory is given by $\mathrm{trace}(\mathbf{x}) = g(s_0^*)g(s_1^*)g(s_2^*)\cdots$. Thus, we obtain $\mathrm{trace}(\mathbf{x}) = \mathrm{trace}(s_{seq}^*) \models \phi$ and so the satisfaction of ϕ is guaranteed. Consequently, we obtain the following result:

Theorem 2: Suppose that for given $x_0 \in \mathcal{R}_{init}$ and LTL formula ϕ, the high level planner finds the sequence of the regions of interest as in (17). Moreover, suppose that the low level planner (Algorithm 1) is implemented. Then, the resulting state trajectory $\mathbf{x} = x_0x_1x_2\cdots$ satisfies ϕ, i.e., $\mathrm{trace}(\mathbf{x}) \models \phi$. This holds for all $w_k \in \mathcal{W}$, $\forall k \in \mathbb{N}$. □

VI. Illustrative Example

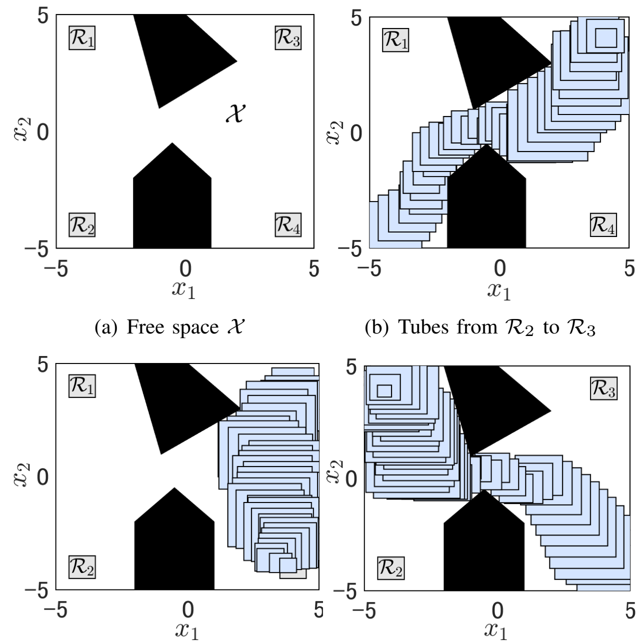
Consider the following system (see [6]):

$$\dot{x} = \begin{bmatrix} 0.2 & -0.3 \\ 0.5 & -0.5 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u + w, \qquad (28)$$

where $x \in \mathbb{R}^2$ is the state, $u \in \mathbb{R}^2$ is the control input, and $w \in \mathbb{R}^2$ is the disturbance. We discretize (28) under a 0.05 sample-and-hold controller to obtain the corresponding discrete-time system in (1). We assume $x_0 = [-4; -4] \in \mathcal{R}_2$ and the control and disturbance size set in (2) are given by $\mathcal{U} = \{u_k \in \mathbb{R} \mid ||u_k||_\infty \leq 6.0\}$ and $\mathcal{W} = \{w_k \in \mathbb{R}^2 \mid ||w_k||_\infty \leq 0.15\}$. The set $\mathcal{X} \subset \mathbb{R}^2$ is shown in Fig. 3(a). In the figure, the white regions represent the free space in which the state can move freely, and the black regions represent obstacles to be avoided. As shown in Fig. 3(a) there exist 4 regions of interest $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$, which are all 1×1 squares.

For example, consider reachability from $\mathcal{R}_2$ to $\mathcal{R}_3$. The trajectory generation step is first implemented by using standard RRT algorithm [9]. The tubes are generated around the nominal trajectory by solving Problem 2 and is illustrated in Fig. 3(b) as the sequence of blue regions. While solving Problem 2, we assume $\mathcal{Z}_{23} = \{z \in \mathbb{R}^2 \mid ||z||_\infty \leq 1\}$. The figure illustrates that the initial tube includes $\mathcal{R}_2$, as well as that once the tube intersects $\mathcal{R}_3$ it does not intersect $\mathcal{R}_2$ afterwards. Therefore, it is shown from Theorem 1 that reachability holds from $\mathcal{R}_2$ to $\mathcal{R}_3$. The generated tubes from $\mathcal{R}_3$ to $\mathcal{R}_4$ and from $\mathcal{R}_4$ to $\mathcal{R}_1$ are also illustrated in Fig. 3(c) and Fig. 3(d), respectively, and we can also show the reachability for these regions. Similarly, we analyze reachability for all the other pairs of the regions of interest and construct the transition system. The resulting transition system $\mathcal{T}$ contains 4 symbolic states and 12 transitions. The total time to construct $\mathcal{T}$ is 1340s on Windows 10, Intel(R) Core(TM) 2.40GHz, 8GB RAM.

To illustrate the implementation result, we consider the following two specifications: $\phi_1 = \square(\lozenge\pi_1 \wedge \lozenge\pi_2 \wedge \lozenge\pi_3 \wedge \lozenge\pi_4)$ (visit all regions infinitely often), $\phi_2 = \square(\lozenge\pi_2 \wedge \lozenge\pi_3)$ (visit $\mathcal{R}_2$ and $\mathcal{R}_3$ infinitely often). Fig. 4 illustrates the resulting state trajectories (blue lines) by implementing Algorithm 1



(a) Free space $\mathcal{X}$ (b) Tubes from $\mathcal{R}_2$ to $\mathcal{R}_3$

(c) Tubes from $\mathcal{R}_3$ to $\mathcal{R}_4$ (d) Tubes from $\mathcal{R}_4$ to $\mathcal{R}_1$

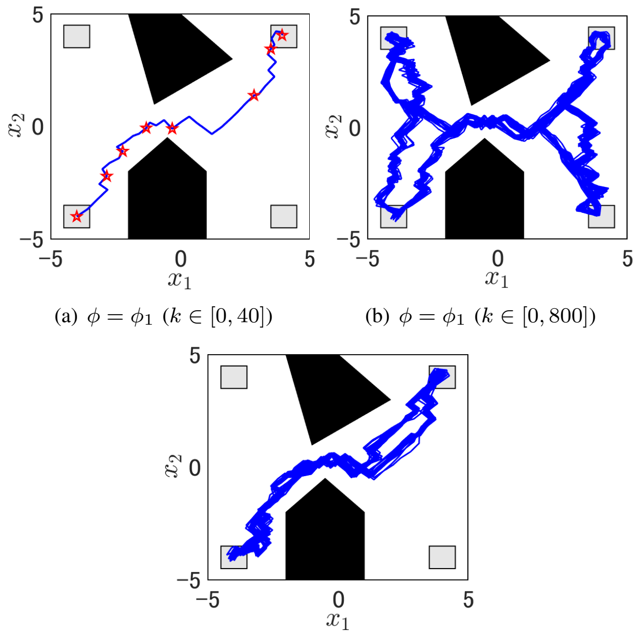
Fig. 3. Illustration of $\mathcal{X}$ and the generated tubes by solving Problem 2.

with $\phi = \phi_1$ for $k \in [0, 40]$ (Fig. 4(a)) and $k \in [0, 800]$ (Fig. 4(b)). During the implementation, we set the prediction horizon as $H = 10$. In Fig. 4(a), red star marks represent the communication instants when control inputs are updated. The number of transmission time instants is 8 during $k \in [0, 40]$, which is 5 times less than the total time steps. It can be seen from the figure that the transmission frequency becomes high when the state is close to the obstables and also to $\mathcal{R}_3$. This is possibly because the generated tubes tend to have small volumes (Fig. 3(b)) when the nominal trajectory is close to the obstacles and $\mathcal{R}_3$, and so the corresponding triggering conditions in (26) become less likely to be satisfied. From Fig. 4(b), it is shown that states are appropriately steered to satisfy the formula ϕ_1 by visiting all regions of interest. Similarly, Fig. 4(c) illustrates another state trajectory by applying Algorithm 1 with $\phi = \phi_2$; from the figure, it is also shown that states are appropriately steered to satisfy ϕ_2.

The number of transmission instants during $k \in [0, 800]$ is given by 121 for $\phi = \phi_1$ and 141 for $\phi = \phi_2$, while the periodic communication ($\ell_k^* = 1, \forall k \in \mathbb{N}$) requires 801. In summary, it achieves less communication load than the periodic scheme, and the effectiveness of the proposed approach is validated.

VII. Conclusion

In this paper, we provide new communication and control synthesis algorithm, which achieves both communication reduction for NCSs and the satisfaction of the LTL formulas. The key idea of the approach is to analyze reachability by producing a sequence of tubes while constructing a transition system. The generated tubes are utilized as criteria not only



(a) $\phi = \phi_1$ ($k \in [0, 40]$)   (b) $\phi = \phi_1$ ($k \in [0, 800]$)

(c) $\phi = \phi_2$ ($k \in [0, 800]$)

Fig. 4. State trajectories by applying Algorithm 1 with $\phi = \phi_1$ and $\phi = \phi_2$.

to check reachability, but also to design a communication strategy such that the resulting state trajectory satisfy the LTL formula. Finally, the benefits of the proposed approach are illustrated through a numerical simulation.

REFERENCES

[1] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proceedings of the 51st IEEE Conference on Decision and Control (IEEE CDC)*, 2012, pp. 3270–3285.

[2] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[3] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, 2013.

[4] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[5] C. Baier and J.-P. Katoen, *Principles of model checking*, The MIT Press, 2008.

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

[7] 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.

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

[9] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press, 2017.

[11] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[12] T. Dang, A. Donze, O. Maler, and N. Shalev, "Sensitive state-space exploration," in *Proceedings of the 47th IEEE International Conference on Decision and Control*, 2008.

[13] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[14] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[15] D. Oddoux and P. Gastin, "LTL2BA software: fast translation from LTL formulae to Büchi automaton," 2001. [Online]. Available: http://www.lsv.ens-cachan.fr/{~}gastin/ltl2ba/

[16] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Revising motion planning under linear temporal logic specifications in partially known workspaces," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[17] X. C. Ding and C. Belta, "Receding horizon surveillance with temporal logic specifications," in *Proceedings of the 49th IEEE International Conference on Decision and Control*, 2010.