Robust Safety Controller Synthesis using Tubes

Kazumune Hashimoto, Shuichi Adachi, and Dimos V. Dimarogonas

***Abstract*—In this paper, we investigate the reachability analysis and safety controller synthesis problem for linear discrete-time systems under additive bounded disturbances. We consider the following problem; design a state feedback controller such that any state trajectories starting from an initial set can be robustly controlled towards a target one in finite time, while at the same time avoiding any prohibited regions. One of the potential disadvantages of existing reachability algorithms when external disturbances are taken into account, may be that the solution to guarantee reachability becomes conservative. Motivated by this, this paper provides a new controller synthesis framework based on the notion of *tube-based control strategy*, in which a suitable sequence of polytopes is generated according to a convex feasibility problem. An illustrative simulation validates the effectiveness of our proposed method.**

I. INTRODUCTION

Reachability and safety controller synthesis have been active areas of research in the community of hybrid systems. The basic concept is to design the control strategy such that the state trajectory reaches a desired target region in finite time, while at the same ensuring certain safety specifications (e.g., avoiding obstacles). This concept has led to a wide variety of applications, such as motion planning of dynamic robots [1], [2], safe platooning or control of manuevers [3], [4], synthetic biology [5], and so on. So far, different theoretical foundations based on different problem formulations have been proposed in the literature, see e.g., [6]–[13]. For example, in [6], [7], reachability analysis is given for continuous-time linear systems on a set of full dimensional polytopes (or simplices) that are partitioned in a state-space. A piece-wise affine control law is designed as a set of vector fields to steer the state to exit a prescribed facet in finite time to enter an adjacent polytope. Similar reachability formulations for the case of discrete-time systems have been also proposed in [8]. Another approach to controller synthesis problem is based on approximately bisimilar abstractions [12], [13]. In this approach, a symbolic model that approximately simulates the behavior of the original (continuous) control system is constructed through the notion of approximate bisimilar relations, and a safety controller is synthesized based on finding appropriate paths by solving symbolic optimal control problems.

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 this paper, we investigate the reachability and controller synthesis problem for linear discrete-time systems, where the basic problem formulation follows the polytope-based approach [6]–[11], as already discussed above. Namely, we consider that the state-space is partitioned by a finite number of polytopes, and for each pair of them the reachability problem is formulated. Following the hierarchical approach in [1], [8], [10], [11], [14], we first construct a finite transition system by analyzing the reachability for all pairs of polytopes. Then, a *high level controller* finds a suitable path of the transition system by implementing standard graph search algorithms to achieve the desired reachablity goals. Based on the generated path, a *low level controller* implements a control strategy such that the corresponding state trajectory can achieve the entrance to the target set while avoiding any prohibited regions.

One of the main contributions of this paper is to provide a new reachability framework that deals with external disturbances. That is, we propose to design a controller such that any state trajectories starting from an initial polytope can be steered towards a target one in finite time, and this is guaranteed under any bounded external disturbances affecting the system model. Note that reachability analysis for linear systems under external disturbances has been also investigated in [10], [11]. In this approach, reachability is analyzed by evaluating open-loop predictions of the states that are propagated over the *worst case* effect of the disturbance sequence. However, this open-loop formulation may lead to a conservative result, since the disturbance effect will be over-estimated as the prediction time step evolves and the feasible sets to satisfy desirable constraints may become significantly tighter.

Instead of analysing reachability via open-loop predictions, this paper provides an alternative reachability framework, which is inspired by a *tube based control strategy* [15], [16]. Originally, tube based strategy has been developed as a Model Predictive Control (MPC) framework for stabilization of (non)linear systems under external disturbances. Unlike standard open-loop MPC formulations, tube based MPC determines a sequence of polytopes (or the so-called *tubes*) and an associated control policy by solving an optimal control problem online. As stated in [15], the tube based approach can moderate the conservativeness compared with the open-loop MPC formulation that needs to have tight constraints to guarantee feasibility. In this paper, we modify the original tube-based framework in order to apply it for reachability analysis, and desirable feedback control policies that steer the state to the target polytope are designed in the hierarchical manner.

The remainder of this paper is organized as follows.

In Section II, the problem formulation is provided. In Section III, reachability analysis and an algorithm to obtain a finite transition system are given. In Section IV, we provide an over-all control algorithm. In Section V, simulation examples validate the effectiveness of the proposed approach. We finally conclude in Section VI.

Notations. Let $\mathbb{R}_+$, $\mathbb{N}$, $\mathbb{N}_+$ be the *positive real, non-negative* and *positive integers*, respectively. For vectors $v_1, \cdots, v_N$, denote by $\mathrm{co}\{v_1, \cdots, v_N\}$ their *convex hull*. A set of vectors $\{v_1, \cdots, v_N\}$ whose convex hull gives a set $\mathcal{P}$ (i.e., $\mathcal{P} = \mathrm{co}\{v_1, \cdots, v_N\}$), and each v_n, $n \in \{1, 2, \cdots, N\}$ is not contained in the convex hull of $v_1, \cdots, v_{n-1}, v_{n+1}, \cdots, v_N$ is called a set of *vertices* of $\mathcal{P}$. 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* $A \ominus B = \{x \in \mathbb{R}^n \mid x + y \in A,\ \forall y \in B\}$.

II. Problem fomulation

In this section, the system description and problem formulation are given.

A. System description

Consider the following discrete-time linear dynamical systems with additive bounded disturbances:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{1}$$

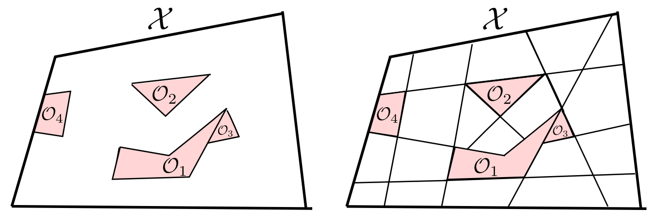
for $k \in \mathbb{N}$, where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ is the control variable, and $w(k) \in \mathbb{R}^n$ is the additive bounded disturbance. We assume that state, control input and disturbance variables satisfy the following constraints:

$$x(k) \in \mathcal{X},\ u(k) \in \mathcal{U},\ w(k) \in \mathcal{W}, \tag{2}$$

for all $k \in \mathbb{N}$, where $\mathcal{X} \subset \mathbb{R}^n, \mathcal{U} \subset \mathbb{R}^m, \mathcal{W} \subset \mathbb{R}^n$ are assumed to be polytopic sets. Regarding the sets $\mathcal{U}$, $\mathcal{W}$, we further assume that the origin is contained in their interiors, i.e., $0 \in \mathcal{U}$, $0 \in \mathcal{W}$. In what follows, we consider that there exist N_o number of bounded polygonal regions $\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_{N_o} \subset \mathcal{X}$, which can be non-convex and represent *prohibited regions* that the state x needs to avoid all the time. In practical situations, such regions may represent obstacles where mobile vehicles or robot manipulators need to avoid while moving in the state-space $\mathcal{X}$. An example of such prohibited regions is illustrated in Fig. 1(a). In the following, we denote by $\mathcal{O} \subset \mathbb{R}^n$ the union of all prohibited regions, i.e., $\mathcal{O} = \bigcup_{k=1}^{N_o} \mathcal{O}_k$. Thus, the areas given by $\mathcal{X} \backslash \mathcal{O}$ represent *safety regions*, in which the state can freely move in the state-space $\mathcal{X}$.

B. Cell decomposition

Before we formulate the problem, several assumptions are given to motivate our control objective of this paper. First, we are not interested in moving the state towards an *exact* point, but in controlling it to a specified *region* in the state-space $\mathcal{X}$. This assumption is reasonable in some practical situations; for example, when a robot aims to move to "*Room A*" as a region of interest, rather than move to a specific point in the area. Moreover, we also assume that such region of interest as well as the other safety regions are all characterized as polytopes. Specifically, we consider that the safety areas $\mathcal{X} \backslash \mathcal{O}$ are decomposed into a finite number of polytopic sets:



(a) Example of some prohibited regions in $\mathcal{X}$ (red regions).

(b) Example of cell decomposition.

Fig. 1. Illustrations of prohibited regions in $\mathcal{X}$ and an example of state-space partitioning.

$$\mathcal{X} \backslash \mathcal{O} = \bigcup_{i=1}^{N} \mathcal{P}_i, \tag{3}$$

which is so-called the *cell decomposition* [17]. In (3), N represents the number of polytopes obtained by the decomposition, and for all pairs $(i, i') \in \{1, \cdots, N\} \times \{1, \cdots, N\}$, $i \neq i'$, it holds that $\mathcal{P}_i \cap \mathcal{P}_{i'}$ is either empty or a common face of $\mathcal{P}_i$ and $\mathcal{P}_{i'}$. The illustration of the cell decomposition is depicted in Fig. 1(b). Here, each $\mathcal{P}_i \subset \mathcal{X}$ may indicate a target region that should be reached in finite time, or it may indicate a region that should be passed through towards a target region.

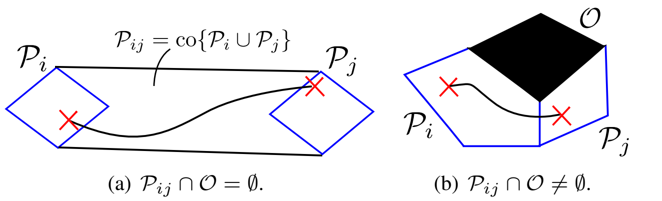
There are numerous techniques in the literature for obtaining the cell decomposition as in (3). In particular, the most well-known decomposition scheme is a triangulation [18], where the bounded state-space is partitioned into a finite number of simplices. The triangulation method is useful, since we can utilize computationally efficient methods to decompose complex polygonal environments. Another decomposition scheme is obtained by using different kinds of polygonal representations, such as rectangles, cylinders [17], or general polytopes [1], [19]. In view of the many different techniques, how the decomposition scheme as illustrated above is applied is beyond the scope of this paper; the main focus of this paper is to provide a reachability and control synthesis framework among the given polytopes obtained by the decomposition. Without loss of generality, we denote each polytope $\mathcal{P}_i$, $i \in \{1, \cdots, N\}$ as

$$\mathcal{P}_i = \mathrm{co}\{v_{i,1}, v_{i,2} \cdots, v_{i,n_i}\}, \tag{4}$$

where $v_{i,1}, \cdots, v_{i,n_i} \in \mathbb{R}^n$ represent the vertices of $\mathcal{P}_i$ and n_i denotes the number of them. Moreover, denote by $\mathbb{P} \in 2^{\mathcal{X}}$ the set of all polytopes obtained by the decomposition, i.e., $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_N\}$.

C. Problem formulation and overview of approach

Suppose that the state is initially somewhere inside a certain polytope, say $\mathcal{P}_{init} \in \mathbb{P}$ (i.e., $x(0) \in \mathcal{P}_{init}$), and it tries to move towards a desired target set $\mathcal{P}_{targ} \in \mathbb{P}$ in finite time. Our goal is then to design a control strategy such that any states starting from $\mathcal{P}_{init}$ can be driven into $\mathcal{P}_{targ}$. That is:



(a) $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$.

(b) $\mathcal{P}_{ij} \cap \mathcal{O} \neq \emptyset$.

Fig. 2. Illustrations for the two cases to check the reachability between $\mathcal{P}_i$, $\mathcal{P}_j$.

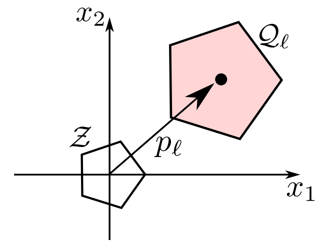


Fig. 3. Illustration of the sets $\mathcal{Z}$ and $\mathcal{Q}_\ell$

Problem 1 (Safety Control Problem)**.** For given initial and target sets $\mathcal{P}_{init}$, $\mathcal{P}_{targ} \in \mathbb{P}$, design a control strategy such that for *any* $x(0) \in \mathcal{P}_{init}$, the closed loop state trajectory enters $\mathcal{P}_{targ}$ in finite time while at the same time avoiding any prohibited regions $\mathcal{O}$. □

To solve Problem 1, this paper adopts a *hierarchical control strategy*, as the relevant approaches presented in [1], [8]–[11], [14]. In the hierarchical approach, we obtain a finite transition system based on the polytopes given by the cell decomposition. The transition system represents an abstracted behavior of the original control system in (1), which consists of a finite number of symbolic states and the corresponding transitions. Obtaining the transition system is useful, since the problem to check the reachability from $\mathcal{P}_{init}$ to $\mathcal{P}_{targ}$ can be solved by finding a suitable path of the transition system through efficient graph search algorithms (e.g., Dijkstra algorithm). Moreover, the transition system has an advantage in adapting control specifications; even though the target (or initial) set has been changed, a suitable path and the corresponding control strategy can be re-designed once the transition is provided in the offline phase.

In this paper, in order to generate the transition system, we propose the so-called *tube based strategy* to analyze the reachability between each pair of polytopes. Specific details of this framework are provided in the next section.

III. Finite abstraction based on reachability

In this section, several steps are provided to generate a finite transition system. In Section III-A, we define the notion of reachability to design controllers among polytopes obtained by the decomposition. We then provide in Section III-B an algorithm to obtain the transition system.

A. Reachability analysis

Consider a pair of two polytopes $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$ (which are not necessarily the initial and the target set), and let $\mathcal{P}_{ij} \subset \mathcal{X}$ be the convex hull of the union of $\mathcal{P}_i$ and $\mathcal{P}_j$, i.e.,

$$\mathcal{P}_{ij} = \text{co}\{\mathcal{P}_i \cup \mathcal{P}_j\}. \tag{5}$$

First, let us analyze if there exists a control strategy to drive the state from $\mathcal{P}_i$ to $\mathcal{P}_j$ while avoiding prohibited regions $\mathcal{O}$. Towards this goal, suppose first that $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$, i.e., the convex hull does not intersect any prohibited regions. In this case, the set $\mathcal{P}_{ij}$ represents a *safety region* since there is no prohibited region inside the set. Thus, it is sufficient to design a controller such that any state starting from $\mathcal{P}_i$ enters $\mathcal{P}_j$, while at the same time always remaining in $\mathcal{P}_{ij}$ to avoid prohibited regions. The illustration is depicted in Fig. 2(a). As we will see later, finding such controller can be formulated by a convex problem, since $\mathcal{P}_{ij}$ as well as $\mathcal{P}_i$, $\mathcal{P}_j$ are all convex sets.

Suppose that, on the other hand, we have $\mathcal{P}_{ij} \cap \mathcal{O} \neq \emptyset$. This case may not be trivial; due to the non-convexity of the safety region $\mathcal{P}_{ij} \backslash \mathcal{O}$, the problem to find a controller may be non-convex and hard to be solved. The problem can still be convex, however, if we try to find a controller such that any state starting from $\mathcal{P}_i$ stays in the same set $\mathcal{P}_i$ (instead of $\mathcal{P}_{ij} \backslash \mathcal{O}$) and it enters $\mathcal{P}_j$ at the final time. The illustration is depicted in Fig. 2(b). Obviously, this constraint is tighter than the former case; as the state needs to be inside $\mathcal{P}_i$ all the time until it enters $\mathcal{P}_j$, it implies that the two polytopes should be adjacent. Yet the problem to find the control strategy becomes convex, as the sets considered here are now all convex. Motivated by the above two cases, the following notion of reachability is given in this paper:

Definition 1 (Reachability)**.** For a given pair of two polytopes $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$, we say that the state is *reachable* in L steps from $\mathcal{P}_i$ to $\mathcal{P}_j$, if for *every* initial state $x(0) \in \mathcal{P}_i$ there exists a set of controllers $u(0), u(1), \cdots, u(L-1) \in \mathcal{U}$, such that the corresponding states $x(1), x(2), \cdots, x(L)$ in accordance with (1) satisfy the following:

- If $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$,
  1) $x(\ell) \in \mathcal{P}_{ij}, \ \ \forall \ell \in \{1, \cdots, L-1\}$;
  2) $x(L) \in \mathcal{P}_j$.
- If $\mathcal{P}_{ij} \cap \mathcal{O} \neq \emptyset$,
  1) $x(\ell) \in \mathcal{P}_i, \ \ \forall \ell \in \{1, \cdots, L-1\}$;
  2) $x(L) \in \mathcal{P}_j$.

The above conditions must hold under any disturbances $w(0), \cdots, w(L-1) \in \mathcal{W}$. □

In the following, we formulate a problem to check reachability in accordance with Definition 1. Suppose that for given $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$ and $L \in \mathbb{N}_+$, we wish to check the reachability from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps. To make sure that the state can be robustly steered to $\mathcal{P}_j$ under the effect of disturbances, the following tube-based strategy is adopted. In the tube based strategy, the problem is to find a suitable sequence of polytopes (or the so-called *tubes*) $\mathcal{Q}_0, \mathcal{Q}_1, \cdots \mathcal{Q}_L \subset \mathcal{X}$ and an associated controller that forces the state trajectories to remain inside the designed polytopes. Aiming to formulate a convex problem, the sequence $\mathcal{Q}_0, \mathcal{Q}_1, \cdots \mathcal{Q}_L$ is more specifically parametrized as follows; for the initial set $\mathcal{Q}_0$, we have $\mathcal{Q}_0 = \mathcal{P}_i$. This means that the initial polytope $\mathcal{Q}_0$ is given fixed and not regarded as a decision variable. For the subsequent sets $\mathcal{Q}_1, \cdots, \mathcal{Q}_L$, we have

$$\mathcal{Q}_\ell = p_\ell \oplus \varepsilon_\ell \mathcal{Z}, \tag{6}$$

for all $\ell \in \{1, 2, \cdots L\}$. The set $\mathcal{Z} \subset \mathbb{R}^n$ is a given polytope characterized by $\mathcal{Z} = \mathrm{co}\{z_1, \cdots, z_{n_i}\}$, where the number of points n_i is the same as the number of vertices of $\mathcal{P}_i$ (see (4)), and $z_1, \cdots, z_{n_i} \in \mathbb{R}^n$ are given points selected such that $0 \in \mathcal{Z}$. $\varepsilon_\ell > 0$ denotes a scalar, decision variable representing the size of $\mathcal{Q}_\ell$, and $p_\ell \in \mathbb{R}^n$ denotes a decision variable representing the center of $\mathcal{Q}_\ell$. The illustration of the two sets $\mathcal{Z}$, $\mathcal{Q}_\ell$ are depicted in Fig. 3. All decision variables to represent the sequence of polytopes are thus given by $(p_{1:L}, \varepsilon_{1:L})$, with $p_{1:L} = \{p_1, \cdots, p_L\}$, $\varepsilon_{1:L} = \{\varepsilon_1, \cdots, \varepsilon_L\}$.

Let $q_n(\ell) = p_\ell + \varepsilon_\ell z_n$, $\forall \ell \in \{1, \cdots, L\}$, $\forall n \in \{1, \cdots, n_i\}$, and $q_n(0) = v_{i,n}$, $\forall n \in \{1, \cdots, n_i\}$ (recall in (4) that $v_{i,1}, \cdots, v_{i,n_i}$ are the vertices of $\mathcal{P}_i$). Then, the sequence of all polytopes $\mathcal{Q}_0, \cdots, \mathcal{Q}_L$ can be also represented by

$$\mathcal{Q}_\ell = \mathrm{co}\{q_1(\ell), q_2(\ell), \cdots, q_{n_i}(\ell)\}, \tag{7}$$

for all $\ell \in \{0, 1, \cdots, L\}$.

Using above notations, we propose the following problem to analyze the reachability from $\mathcal{P}_i$ to $\mathcal{P}_j$:

Problem 2 (Problem to check reachability from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps)**.** Let $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$ and $\mathcal{P}_{ij} = \mathrm{co}\{\mathcal{P}_i \cup \mathcal{P}_j\}$. For given $L \in \mathbb{N}$ and $\mathcal{Z} = \mathrm{co}\{z_1, \cdots, z_{n_i}\}$, find $\mathcal{Q}_{1:L} = \{\mathcal{Q}_1, \cdots, \mathcal{Q}_L\}$ and $\mathcal{U}_{0:L-1} = \{\mathcal{U}_0, \cdots, \mathcal{U}_{L-1}\}$ with $\mathcal{U}_\ell = \mathrm{co}\{u_1(\ell), \cdots, u_{n_i}(\ell)\}$, $\forall \ell \in \{0, \cdots, L-1\}$, by solving the following feasibility problem:

$$\min_{\mathcal{Q}_{1:L},\, \mathcal{U}_{0:L-1}} \quad 0 \tag{8}$$

subject to the following constraints:

- If $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$,

$$\begin{cases} \varepsilon_\ell > 0, \quad \forall \ell \in \{1, \cdots, L\} & (9) \\ \mathcal{Q}_\ell \subseteq \mathcal{P}_{ij}, \ \forall \ell \in \{1, \cdots, L-1\}, & (10) \\ \mathcal{Q}_L \subseteq \mathcal{P}_j, & (11) \\ Aq_n(\ell) + Bu_n(\ell) \in \mathcal{Q}_{\ell+1} \ominus \mathcal{W}, & (12) \\ u_n(\ell) \in \mathcal{U}, & (13) \end{cases}$$

  where the constraints (12) and (13) must hold for all $\ell \in \{0, \cdots, L-1\}$, $n \in \{1, \cdots, n_i\}$.
- If $\mathcal{P}_{ij} \cap \mathcal{O} \neq \emptyset$, the constraints are given by (9), (11), (12), (13) and $\mathcal{Q}_\ell \subseteq \mathcal{P}_i$, $\forall \ell \in \{1, \cdots, L\}$ (i.e., only the constraint (10) is replaced by $\mathcal{Q}_\ell \subseteq \mathcal{P}_i$, $\forall \ell \in \{1, \cdots, L\}$ and the other constraints are the same). □

As shown in Problem 2, the problem is to find feasible sequences of polytopes $\mathcal{Q}_{1:L}$ and control inputs $\mathcal{U}_{0:L-1}$ such that all constraints as illustrated above must be satisfied. Note that although $\mathcal{Q}_0 (= \mathcal{P}_i)$ is not regarded as a decision variable, its vertices $q_1(0), \cdots, q_{n_i}(0)$ appear in (12) (with $\ell = 0$). The

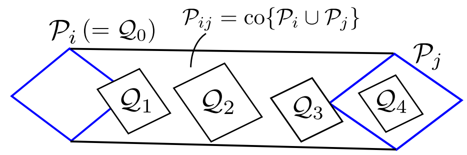


Fig. 4. Illustration of the sets $\mathcal{Q}_{1:L}$ satisfying (10), (11) with $L = 4$ (for the case $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$).

constraints (10), (11) impose that each set $\mathcal{Q}_\ell$ must be inside $\mathcal{P}_{ij}$ and the last one $\mathcal{Q}_L$ must be inside $\mathcal{P}_j$. The illustration of the sequence $\mathcal{Q}_{1:L}$ satisfying such constraints is depicted in Fig. 4. The constraint (12) indicates that there exists a set of controllers $u_1(\ell), \cdots, u_{n_i}(\ell)$ such that all vertices of $\mathcal{Q}_\ell$ can be steered into a tighter set $\mathcal{Q}_{\ell+1} \ominus \mathcal{W}$. As we will see in the analysis that follows to prove reachability, this guarantees that every state in $\mathcal{Q}_\ell$ can be steered towards inside $\mathcal{Q}_{\ell+1}$ under any effect of the disturbance $w \in \mathcal{W}$. All decision variables in Problem 2 are given by $(p_{1:L}, \varepsilon_{1:L}, \mathcal{U}_{0:L-1})$. Problem 2 is formulated as a convex problem, since all inclusion constraints imposed in Problem 2 can be translated into the linear matrix inequalities (LMIs). The following lemma states that the reachability holds if Problem 2 has a feasible solution:

Lemma 1. Suppose that Problem 2 has a solution for given $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$ and a time step $L \in \mathbb{N}_+$. Then, the state is reachable from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps. □

Proof. We prove only for the case $\mathcal{P}_{ij} \cap \mathcal{O} = \emptyset$, since for the other case it can be proven in a similar manner. Let $\mathcal{Q}^*_{1:L}$, $\mathcal{U}^*_{0:L-1}$ be the feasible solution to Problem 2. For convenience, we also let $\mathcal{Q}^*_0 = \mathcal{Q}_0 = \mathcal{P}_i$ and

$$\begin{aligned} \mathcal{Q}^*_\ell &= \mathrm{co}\{q^*_1(\ell), \cdots, q^*_{n_i}(\ell)\}, \quad \forall \ell \in \{0, \cdots, L\} & (14) \\ \mathcal{U}^*_\ell &= \mathrm{co}\{u^*_1(\ell), \cdots, u^*_{n_i}(\ell)\}, \ \forall \ell \in \{0, \cdots, L-1\}. & (15) \end{aligned}$$

From (10) and (11), we obtain $\mathcal{Q}^*_\ell \subseteq \mathcal{P}_{ij}$ for all $\ell \in \{1, \cdots, L\}$ and $\mathcal{Q}^*_L \subseteq \mathcal{P}_j$. Thus, it is sufficient to prove that for every $x(0) \in \mathcal{Q}^*_0 (= \mathcal{P}_i)$, there exist $u(0), \cdots, u(L-1) \in \mathcal{U}$ such that $x(\ell) \in \mathcal{Q}^*_\ell$, $\forall \ell \in \{1, \cdots, L\}$. This can be shown inductively as follows.

Suppose $x(\ell) \in \mathcal{Q}^*_\ell$ for some $\ell \in \{0, \cdots, L-1\}$. Then, there exists $\lambda_n \in [0, 1]$, $n \in \{1, \cdots, n_i\}$ such that $x(\ell) = \sum_{n=1}^{n_i} \lambda_n q^*_n(\ell)$ with $\sum_{n=1}^{n_i} \lambda_n = 1$. Then, let $u(\ell)$ be given by

$$u(\ell) = \sum_{n=1}^{n_i} \lambda_n u^*_n(\ell) \in \mathcal{U}, \tag{16}$$

where the last inclusion follows from the fact that we have $u^*_n(\ell) \in \mathcal{U}$, $\forall \ell \in \{0, \cdots, L-1\}$, $\forall n \in \{1, \cdots, n_i\}$. We obtain

$$\begin{aligned} x(\ell+1) &= Ax(\ell) + Bu(\ell) + w(\ell) \\ &= \sum_{n=1}^{n_i} \lambda_n (Aq^*_n(\ell) + Bu^*_n(\ell)) + w(\ell) \qquad (17) \\ &\in \mathcal{Q}^*_{\ell+1}, \end{aligned}$$

where the last inclusion follows from the fact that we have $Aq_n^*(\ell) + Bu_n^*(\ell) \in \mathcal{Q}_{\ell+1}^* \ominus \mathcal{W}$, $\forall n \in \{1, \cdots, n_i\}, \forall \ell \in \{1, \cdots, L-1\}$ from the constraint given by (12). Thus, $x(\ell) \in \mathcal{Q}_\ell^*$ implies $x(\ell+1) \in \mathcal{Q}_{\ell+1}^*$, $\forall \ell \in \{0, \cdots, L-1\}$ with a suitable choice of control input (given by (16)). Therefore, starting from any $x(0) \in \mathcal{Q}_0^* = \mathcal{P}_i$, there exists $u(0), \cdots, u(L-1) \in \mathcal{U}$ such that $x(\ell) \in \mathcal{Q}_\ell^* \subseteq \mathcal{P}_{ij}$, $\forall \ell \in \{0, \cdots, L-1\}$ and $x(L) \in \mathcal{Q}_L^* \subseteq \mathcal{P}_j$. This concludes that the state is reachable from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps. □

Suppose that Problem 2 has a solution for a given $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$, and a time step L (i.e., the state is reachable from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps), providing feasible sequence of polytopes and the control sets denoted as (14), (15), respectively. Then, starting from any initial state from $\mathcal{P}_i$, the following control strategy can be implemented to steer the state towards $\mathcal{P}_j$:

Algorithm 1 (Control strategy from $\mathcal{P}_i$ to $\mathcal{P}_j$)**.** For a given initial state $x(0) \in \mathcal{P}_i$:

1) (Initialization): Set $\ell = 0$.
2) Given $x(\ell)$, $\ell \in \{0, 1, \cdots, L\}$, compute $\lambda_n \in [0, 1]$, $n \in \{1, \cdots, n_i\}$ such that $x(\ell) = \sum_{n=1}^{n_i} \lambda_n q_n^*(\ell)$ with $\sum_{n=1}^{n_i} \lambda_n = 1$. Then, set $u(\ell) \in \mathcal{U}$ given by (16).
3) Apply $u(\ell)$ to the plant, and set $\ell \leftarrow \ell + 1$. If $\ell = L$, terminate the algorithm. Otherwise, go back to Step 2).

□

As shown in Algorithm 1, the control input $u(\ell) \in \mathcal{U}$ for each $\ell \in \{0, \cdots, L\}$ is computed in the form of (16). Thus, it is inductively shown from (17) that for any $x(0) \in \mathcal{Q}_0 = \mathcal{P}_i$, it holds that $x(\ell) \in \mathcal{Q}_\ell^* \subseteq \mathcal{P}_{ij}$ (or $\mathcal{P}_i$) for all $\ell \in \{1, \cdots, L-1\}$. Moreover, since $\mathcal{Q}_L^* \subseteq \mathcal{P}_j$, we obtain $x(L) \in \mathcal{Q}_L^* \subseteq \mathcal{P}_j$, which means that reachability in L steps has been achieved.

B. Algorithm to generate finite transition system

The transition system that we construct here is based on the reachability presented in the previous subsection. The transition system consists of a set of symbolic states $\mathbb{S} = \{s_1, s_2, \cdots, s_N\}$ and the transition relation $\delta \subseteq \mathbb{S} \times \mathbb{S}$. Here, each $s_i \in \mathbb{S}$ indicates the polytope $\mathcal{P}_i \in \mathbb{P}$ (i.e., the polytope having the same index i), and the transition $(s_i, s_j) \in \delta$ indicates that the reachability holds from $\mathcal{P}_i$ to $\mathcal{P}_j$. Namely, the transition from s_i to s_j is allowed only if the reachability holds from $\mathcal{P}_i$ to $\mathcal{P}_j$ according to Definition 1. To indicate the relation, let $\Gamma : \mathbb{P} \to \mathbb{S}$ be the mapping that sends each polytope to the corresponding symbolic state, i.e., $\Gamma(\mathcal{P}_i) = s_i$. Conversely, let Γ^{-1} be the mapping that sends each symbolic state to the corresponding polytope. The transition system is formally defined as follows:

Definition 2 (Transition system $\mathcal{T}$)**.** A transition system based on the reachability is a tuple $\mathcal{T} = (\mathbb{S}, s_{init}, \delta, s_{targ}, C)$, where;

- $\mathbb{S} = \{s_1, \cdots, s_N\}$ is a set of symbolic states;
- s_{init} is an initial state, where $s_{init} = \Gamma(\mathcal{P}_{init})$;
- $\delta \subseteq \mathbb{S} \times \mathbb{S}$ is a transition relation, where $(s_i, s_j) \in \delta$ only if the reachability holds from $\mathcal{P}_i$ to $\mathcal{P}_j$ according to Definition 1;
- s_{targ} is a terminal state, where $s_{targ} = \Gamma(\mathcal{P}_{targ})$;
- $C : \delta \to \mathbb{R}_+$ is a cost function, where $C(s_i, s_j) = L$ if the reachability holds from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps. □

In Definition 2, a cost function C is defined to represent how many time steps are taken to achieve the reachability. To obtain the transition system $\mathcal{T}$, we need to characterize both the transition relation $\delta \subseteq \mathbb{S} \times \mathbb{S}$ and the cost function C. This can be done by solving Problem 2 iteratively to check the reachability for each pair of two polytopes in $\mathbb{P}$. An overall procedure to characterize both δ and C is presented in Algorithm 2. In the algorithm, we arbitrary pick up a pair of two polytopes in $\mathbb{P}$ and solve Problem 2 to check the reachability between them. While checking the reachability, we increment the time steps L until Problem 2 finds a feasible solution, or it exceeds a given threshold $L_{\max}$. If a feasible solution has been found, we add the corresponding pair of symbolic states to δ and assign C to the reachable time steps L (line 7,8).

Algorithm 2: Transition system generator

Input : $\mathbb{P}$, $\mathbb{S}$, $L_{\max}$ (Set of polytopes, symbolic state domain and a threshold of time steps)
Output: δ, C (Transition relation and cost function)
1 Initialization: set $\delta = \{\emptyset\}$ and $C(s_i, s_j) = \infty$ for all $(s_i, s_j) \in \mathbb{S} \times \mathbb{S}$;
2 **for** *each pair of* $(\mathcal{P}_i, \mathcal{P}_j) \in \mathbb{P} \times \mathbb{P}$ **do**
3   set $L = 1$, $flag = 0$;
4   **while** $flag = 0$ *or* $L < L_{\max}$ **do**
5   solve Problem 2 to check the reachability from $\mathcal{P}_i$ to $\mathcal{P}_j$ in L steps;
6   **if** *Problem 2 has a solution* **then**
7             $\delta \leftarrow \{\delta \cup (s_i, s_j)\}$;
8             $C(s_i, s_j) = L$;
9             $flag = 1$;
10   **end**
11         $L \leftarrow L + 1$;
12   **end**
13 **end**

Remark 1. Note that the computational load of Algorithm 2 may be relatively high, since we need to solve Problem 2 iteratively by incrementing the step size L until the solution has been found. To alleviate such computational burden, one may instead solve Problem 2 for a *fixed* step size L, rather than solving it multiple times with different step sizes. □

IV. Overall Control strategy

Based on the transition system obtained in the previous section, we now present an overall control strategy. The control strategy is given in a hierarchical manner, consisting of a *high level control layer*, and *low level control layer*. In the high level layer, the controller finds a finite path from s_{init} to s_{targ} in the transition system $\mathcal{T}$, such that the summation of the cost function C is minimized. Finding such optimal path can be implemented by using standard graph search algorithms, such as Dijkstra algorithm [17]. The generated path indicates the sequence of polytopes that the states should follow to reach the target set $\mathcal{P}_{targ}$. In the low level control layer, the actual control law is given to the real system based on the sequence of polytopes obtained in the high level layer. During the execution, a local polytope to polytope controller is implemented according to Algorithm 1. The overall control strategy is provided below.

Algorithm 3 (Overall control strategy from $\mathcal{P}_{init}$ to $\mathcal{P}_{targ}$)**.**

1) (High level control): The controller searches a finite path of $\mathcal{T}$:
$$s(0), s(1), \cdots, s(d) \tag{18}$$
for some $d \in \mathbb{N}_+$, where $s(0) = s_{init}$, $s(d) = s_{targ}$ such that the total cost $\sum_{j=0}^{d} C(s(j), s(j+1))$ is minimized by applying, e.g., Dijkstra algorithm. If it does not find such path, stop the algorithm.
2) (Low level control): Let
$$\mathcal{P}(0), \mathcal{P}(1), \cdots, \mathcal{P}(d), \tag{19}$$
where $\mathcal{P}(j) = \Gamma^{-1}(s(j)) \in \mathbb{P}$ for all $j \in \{0, 1, \cdots, d\}$. Apply the following:
   a) (Initialization) Set $j = 0$;
   b) Apply Algorithm 1 as a control strategy from $\mathcal{P}(j)$ to $\mathcal{P}(j+1)$.
   c) Set $j \leftarrow j + 1$. If $j = d$, terminate the algorithm. Otherwise, go back to step b). $\square$

V. Simulation results

As a simulation example, we consider a point-mass robot moving in the 2-D workspace $\mathcal{X} \subset \mathbb{R}^2$, which is illustrated in Fig. 5. In the figure, all black colored regions represent obstacles to be avoided, while all white colored regions represent safety regions in which the robot can move freely. As shown in Fig. 5, the safety regions are decomposed into subsets as 1×1 squares, which are regarded as polytopes to analyze the reachability. We assume that the state of the robot is the position in the workspace $x = [x_1, x_2]$, following the dynamics;
$$\begin{aligned} x_1(k+1) &= x_1(k) + u_1(k)\Delta t + w_1(k), \\ x_2(k+1) &= x_2(k) + u_2(k)\Delta t + w_2(k), \end{aligned} \tag{20}$$
with $\Delta t = 0.5$, where $u = [u_1; u_2] \in \mathbb{R}^2$ denotes the control signal applied to the robot and $w = [w_1; w_2] \in \mathbb{R}^2$ denotes the additive disturbances. For input constraints we assume $\mathcal{U} = \{u \in \mathbb{R} \ : \ |u_1 + u_2| \leq 1\}$ and the disturbance size set is given by $\mathcal{W} = \{w \in \mathbb{R}^2 \ : ||w|| \leq 0.20\}$. Although the dynamics is relatively simple, we utilize this since it is often used as a benchmark example of motion planning (see e.g., [9], [10], [14]). A more complex linear robot motion model can be also considered such as the one presented in [11].

To illustrate the proposed scheme, let us consider an example to check reachability by solving Problem 2 from the polytope labeled as 1 to the one labeled as 2 in Fig. 5. For

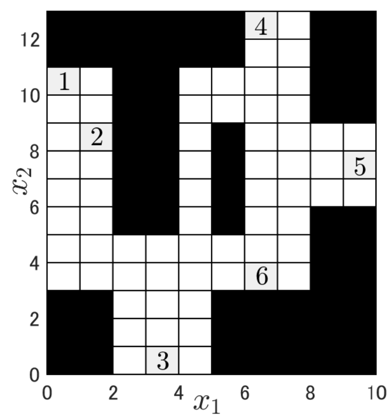
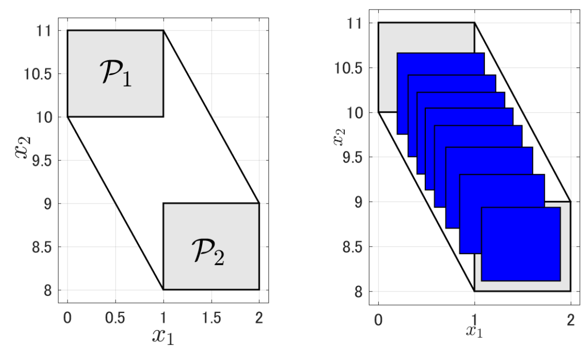


Fig. 5. State space $\mathcal{X} \in \mathbb{R}^2$ considered in the simulation example.



(a) $\mathcal{P}_1$ and $\mathcal{P}_2$ and their convex hull.

(b) Sequence of polytopes $\mathcal{Q}^*_{1:8}$ (blue colored regions) obtained by solving Problem 2.

Fig. 6. Reachability example from $\mathcal{P}_1$ to $\mathcal{P}_2$.

convenience, we denote these polytopes as $\mathcal{P}_1$, $\mathcal{P}_2$, respectively. The enlarged view of these two sets and their convex hull $\mathcal{P}_{12} = \mathrm{co}\{\mathcal{P}_1 \cup \mathcal{P}_2\}$ are depicted in Fig. 6(a). We assume that the set $\mathcal{Z}$ in (6) is given by $\mathcal{Z} = \{[z_1; \ z_2] \in \mathbb{R}^2 : |z_1| \leq 1, |z_2| \leq 1\}$. The solution to Problem 2 has been found with $L = 8$ and the feasible sequence of polytopes $\mathcal{Q}^*_1, \mathcal{Q}^*_2, \cdots, \mathcal{Q}^*_8$ is illustrated in Fig. 6(b). As shown in Fig. 6(b), all generated polytopes are inside $\mathcal{P}_{12}$, and the last polytope $\mathcal{Q}^*_8$ is placed in $\mathcal{P}_2$. In a similar manner, we implement Algorithm 2 with $L_{\max} = 10$ to check the reachability for each pair of polytopes to obtain the transition system $\mathcal{T}$. The resulting $\mathcal{T}$ has 67 symbolic states (equivalent to the number of white cells in Fig. 5), and in total 655 transitions. In this example, it took 950 seconds to construct the transition system on Windows 10, Intel(R) Core(TM) 2.40 GHz, 8 GB RAM.

As for the control specification, we consider a senario where the robot starts from the region 1 (upper left region in Fig. 5), and tries moving towards the regions labeled as 3, 4, 5, and 6 sequentially in that order. Although one might express this as a linear temporal logic specification and could generate the desired path through model checking algorithms (see, e.g., [2]), we simply achieve this task by implementing

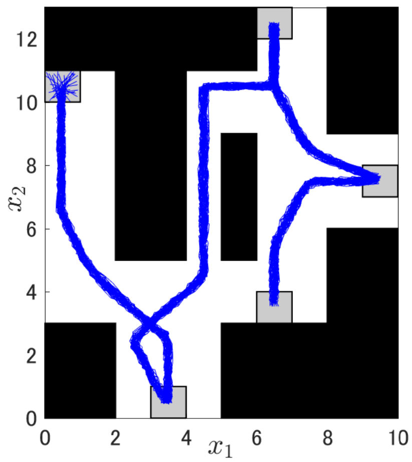


Fig. 7. Trajectories of robot paths (blue lines).

Algorithm 3 iteratively by replacing the initial and target set in accordance with the specified order. That is, we first implement Algorithm 3 by regarding $\mathcal{P}_{init}$ as the region 1 and $\mathcal{P}_{targ}$ as the region 3. Then, once the robot enters the region 3 we replace $\mathcal{P}_{init}$ and $\mathcal{P}_{targ}$ with 3, 4, respectively, and again Algorithm 3 is implemented. This procedure has been taken until the robot finally reaches 6. The resulting state trajectories are shown in Fig. 7 by implementing this procedure. The simulation has been conducted 50 times starting from different initial states $x(0)$ randomly selected in the region 1. From the figure, it is shown that all state trajectories could reach the specified regions in finite time, while avoiding any obstacles regardless of any effect of bounded disturbances.

VI. Conclusion and Future work

In this paper, we propose reachability and control synthesis framework for linear system with additive bounded disturbances. To take into account disturbance effects, we propose a new reachability framework inspired by a tube based strategy, in which a sutable sequence of polytopes is generated by solving a convex feasibility problem. The effectiveness of the proposed control synthesis scheme is verified through a numerical simulation of a robot motion planning.

Our future work is to consider the problem of *scalability*. That is, an algorithm to construct a finite transition system through our current abstraction scheme does not scale well for complex, high-order systems. This problem may be treated by incorporating the idea from the novel approach presented in [1]. Our future work also involves extending our proposed scheme to temporal logic based control [10], [19]. Namely, we assume that the control specification is expressed by termporal logic formula, and provide a control synthesis framework such that high level goals can be achieved.

References

[1] Y. Shoukry, P. Nuzzo, I. Saha, A. L. Sangiovanni-Vincentelli *et al.*, “Scalable Motion Planning Using Lazy SMT-Based Solving,” in *55th IEEE Conference on Decision and Control (IEEE CDC)*, 2016, pp. 6683–6688.

[2] 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.

[3] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, “Safe platooning of unmanned aerial vehicles via reachability,” in *54th IEEE Conference on Decision and Control (IEEE CDC)*, 2015, pp. 4695–4701.

[4] M. Vukosavljev, I. Jansen, M. E. Broucke, and A. P. Schoellig, “Safe and robust robot maneuvers based on reach control,” in *IEEE Conference on Robotics and Automation (ICRA)*, 2016.

[5] G. Batt, B. Yordanov, R. Weiss, and C. Belta, “Robustness analysis and tuning of synthetic gene networks,” *Bioinformatics*, vol. 23, no. 18, pp. 2415–2422, 2007.

[6] L. C. G. J. M. Habets and J. H. van Schuppen, “A control problem for affine dynamical systems on a full-dimensional polytope,” *Automatica*, vol. 40, no. 1, pp. 21–35, 2004.

[7] M. E. Broucke, “Reach control on simplices by continuous state feedback,” *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3482–3500, 2010.

[8] T. E. Hodrus, M. Buchholz, and V. Krebs, “A new local control strategy for control of discrete-time piecewise affine systems,” in *IEEE Conference on Decision and Control and the European Control Conference (IEEE CDC-ECC)*, 2005, pp. 4181–4186.

[9] C. Belta, V. Isler, and G. J. Pappas, “Discrete abstractions for robot motion planning and control in polygonal environments,” *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 864–874, 2004.

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

[11] ——, “Receding horizon temporal logic planning for dynamical systems,” in *IEEE Conference on Decision and Control (IEEE CDC)*, 2009.

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

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

[14] P. Nilsson, N. Özay, U. Topcu, and R. M. Murray, “Temporal logic control of switched affine systems with an application in fuel balancing,” in *IEEE Conference on Decision and Control (IEEE CDC)*, 2012.

[15] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, “Robust model predictive control using tubes,” *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[16] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, “Tube-based robust nonlinear model predictive control,” *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[17] S. M. LaValle, *Planning algorithms*, Cambridge, UK: Cambridge University Press, 2006.

[18] D. T. Lee and B. J. Schachter, “Two algorithms for constructing a delaunay triangulation,” *International Journal of Computer and Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[19] 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.