2013-11-01

Time-constrained temporal logic control of multi-affine systems

Version	
Citation (published version):	Ebru Aydin Gol, Calin Belta. 2013. "Time-constrained temporal logic control of multi-affine systems." Nonlinear Analysis-Hybrid Systems, Volume 10, pp. 21 - 33 (13). doi:10.1016/j.nahs.2013.03.002

https://hdl.handle.net/2144/29960
Boston University

Time-Constrained Temporal Logic Control of Multi-Affine Systems

Ebru Aydin Gol [*] **Calin Belta** [*]

[*] *Boston University, Boston, MA 02215, USA*
e-mail: {ebru,cbelta}@bu.edu

Abstract: In this paper, we consider the problem of controlling a dynamical system such that its trajectories satisfy a temporal logic property in a given amount of time. We focus on multi-affine systems and specifications given as syntactically co-safe linear temporal logic formulas over rectangular regions in the state space. The proposed algorithm is based on the estimation of time bounds for facet reachability problems and solving a time optimal reachability problem on the product between a weighted transition system and an automaton that enforces the satisfaction of the specification. A random optimization algorithm is used to iteratively improve the solution.

1. INTRODUCTION

Temporal logics and model checking algorithms have been primarily used for specifying and verifying correctness of software and hardware systems. Due to their expressivity and resemblance to natural language, temporal logics have gained popularity as specification languages in other areas including dynamical systems. Recently, there has been increasing interest in formal synthesis of dynamical systems, where the goal is to generate a control strategy for a dynamical system from a specification given as a temporal logic formula, such as Linear Temporal Logic (LTL) (Kloetzer and Belta (2008a); Tabuada and Pappas (2003); Girard (2010a)), or fragments of LTL, such as GR(1) (Gazit et al. (2007); Wongpiromsarn et al. (2009)) and syntactically co-safe LTL (Bhatia et al. (2010)).

We focus on a particular class of nonlinear affine control systems, where the drift is a multi-affine vector field (*i.e.,* affine in each state component), the control distribution is constant, and the control is constrained to a convex set. This class of dynamics includes the Euler, Volterra (Volterra (1926)) and Lotka-Volterra (Lotka (1925)) equations, attitude and velocity control systems for aircraft (Nijmeijer and van der Schaft (1990)) and underwater vehicles (Belta (2004)), and models of biochemical networks (de Jong (2002)). In Belta and Habets (2006), the authors studied the problem of synthesizing a state feedback controller such that the trajectories originating in a rectangle leave it through a specified facet. These results were generalized in Habets et al. (2006) by allowing the trajectories to leave through a set of exit facets.

In this paper, we consider the following problem: given a multi-affine control system and a syntactically co-safe LTL formula over rectangular subregions of the state space, find a set of initial states for which there exists a control strategy such that all the trajectories of the closed-loop system satisfy the formula within a given time bound. Syntactically co-safe LTL formulas can be used to describe finite horizon specifications such as target reachability with obstacle avoidance: “always avoid obstacle O until reaching target T”, sequencing constraints “do not go to A or B unless C was visited before”, and more complex temporal and Boolean logic combinations of these. Our approach to this problem consists of two main steps. First, we construct a finite abstraction of the system by solving facet reachability problems on a rectangular partition of the state space. We build on the results from Belta and Habets (2006); Habets et al. (2006) to derive bounds for the exit times of the trajectories. Second, we solve time optimal reachability problems on the product between the abstraction and an automaton that enforces the satisfaction of the specification. We propose an iterative refinement procedure via a random optimization algorithm.

Finite abstractions for controlling dynamical systems have been widely used, e.g by Tabuada and Pappas (2003). Time optimal control of dynamical systems through abstractions has been studied by Mazo and Tabuada (2011) and Girard (2010b). In both cases, an optimal controller is synthesized for an approximate abstraction, which is then mapped to a suboptimal solution for the original system for specifications given in the form of “reach and avoid” sets. While our solution also involves an optimal control problem on the abstraction, our automata-theoretic approach allows for richer, temporal logic control specifications.

The remainder of the paper is organized as follows. We review some notions necessary throughout the paper in Sec. 2 before formulating the problem and outlining the approach in Sec. 3. A review of facet reachability problems and the derivation of the exit time bounds are presented in Sec. 4. The control strategy providing a solution to the main problem is described in Sec.5 and the random optimization method for refinement is given in Sec. 6. An example is given in Sec. 7 and conclusions are summarized in Sec. 8.

[⋆] This work was partially supported at Boston University by grants AFOSR YIP FA9550-09-1-0209, ARO W911NF-09-1-0088, NSF CNS-0834260, ONR MURI N00014-10-10952, and ONR MURI N00014-09-1051.

2. PRELIMINARIES

2.1 Transition systems and linear temporal logic

Definition 1. A weighted transition system is a tuple $\mathcal{T} = (Q, \Sigma, \delta, O, o, w)$, where Q and Σ are sets of states and inputs, $\delta : Q \times \Sigma \longrightarrow 2^Q$ is a transition map, O is a set of observations, $o : Q \longrightarrow O$ is an observation map, and $w : Q \times \Sigma \longrightarrow \mathbb{R}_+$ is a map that assigns a positive weight to each state and input pair.

$\delta(q, \sigma)$ denotes the set of successor states of q under the input σ. If the cardinality of $\delta(q, \sigma)$ is one, the transition $\delta(q, \sigma)$ is deterministic. A transition system $\mathcal{T}$ is called deterministic if all its transitions are deterministic.

A finite input word $\sigma_1 \ldots \sigma_n$, $\sigma_i \in \Sigma$, $i = 1, \ldots, n$ and an initial state $q_0 \in Q$ define a trajectory $r = q_0 \ldots q_n$ of the system with the property that $q_{i+1} \in \delta(q_i, \sigma_{i+1})$ for all $0 \le i \le n-1$. The cost $J^{\mathcal{T}}(r)$ of trajectory r is defined as the sum of the corresponding weights, i.e.,

$$J^{\mathcal{T}}(r) = \sum_{i=0}^{n-1} w(q_i, \sigma_{i+1}).$$

A trajectory $r = q_0 \ldots q_n$ produces a word $o(q_0) \ldots o(q_n)$.

Definition 2. (Kupferman and Vardi (2001)) A syntactically co-safe LTL (scLTL) formula over a set of atomic propositions Π is inductively defined as follows:

$$\Phi := \pi | \neg\pi | \Phi \vee \Phi | \Phi \wedge \Phi | \Phi \mathcal{U} \Phi | \mathcal{F}\Phi, \tag{1}$$

where $\pi \in \Pi$ is an atomic proposition, $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction) are Boolean operators, and $\mathcal{U}$ ("until"), and $\mathcal{F}$ ("eventually") are temporal operators [1].

The semantics of scLTL formulas is defined over infinite words over 2^Π. Informally, $\pi_1 \mathcal{U} \pi_2$ states that π_1 is true until π_2 is true and π_2 becomes eventually true in a word; $\mathcal{F}\pi_1$ states that π_1 becomes true at some position in the word. More complex specifications can be defined by combing temporal and Boolean operators (see Eqn. (27)).

An important property of scLTL formulas is that, even though they have infinite-time semantics, their satisfaction is guaranteed in finite time. Explicitly, for any scLTL formula Φ over Π, any satisfying infinite word over 2^Π contains a satisfying finite prefix.

Definition 3. A deterministic finite state automaton (FSA) is a tuple $\mathcal{A} = (S, \Pi, \delta_{\mathcal{A}}, S_0, F)$ where S is a finite set of states, Π is an input alphabet, $S_0 \subseteq S$ is a set of initial states, $F \subseteq S$ is a set of final states, and $\delta_{\mathcal{A}} : S \times \Pi \longrightarrow S$ is a deterministic transition relation.

An accepting run $r_{\mathcal{A}}$ of an automaton $\mathcal{A}$ on a finite word $w = w_0 \ldots w_d$ over Σ is a sequence of states $r_{\mathcal{A}} = s_0 \ldots s_{d+1}$ such that $s_0 \in S_0$, $s_{d+1} \in F$ and $\delta_{\mathcal{A}}(s_i, w_i) = s_{i+1}$ for all $i = 0, \ldots, d$. For any scLTL Φ formula over Π, there exists a FSA $\mathcal{A}$ with input alphabet 2^Π that accepts the prefixes of all the satisfying words. There are algorithmic procedures and off-the-shelf tools, such as *scheck2* by Latvala (2003), for the construction of such an automaton.

Definition 4. Given a weighted transition system $\mathcal{T} = (Q, \Sigma, \delta, O, o, w)$ and a FSA $\mathcal{A} = (S, \Pi, \delta_{\mathcal{A}}, S_0, F)$ with $O = \Pi$, their product automaton is a FSA $\mathcal{A}^P = (S_P, \Sigma, \delta_P, S_{P0}, F_P)$ where $S_P = Q \times S$ is the set of states, Σ is the input alphabet, $\delta_P : S_P \times \Sigma \longrightarrow 2^{S_P}$ is the transition relation with $\delta_P((q, s), \sigma) = \{(q', s') \mid q' \in \delta(q, \sigma), \delta_{\mathcal{A}}(s, o(q)) = s'\}$, $S_{P0} = Q \times S_0$ is the set of initial states, and $F_P = Q \times F$ is the set of final states.

An accepting run $r_P = (q_0, s_0) \ldots (q_n, s_n)$ of $\mathcal{A}^P$ defines an accepting run $s_0 \ldots s_n$ of $\mathcal{A}$ over input word $o(q_0) \ldots o(q_{n-1})$. The weight function of the transition system can directly be used to assign weights to transitions of $\mathcal{A}^P$, i.e., we can define a weight function for the product automaton in the form $w_P(\delta_P((q, s), \sigma)) = w(\delta(q, \sigma))$. The corresponding cost for a run $r_P = (q_0, s_0) \ldots (q_n, s_n)$ of $\mathcal{A}^P$ over $\sigma_1 \ldots \sigma_n$ is defined as

$$J^P(r_P) = \sum_{i=1}^{n} w_P(\delta_P((q_{i-1}, s_{i-1}), \sigma_i)).$$

2.2 Rectangles and multi-affine functions

For $N \in \mathbb{N}$, an N-dimensional rectangle $R_N(a, b) \subset \mathbb{R}^N$ is characterized by two vectors $a = (a_1, \ldots, a_N)$ and $b = (b_1, \ldots, b_N)$ with the property that $a_i < b_i$ for all $i = 1, \ldots, N$:

$$R_N(a, b) = \{x \in \mathbb{R}^N \mid \forall i \in \{1, \ldots, N\} : \ a_i \le x_i \le b_i\}. \tag{2}$$

Let $\mathcal{V}(a, b)$ and $\mathcal{F}(a, b)$ be the set of vertices and facets of of $R_N(a, b)$, respectively. Let $F^{\pm e_i}$ denote the facet with normal $\pm e_i$, where e_i, $i = 1, \ldots, N$ denote the standard basis of $\mathbb{R}^N$. For a facet $F \in \mathcal{F}(a, b)$, $\mathcal{V}(F)$ denotes its set of vertices and n_F denotes its outer normal. For a vertex $v \in \mathcal{V}(a, b)$, $\mathcal{F}_v$ denotes the set of facets containing v.

Definition 5. A *multi-affine function* $h : \mathbb{R}^N \longrightarrow \mathbb{R}^q$ (with $N, q \in \mathbb{N}$) is a function that is affine in each of its variables, i.e., h is of the form

$$h(x_1, \ldots, x_N) = \sum_{i_1, \ldots, i_N \in \{0,1\}} c_{i_1, \ldots, i_N} x_1^{i_1} \cdots x_N^{i_N},$$

with $c_{i_1, \ldots, i_N} \in \mathbb{R}^q$ for all $i_1, \ldots, i_N \in \{0, 1\}$, and using the convention that if $i_k = 0$, then $x_k^{i_k} \equiv 1$.

Belta and Habets (2006) showed that a multi-affine function h on a rectangle $R_N(a, b)$ is uniquely defined by its values at the vertices, and inside the rectangle the function is a convex combination of its values at the vertices:

$$h(x_1, \ldots, x_N) = \sum_{v \in \mathcal{V}(a,b)} \prod_{i=1}^{N} \left(\frac{x_i - a_i}{b_i - a_i}\right)^{\xi_i(v_i)} \left(\frac{b_i - x_i}{b_i - a_i}\right)^{1 - \xi_i(v_i)} \cdot h(v). \tag{3}$$

where $\xi_i : \{a_i, b_i\} \longrightarrow \{0, 1\}$ is an indicator function such that $\xi_i(a_i) = 0$ and $\xi_i(b_i) = 1$ for all $i = 1, \ldots, N$.

3. PROBLEM FORMULATION

Consider a continuous-time multi-affine control system of the form

[1] The scLTL syntax usually includes a "next" temporal operator. We do not use it here because it is irrelevant for the particular semantics of continuous trajectories that we define later.

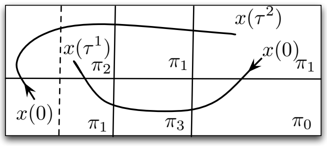


Fig. 1. Examples of continuous trajectories of system (4). The atomic propositions are shown in the rectangles where they are satisfied.

$$\dot{x}(t) = h(x(t)) + Bu(t), \qquad x(t) \in R_N(a^{\mathbb{X}}, b^{\mathbb{X}}), u(t) \in U \tag{4}$$

where $R_N(a^{\mathbb{X}}, b^{\mathbb{X}}) \subset \mathbb{R}^N$, $B \in \mathbb{R}^{N \times m}$, and the control input $u(t)$ is restricted to a polyhedral set $U \subset \mathbb{R}^m$.

Rectangular regions of interests in $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$ are defined using a set of atomic propositions $\Pi = \{\pi_i \mid i = 0, \ldots, l\}$. Each atomic proposition π_i is satisfied in a set of rectangular subsets of the state space of system (4), which is denoted as:

$$[\pi_i] = \cup_{j=1}^{d_i} R_N(a^{j,\pi_i}, b^{j,\pi_i}) \subset R_N(a^{\mathbb{X}}, b^{\mathbb{X}}), \quad d_i \in \mathbb{N}. \tag{5}$$

The specifications are given as scLTL formulas over the set of predicates Π. A trajectory of system (4) satisfies the specification if the word produced by the trajectory satisfies the corresponding formula. Informally, while a trajectory of system (4) evolves, it produces the satisfying predicates and the sequence of predicates defines the word produced by a trajectory. Specifically, a trajectory produces predicate π_i whenever it spends a finite amount of time in a rectangle where π_i is satisfied. For example, trajectories $\{x(t)\}_{0 \le t \le \tau^1}$ and $\{x(t)\}_{0 \le t \le \tau^2}$ shown in Fig. 1 produce the words $\pi_1\pi_0\pi_3\pi_1\pi_2$ and $\pi_1\pi_2\pi_1\pi_1$, respectively. The word produced by a trajectory depends on how the rectangles are defined. The presented approach employs a refinement procedure based on adding hyperplanes, which induces smaller rectangles that inherit the predicate. For example, if the dashed line in Fig. 1 is added, the trajectory $\{x(t)\}_{0 \le t \le \tau^2}$ produces $\pi_1\pi_2\pi_2\pi_1\pi_1$. As discussed by Kloetzer and Belta (2008a), when LTL without next operator is considered, $\pi_1\pi_2\pi_1\pi_1$ and $\pi_1\pi_2\pi_2\pi_1\pi_1$ satisfy the same set of LTL formulas.

Remark 1. In this paper, we study finite time trajectories of system (4). When infinite time trajectories are of interest, invariant controllers can be considered as in Habets et al. (2006).

Problem 1. Given a syntactically co-safe LTL formula Φ over a set of predicates Π and a time bound T, find a set of initial states $\mathbb{X}_0 \subset R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$ and a feedback control strategy such that all words produced by the closed-loop trajectories of system (4) originating in $\mathbb{X}_0$ satisfy the formula in time less than T.

Our proposed solution to Prob.1 starts with a proposition-preserving rectangular partition[2] of $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$, i.e., each element of the partition is a rectangle $R_N(a, b) \subseteq R_N(a^{j,\pi_i}, b^{j,\pi_i})$ for some $j = 1, \ldots, d_i$, $i = 0, \ldots, l$ from Eqn. (5). For each rectangle in the partition, and for each subset of its set of facets, we derive state-feedback controllers driving all the initial states in the rectangle through the set of facets in finite time by using the sufficient conditions derived in Habets et al. (2006). We compute upper bounds for these times and choose the feedback controllers that minimize the upper bounds for each rectangle and each set of exit facets. We then construct a weighted transition system, in which the states label the rectangles from the partition, the inputs label the controllers, and the weights capture the time bounds. We find an optimal run of this transition system that satisfies the formula by solving an optimal reachability problem on its product with an FSA that accepts the language satisfying the formula. The rectangles corresponding to the initial states with costs less than T compose the set $\mathbb{X}_0$. In order to increase this set, we use an iterative refinement of the partition based on a random optimization algorithm.

4. FACET REACHABILITY PROBLEMS

In this section, we focus on the derivation of the facet reachability controllers and their corresponding time bounds. We first summarize the sufficient conditions for facet reachability from Habets et al. (2006):

Theorem 1. Let $R_N(a, b)$ be a rectangle and $\mathcal{E} \subset \mathcal{F}(a, b)$ be a non-empty subset of its facets. There exists a multi-affine feedback controller $k : R_N(a, b) \longrightarrow U$ such that all the trajectories of the closed-loop system (4) originating in $R_N(a, b)$ leave it through a facet from the set $\mathcal{E}$ in finite time if the following conditions are satisfied:

$$n_F^\top(h(v) + Bk(v)) \le 0, \forall F \in \mathcal{F}_v \setminus \mathcal{E} \quad \forall v \in \mathcal{V}(a, b), \tag{6}$$

$$0 \notin Conv(\{h(v) + Bk(v) \mid v \in \mathcal{V}(a, b)\}) \tag{7}$$

where $Conv$ denotes the convex hull.

In particular, when the cardinality of $\mathcal{E}$ is 1, i.e. $\mathcal{E} = \{F\}$, then Eqns. (6) and (7) imply that the speed towards the exit facet F has to be positive everywhere in $R_N(a, b)$, i.e.

$$0 < n_F^\top(h(v) + Bk(v)), \forall v \in \mathcal{V}(a, b). \tag{8}$$

As a consequence, for this particular case, the sufficient conditions (6) and (7) can be replaced with (6) and (8).

The linear inequalities given in (6) and (8) (or (6) and (7)) define a set of admissible controls U_v for each vertex $v \in \mathcal{V}(a, b)$. By choosing a control for each vertex v from the corresponding set U_v, we can construct a multi-affine state feedback controller k that solves the corresponding control problem by using Eqn. (3). We first provide a time upper bound for the case when there is only one exit facet (Prop. 1), and then use this result to provide an upper bound for the general case (Cor. 1).

Proposition 1. Assume that $k : R_N(a, b) \longrightarrow U$ is an admissible multi-affine feedback controller that solves the control-to-facet problem for a facet $F \in \mathcal{F}(a, b)$ with outer normal e_i of a rectangle $R_N(a, b)$. Then all the trajectories of the closed loop system starting in rectangle $R_N(a, b)$ leave the rectangle through facet F in time less than T^F, where

$$T^F = \ln\left(\frac{s_F}{\overline{s_F}}\right) \frac{b_i - a_i}{s_F - \overline{s_F}}, \tag{9}$$

with

$$s_F = \min_{v \in \mathcal{V}(F)} ((h(v) + Bk(v))_i),$$

$$\overline{s_F} = \min_{v \in \mathcal{V}(\overline{F})} ((h(v) + Bk(v))_i),$$

[2] We use the term "partition" loosely in this paper. The rectangle boundaries are irrelevant, since due to the synthesized controllers the trajectories never slide along the boundaries.

where $\overline{F}$ denotes the facet opposite to F, i.e. with normal $-e_i$.

Proof: Let $x \in R_N(a,b)$ and $x^p, \overline{x^p}$ be the projections of x on F and $\overline{F}$, respectively. Then, we have $x = \frac{b_i - x_i}{b_i - a_i}\overline{x^p} + (1 - \frac{b_i - x_i}{b_i - a_i})x^p$. For every $x \in R_N(a,b)$, $h(x)$ is a convex combination of $\{h(v) \mid v \in \mathcal{V}(a,b)\}$. Furthermore, if x belongs to a facet of $R_N(a,b)$, then $h(x)$ is a convex combination of the values of h at the vertices of that facet. Therefore, we have

$$s_F \leq (h(x^p) + Bk(x^p))_i, \tag{10}$$

$$\overline{s_F} \leq (h(\overline{x^p}) + Bk(\overline{x^p}))_i. \tag{11}$$

Since $k(x)$ is a solution of the control-to-facet problem for facet F, the speed towards F is positive everywhere in $R_N(a,b)$, hence

$$0 < s(x) := \frac{b_i - x_i}{b_i - a_i}\overline{s_F} + (1 - \frac{b_i - x_i}{b_i - a_i})s_F \leq (h(x) + Bk(x))_i. \tag{12}$$

For any $x \in R_N(a,b)$, the speed in the i^{th} direction is lower bounded by $s(x)$ (Eqn. (12)), which depends linearly on x_i. Since system (13) defined below is always slower than the original one, its time upper bound to reach facet F gives a valid upper bound for the original system.

$$\dot{x_i}(t) = \frac{b_i - x_i}{b_i - a_i}\overline{s_F} + (1 - \frac{b_i - x_i}{b_i - a_i})s_F, x_i \in [a_i, b_i] \tag{13}$$

The explicit solution of Eqn. (13) is given in Eqn. (14), where x_i^0 denotes the i^{th} component of the initial condition.

$$x_i(t) = \exp(\frac{s_{Fi} - \overline{s_F}}{b_i - a_i}t)(x_i^0 + \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}}) - \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}} \tag{14}$$

Solving (14) for time $T_{x_i^0}^F$ at $x(T_{x_i^0}^F) = b_i$ gives the time upper bound from Eqn. (15). Any trajectory starting from an initial point x in $R_N(a,b)$ with $x_i = x_i^0$ reaches the facet F^{e_i} in time less than $T_{x_i^0}^F$.

$$T_{x_i^0}^F = \ln(\frac{b_i + \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}}}{x_i^0 + \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}}})\frac{b_i - a_i}{s_F - \overline{s_F}} \tag{15}$$

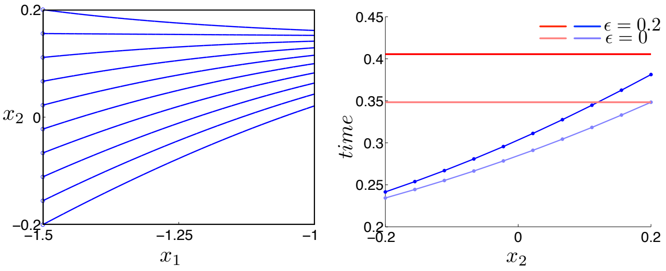
As $T_{x_i^0}^F$ attains its maximum when $x_i^0 = a_i$,

$$T^F = \ln(\frac{b_i + \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}}}{a_i + \frac{b_i\overline{s_F} - a_i s_F}{s_F - \overline{s_F}}})\frac{b_i - a_i}{s_F - \overline{s_F}} = \ln(\frac{s_F}{\overline{s_F}})\frac{b_i - a_i}{s_F - \overline{s_F}} \tag{16}$$

gives the upper bound for all $x \in R_N(a,b)$. □

Prop. 1 uses the fact that if $k : R_N(a,b) \longrightarrow U$ is a solution to the considered control-to-facet problem, then the speed $n_F^\top(h(x) + Bk(x))$ towards the exit facet is positive for all $x \in R_N(a,b)$. By defining a slower system using minimum speeds on F and $\overline{F}$ towards the exit facet, a time bound for the original system is found. A more conservative time bound T'^F can be computed using only the minimum speed towards F, i.e. $T'^F = \frac{b_i - a_i}{\min(s_F, \overline{s_F})}$. While it is more efficient to compute T'^F, T^F gives a tighter bound ($T^F \leq T'^F$). Indeed, the computation of T^F considers the change on the lower bound of speed with respect to x_i. Moreover, while s_F gets closer to $\overline{s_F}$, T'^F approaches T^F:

$$\lim_{s_F \to \overline{s_F}} \ln(\frac{s_F}{\overline{s_F}})\frac{b_i - a_i}{s_F - \overline{s_F}} = \frac{b_i - a_i}{\overline{s_F}} \tag{17}$$



(a) (b)

Fig. 2. (a) Rectangle $[-1.5, -1] \times [-0.2, 0.2]$ and sample trajectories originating in facet F^{-e_1}. (b) Simulation times according to the initial condition in x_2 and time bounds $T^{F^{e_1}}$ (red lines) computed using Eqn. (9) for controllers synthesized from (19) for $\epsilon = 0$ and $\epsilon = 0.2$.

Remark 2. The time bound T^F from Eqn. (9) is attainable in some cases. Let $v_{s_F} = \arg\min_{v \in \mathcal{V}(F)} ((h(v) + Bk(v))_i)$ and $\overline{v_{s_F}} = \arg\min_{v \in \mathcal{V}(\overline{F})} ((h(v) + Bk(v))_i)$. If

$$\begin{array}{l} (v_{s_F})_j = (\overline{v_{s_F}})_j, \\ (h(v_{s_F}) + Bk(v_{s_F}))_j = 0, \\ (h(\overline{v_{s_F}}) + Bk(\overline{v_{s_F}}))_j = 0, \quad j = 1, \ldots, N, j \neq i, \end{array} \tag{18}$$

then the trajectory originating at $\overline{v_{s_F}} \in \overline{F}$ reaches $v_{s_F} \in F$ at time T^F.

For each vertex $v \in \mathcal{V}(a,b)$, we can minimize the time bound given in Prop. 1 if we choose a control $u_v \in U_v$ that maximizes $n_F^\top(h(v) + Bu_v)$. Computationally, this involves solving a linear program at each vertex of a rectangle. Formally, at each vertex v, the optimization problem can be written as:

$$\begin{array}{l} \max\limits_{u_v} \quad n_F^\top(h(v) + Bu_v) \\ n_{F'}^\top(h(v) + Bu_v) \leq -\epsilon, \forall F' \in \mathcal{F}_v \setminus F \\ u_v \in U \end{array} \tag{19}$$

where $0 < \epsilon$, which is a robustness parameter guaranteeing that a trajectory never reaches a facet other than F while moving towards F. Decreasing ϵ relaxes the problem (19) by increasing the size of the feasible region, which results in higher speeds and tighter time bounds. Note that when $0 < \epsilon$ the equalities given in Eqn. (18) can not hold, since for a vertex v the speed towards a facet $F' \in \mathcal{F}_v \backslash F$ is upper bounded by $-\epsilon$. Therefore the robustness parameter ϵ also affects the distance between the time bound from Eqn. (9) and the actual maximal amount of time required to reach F.

The tightness of the time bound from Eqn. (9) and the effects of the robustness parameter ϵ are illustrated through an example in Fig. 2, where the control problem for exit facet F^{e_1} of rectangle $[-1.5, -1] \times [-0.2, 0.2]$ is considered for the control system from Eqn. (26). Some trajectories of the closed loop system obtained by using the feedback controller that minimizes $T^{F^{e_1}}$ when $\epsilon = 0.2$ are shown in Fig. 2a. The corresponding times for reaching F^{e_1} for $\epsilon = 0$ and $\epsilon = 0.2$ are shown in Fig. 2b. Note that when $\epsilon = 0$, the trajectory starting from $(-1.5, 0.2)$ reaches facet F^{e_1} exactly at time $T^{F^{e_1}}$.

Corollary 1. Given a rectangle $R_N(a,b)$ and an admissible multi-affine feedback $k : R_N(a,b) \longrightarrow U$ that solves the control problem from Thm. 1 with set of exit facets $\mathcal{E}$, all trajectories of the closed loop system originating in rectangle $R_N(a,b)$ leave it through a facet $F \in \mathcal{E}$ in time less than

$$T^{\mathcal{E}} = \min_{F \in \mathcal{E}} T^F,$$

where each T^F is computed as in Prop. 1 if $0 < n_F^\top(h(v) + Bk(v))$ for all $v \in \mathcal{V}(a,b)$. Otherwise T^F is set to ∞.

Proof: Let $F \in \mathcal{E}$ with $0 < n^F(h(v) + Bk(v))$ for all $v \in \mathcal{V}(a,b)$. Then by Prop. 1 every trajectory originating in $R_N(a,b)$ reaches F within time T^F (9) unless it leaves $R_N(a,b)$ before reaching F. Hence, $\min_{F\in\mathcal{E}} T^F$ gives a valid bound to the control-to-set-of-facets problem for $\mathcal{E}$. $\square$

For a facet reachability problem with $\mathcal{E}$ as the set of exit facets, T^F is computed for each $F \in \mathcal{E}$ through choosing controls that minimize T^F (9) and satisfy the linear inequalities defined in Thm. 1. Computationally, this translates to solving the following linear program for each $v \in \mathcal{V}(a,b)$ and for each $F \in \mathcal{E}$:

$$\begin{aligned} &\max_{u_v} \quad n_F^\top(h(v) + Bu_v) \\ &n_{F'}^\top(h(v) + Bu_v) \le -\epsilon, \forall F' \in \mathcal{F}_v \setminus \mathcal{E} \\ &u_v \in U \end{aligned} \tag{20}$$

where ϵ is defined as in optimization problem (19).

As already stated, T^F for $F \in \mathcal{E}$ is calculated as in (9) if the speeds at all vertices are positive towards F. In this case, the condition from (7) is trivially satisfied. Then a multi-affine feedback k is constructed by using the controls where $\min_{F\in\mathcal{E}} T^F$ attains its minimum.

5. CONTROL STRATEGY

In this section, we provide a solution to Prob. 1 for a proposition-preserving partition of $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$. We use the results from Sec. 4 to construct a weighted transition system from the partition and find an optimal control strategy for the weighted transition system. The control strategy enforces the satisfaction of the specification and maps directly to a strategy for system (4).

A proposition-preserving partition of $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$ and solutions of facet reachability problems for the rectangles in the partition set define a weighted transition system $\mathcal{T} = (Q, \Sigma, \delta, O, o, w)$. Each state $q \in Q$ of $\mathcal{T}$ corresponds to a rectangle $R_N(a^q, b^q)$ in the partition set. An input $\sigma \in \Sigma$ of $\mathcal{T}$ indicates a non-empty subset of the facets of a rectangle and a transition $\delta(q, \sigma)$ is introduced if the corresponding control problem has a solution. Specifically, we consider a facet reachability problem for each state $q \in Q$ and each non-empty subset of $\mathcal{F}(a^q, b^q)$, and find the multi-affine feedback control which minimizes the corresponding time bound as explained in Sec. 4. The successors of $\delta(q,\sigma)$ are the states q' such that $R_N(a^q, b^q)$ and $R_N(a^{q'}, b^{q'})$ have a common facet in σ. The transition weights are assigned according to the time bounds computed as described in Prop. 1 and Cor. 1. O equals to the set of predicates Π and $o(q) = \pi_i$ if $R_N(a^q, b^q) \subseteq [\pi_i]$.

All words that satisfy the specification formula Φ are accepted by a FSA $\mathcal{A} = (S, \Pi, \delta_{\mathcal{A}}, S_0, F)$ [3]. We construct a product automaton $\mathcal{A}^P = (S_P, \Sigma, \delta_P, S_{P0}, F_P)$ from $\mathcal{T}$ and $\mathcal{A}$ as described in Def. 4.

A control strategy (S_Ω, Ω) for $\mathcal{A}^P$ is defined as a set of initial states S_Ω and a state feedback control function $\Omega : S_P \longrightarrow \Sigma$ implying that $\Omega(s)$ will be the input at state s. The state feedback function Ω characterizes the set of initial states $S_\Omega \subset S_{P0}$ such that every run $s_0 s_1 \ldots s_n$ of $\mathcal{A}^P$ starting from a state s_0 in S_Ω is an accepting run over the word $\Omega(s_0) \ldots \Omega(s_{n-1})$. Since $\mathcal{A}^P$ is non-deterministic, there can be multiple runs starting from a state $s_0 \in S_\Omega$ under the feedback control Ω. In literature (Kloetzer and Belta (2008b), Wolfgang (2002)), non-determinism is resolved through a reachability game played between a protagonist and an adversary, and S_Ω is defined as the set of initial states such that the protagonist always wins the game by applying Ω. Next, we introduce an algorithm based on fixed-point computation to find a maximal S_Ω and corresponding feedback control Ω through optimizing a cost for each $s \in S_P$. Asarin and Maler (2009) used a similar algorithm to solve optimal reachability problems on timed game automata.

Remark 3. Generally, the reachability games are considered over an infinite horizon such as Buchi games, where winning a game for the protagonist means identifying and reaching an invariant set of "good" states. As we consider FSAs, the acceptance condition coincides with finite time reachability. Hence, a simple reachability algorithm is sufficient in our case.

Let $J_\Omega : S_P \to \mathbb{R}_+$ be a cost function with respect to a set of final states F_P and feedback control Ω such that any run of $\mathcal{A}^P$ starting from s reaches a state $f \in F_P$ under the feedback control Ω with a cost upper bounded by $J_\Omega(s)$. Note that if there exists a run starting from s that can not reach F_P, the cost is infinity, $J_\Omega(s) = \infty$.

The solution of the fixed-point problem given in Eqn. (21) gives the optimal cost for each $s \in S$.

$$J(s) = \min(J(s), \min_{\sigma\in\Sigma} \max_{s'\in\delta_P(s,\sigma)} J(s') + w_P(\delta_P(s,\sigma))) \tag{21}$$

Algorithm 1 Compute J and Ω for $\mathcal{A}^P = (S_P, \Sigma, \delta_P, S_{P0}, F_P)$

```
$J(s) = \infty, \forall s \in S_P$
$J(f) = 0, \forall f \in F_P$
$SC = \{s | \exists \sigma \in \Sigma$ and $f \in F_P$ such that $f \in \delta_P(s,\sigma)\}$
while $SC \neq \emptyset$ do
    $SC = SC \setminus \{s\}$, for some $s \in SC$
    if $\min_{\sigma\in\Sigma} \max_{s'\in\delta_P(s,\sigma)} J(s') + w_P(\delta_P(s,\sigma)) < J(s)$ then
        $\Omega(s) = \arg\min_{\sigma\in\Sigma} \max_{s'\in\delta_P(s,\sigma)} J(s') + w_P(\delta_P(s,\sigma))$
        $J(s) = \max_{s'\in\delta_P(s,\Omega(s))} J(s') + w_P(\delta_P(s,\Omega(s)))$
        $SC = SC \cup \{s' | \exists \sigma \in \Sigma, s \in \delta_P(s',\sigma)\}$
    end if
end while
```

Alg. 1 implements the solution for the fixed-point problem in Eqn. (21) for the states of $\mathcal{A}^P$ and finds the optimal

[3] In the general case, as described in Sec. 2, the input alphabet of this automaton is 2^Π. However, since the words generated by system (4) are over Π, it is sufficient to consider Π as the input alphabet for the automaton.

feedback control Ω. A finite state cost, $J(s) < \infty$, and a feedback control Ω resulted from Alg. 1 means that every run starting from s reaches a state f in F_P under the feedback control Ω with a cost at most $J(s)$. Therefore, $S_\Omega = \{s \mid J(s) < \infty, s \in S_0\}$ is the maximal set of initial states of $\mathcal{A}^P$ such that under the feedback control Ω all runs starting from S_Ω are accepting. Consequently,

$$S_\Omega^T = \{s \mid J(s) < T, s \in S_0\} \tag{22}$$

is the maximal set of initial states such that under the feedback control Ω cost of a run starting from S_Ω^T is upper bounded by T.

If only control-to-facet problems are considered while constructing the transition system $\mathcal{T}$, $\mathcal{T}$ and the product automaton $\mathcal{A}^P$ become deterministic. Hence, in this case it is sufficient to use a shortest path algorithm to find optimum costs and feedback control Ω instead of Alg. 1.

If a multi-affine feedback k solves facet reachability problem for the set of exit facets $\mathcal{E} \subset \mathcal{F}(a,b)$ of rectangle $R_N(a,b)$, then k is a solution of the facet reachability problem for every superset $\mathcal{E}'$ of $\mathcal{E}$ with the same time bound $T^{\mathcal{E}}$ by Cor. 1. While constructing δ of $\mathcal{T}$, a solution is searched for every subset of $\mathcal{F}(a,b)$, hence

$$w_P(\delta_P(s,\mathcal{E}')) \leq w_P(\delta_P(s,\mathcal{E})), if \mathcal{E} \subseteq \mathcal{E}'. \tag{23}$$

In line 6 of Alg. 1, cost of a state is updated according to the state with maximum cost among a transitions successor states, hence Alg. 1 tends to choose the $\mathcal{E}$ with minimum cardinality among the sets $\mathcal{E}' \subset \mathcal{F}(a,b)$ with the same transition cost.

Control Strategy for $\mathcal{T}$: (Kloetzer and Belta (2008b)) We construct a control strategy $(Q_0, \mathcal{A}^C)$ for $\mathcal{T}$ using the control strategy (S_Ω^T, Ω) for $\mathcal{A}^P$ resulted from Alg. 1 and Eqn.(22). The set of initial states Q_0 is the projection of S_Ω^T to the states of $\mathcal{T}$. Since the feedback control Ω for $\mathcal{A}^P$ becomes non-stationary when projected to the states of $\mathcal{T}$, we construct a feedback control for $\mathcal{T}$ in the form of a feedback control automaton $\mathcal{A}^C = (S_C, Q, \delta_C, S_{C0}, F_C, \Omega_C, \Sigma)$. The feedback control automaton $\mathcal{A}^C$ reads the current state of $\mathcal{T}$ and outputs the input to be applied to that state. The set of states S_C, the set of initial states S_{C0} and the set of final states F_C of $\mathcal{A}^C$ are inherited from $\mathcal{A}$, the set of inputs Q is the states of $\mathcal{T}$. The memory update function $\delta_C : S_C \times Q \longrightarrow S_C$ is defined as $\delta_C(s,q) = \delta_{\mathcal{A}}(q, o(q))$ if $\delta_{\mathcal{A}}(q,o(q))$ is defined. The output alphabet Σ is the input alphabet of $\mathcal{T}$. $\Omega_C : S_C \times Q \longrightarrow \Sigma$ is the output function, $\Omega_C(s,q) = \Omega((q,s))$ if $J((q,s)) < T$ and $\Omega_C(s,q)$ is undefined otherwise.

If we set the set of observations of $\mathcal{T}$ to Q and define the observation map o as an identity map, then the product of $\mathcal{T}$ and $\mathcal{A}^C$ will have same states and transitions as $\mathcal{A}^P$. Hence, the words produced by trajectories of $\mathcal{T}$ starting from Q_0 in closed loop with $\mathcal{A}^C$ satisfy Φ.

Control strategy $(Q_0, \mathcal{A}^C)$ for $\mathcal{T}$ is used as a control strategy for system (4) by mapping the output of $\mathcal{A}^C$ to the corresponding multi-affine feedback controller. This strategy guarantees that every trajectory of system (4) originating in $\mathbb{X}_0$ given in Eqn. (24) satisfies Φ in time less than T.

$$\mathbb{X}_0 = \bigcup_{q \in Q_0} R_N(a^q, b^q) \tag{24}$$

For every $x_0 \in \mathbb{X}_0$, there exists an initial state $q \in Q_0$ and $s \in S_{C0}$ such that $x_0 \in R_N(a^q, b^q)$ and $(q,s) \in S_\Omega^T$ from Eqn. (22). Let $k_{s,\Omega_C(s,q)}$ be the multi-affine feedback which solves control-to-facet (or control-to-set-of-facets) problem on $R_N(a^q, b^q)$ for $\Omega_C(s,q)$ as the set of exit facets. Starting from x_0 multi-affine feedback $k_{s,\Omega_C(s,q)}$ is applied to system (4) until the trajectory reaches a facet $F \in \Omega_C(s,q)$ with a positive speed towards F. By construction of $\mathcal{A}^C$, it is guaranteed that the trajectory reaches a facet $F \in \Omega_C(s,q)$ in time less than $w(\delta(q, \Omega_C(s,q)))$. Then the applied multi-affine feedback switches to $k_{s',\Omega_C(s',q')}$ where $F = R_N(a^q, b^q) \cap R_N(a^{q'}, b^{q'})$ and $s' = \delta_C(s,q)$. This process continues until a final state $f \in F_C$ of $\mathcal{A}^C$ is reached.

Theorem 2. The trajectories of system (4) originating in $\mathbb{X}_0$ (24) with control strategy $(Q_0, \mathcal{A}^C)$ satisfies Φ in time less than T.

Proof: By Def. 4, every word produced by an accepting run of $\mathcal{A}^P$ satisfies Φ. Hence, by construction of $(Q_0, \mathcal{A}^C)$ and $\mathbb{X}_0$ the words produced by closed loop trajectories of system (4) originating in $\mathbb{X}_0$ satisfy Φ. Consider a finite trajectory $\{x(t)\}_{0 \leq t \leq \tau}$ of system (4) with $x(0) \in \mathbb{X}_0$ evolving under the control strategy $(Q_0, \mathcal{A}^C)$. Let $r_C = s_0 s_1 \ldots s_n$ be the corresponding run of $\mathcal{A}^C$, $r_{\mathcal{T}} = q_0 q_1 \ldots q_n$ be the corresponding trajectory of $\mathcal{T}$ and t_i be a time instant when control switch occurs, i.e. at time t_i, the trajectory hits a facet $F \in \Omega_C(s_{i-1}, q_{i-1})$ with a positive speed towards F while evolving under the multi-affine feedback $k_{s_{i-1},\Omega_C(s_{i-1},q_{i-1})}$, for all $i = 1, \ldots, n$ and $t_n = \tau$. By Prop. 1 and Cor. 1, for all $i = 1, \ldots, n$:

$$t_i - t_{i-1} \leq w(\delta(q_{i-1}, \Omega_C(s_{i-1}, q_{i-1}))) = T^{\Omega_C(s_{i-1},q_{i-1})}. \tag{25}$$

By Alg. 1, $\tau \leq J((q_0, s_0))$ and by Eqn. (24) $\tau \leq T$. □

In Thm. 2, we showed that the proposed feedback control strategy solves Prob. 1 for a proposition-preserving partition of $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$. Next we describe an iterative refinement procedure to increase the volume of $\mathbb{X}_0$.

6. REFINEMENT

An iterative refinement procedure is employed to enlarge the set $\mathbb{X}_0$ (24). As mentioned before, the rectangles defined by the set of predicates induce an initial proposition-preserving grid partition of $R_N(a^{\mathbb{X}}, b^{\mathbb{X}})$. A grid partition is defined by a set of thresholds $\{d_i^j\}_{i \in \mathbb{N}}$ for each dimension $1 \leq j \leq N$.

Introducing a new threshold d_*^j in dimension j can affect $\mathbb{X}_0$ in different ways and it does not always enlarge the set $\mathbb{X}_0$. Consider a state $s \in S_{P0}$ with $J(s)$ as computed in Alg. 1 and corresponding rectangle $R_N(a,b)$ with $a_j < d_*^j < b_j$. Assume a multi-affine feedback $k : R_N(a,b) \longrightarrow U$ solves the control-to-facet problem for a facet $F \in \mathcal{F}(a,b)$ with outer normal e_i and assume the corresponding time bound is T^F as given in Prop. 1. When $R_N(a,b)$ is partitioned into two rectangles $R_N(a,b^*)$ and $R_N(a^*,b)$ through a hyperplane $x_j = d_*^j$, we need to consider two cases: $j = i$ and $j \neq i$, which are illustrated in Fig. 3 on a rectangle in $\mathbb{R}^2$.

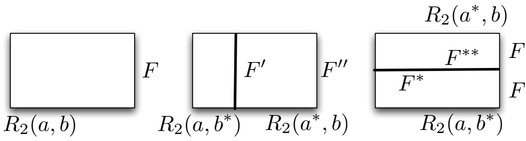


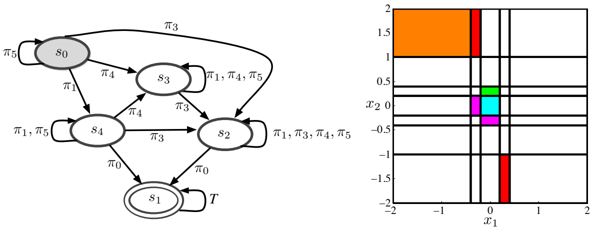
Fig. 3. Two partitioning schemes for $R_2(a.b) \subset \mathbb{R}^2$.

(a)$\{i = j\}$ Since state feedback k solves the control-to-facet problem on $R_N(a,b)$ for F, the speed towards the exit facet is positive for all $x \in R_N(a,b)$. Moreover, no trajectory leaves $R_N(a,b)$ through another facet. Hence, k solves the control-to-facet problems on $R_N(a,b^*)$ and $R_N(a^*,b)$ for the facets with normal e_i. Let $T^{F'}$ and $T^{F''}$ be the corresponding time bounds. Then when k is applied, any trajectory starting in $R_N(a,b^*)$ and $R_N(a^*,b)$ reaches F within time $T^{F'} + T^{F''}$, which is upper bounded by T^F. The proof follows from the proof of the Prop. 1, the minimal speed towards F on the intersection of $R_N(a,b)$ and $x_i = d^j_*$ is lower bounded by $\frac{b_i - d^j_*}{b_i - a_i}\overline{s_F} + \frac{d^j_* - a_i}{b_i - a_i} s_F$. As the actual minimal speed could be higher than $\frac{b_i - d^j_*}{b_i - a_i}\overline{s_F} + \frac{d^j_* - a_i}{b_i - a_i} s_F$ and other multi-affine feedbacks could solve the same problem on $R_N(a,b^*)$ and $R_N(a^*,b)$ with lower time bounds, when $i = j$, partitioning results in tighter time bounds.

(b)$\{i \neq j\}$ The multi-affine feedback k solves the control-to-set-of-facets problem on $R_N(a,b^*)$ for exit facets $\mathcal{E}^1 = \{F', F^*\}$ where $n_{F'} = e_i$ and $n_{F^*} = e_j$. Moreover, k solves control-to-set-of-facets problem on $R_N(a^*,b)$ for exit facets $\mathcal{E}^2 = \{F'', F^{**}\}$ where $n_{F''} = e_i$ and $n_{F^{**}} = -e_j$. Then the corresponding time bounds $T^{\mathcal{E}^1}$ and $T^{\mathcal{E}^2}$ are upper bounded by T^F by Cor. 1. However, $T^{F'}$ or $T^{F''}$ could be higher than T^F, hence, the costs of the resulting automaton states could be higher than $J(s)$.

In *(a)* and *(b)*, the effects of partitioning are analyzed on a rectangular region for a simple case where the initial rectangle has a solution to the control-to-facet problem for facet F. It is concluded that when a rectangle $R_N(a,b)$ of a state $s \in S_P$ with $J(s)$ is partitioned, the costs of the resulting states s' and s'' can be higher or lower than $J(s)$. Hence, even for that simple case, partitioning can have negative and positive effects on the defined time bound for a single rectangle. Moreover, there is no closed form relationship between the partitioning scheme $\{d^j_i\}_{i \in \mathbb{N}}$ and the volume of the set $\mathbb{X}_0$.

In order to overcome these difficulties, we use a Particle Swarm Optimization (PSO)(Trelea (2003)) algorithm to find the new thresholds. The objective of the optimization is maximizing the volume of the set $\mathbb{X}_0$ (24). We run the PSO algorithm iteratively. At each iteration, a new threshold d^j_{i*} is added between two consecutive ones d^j_i, d^j_{i+1} depending on the distance between them and the value of the corresponding optimization variable. An optimization variable for d^j_{i*} is defined with range $[d^j_i, d^j_{i+1} - d]$ if the distance between two consecutive thresholds is twice as large as the minimum allowed edge size, $2d < d^j_{i+1} - d^j_i$. Part of the range $[d^j_i, d^j_i + d)$ is used to decide whether to add the threshold or not, i.e. a new threshold is added only if $d^j_{i*} \in [d^j_i + d, d^j_{i+1} - d]$. The dimension of the optimization problem depends on the grid configuration $\{d^j_i\}_{i \in \mathbb{N}}$ of the iteration. The iterative procedure terminates when either all the intervals are smaller than $2d$ or there is no change in the optimum objective value for the last two iterations.



(a) (b)

Fig. 4. (a) FSA $\mathcal{A}$ that accepts the language satisfying Φ (27) (T stands for Boolean constant true). The initial state of the automaton is filled with grey and the final state is marked with a double circle. (b) The initial partition induced by the predicate set Π. π_0, π_1, π_2, π_3 and π_4 are satisfied in cyan, magenta, red, green and orange colored rectangles, respectively and π_5 is satisfied in white rectangles.

Remark 4. Let $d^j = |\{d^j_i\}_{i \in \mathbb{N}}|$, then the cardinality of the resulting partition is $\Pi^N_{j=1}(d^j - 1)$. Construction of the transition system $\mathcal{T}$ (see Sec. 5) from the partition $\{d^j_i\}_{i \in \mathbb{N}}$ requires to solve $(2^{2N} - 1)\Pi^N_{j=1}(d^j - 1)$ linear programs. For each partition, in addition to solving these linear programs, we take the product between $\mathcal{T}$ and $\mathcal{A}$, and run Alg. 1 to find the volume of the set $\mathbb{X}_0$.

7. CASE STUDY

Consider the following multi-affine system

$$\begin{aligned} \dot{x}_1 &= -x_1 + x_1 x_2 + u \\ \dot{x}_2 &= -x_2 + x_1 x_2 + u, \end{aligned} \tag{26}$$

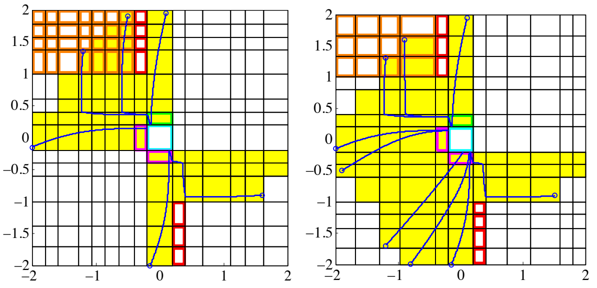
where the state x and the control input u are constrained to sets $R_N(a^{\mathbb{X}}, b^{\mathbb{X}}) = [-2, 2] \times [-2, 2]$ and $U = [-1, 1]$, respectively. The specification is to visit one of the rectangles that satisfy π_1 or π_3, then a rectangle where π_0 is satisfied, while always avoiding the rectangles that satisfy π_2. Moreover, if a trajectory visits a rectangle where π_4 is satisfied, then it has to visit a rectangle that satisfies π_3 before visiting a rectangle that satisfies π_0. Predicates π_i, $i = 0, \ldots, 4$ are defined in Fig. 4b. Formally, this specification translates to the following scLTL formula Φ over $\Pi = \{\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5\}$:

$$\Phi = ((\neg\pi_4 \mathcal{U} \pi_0) \vee (\neg\pi_0 \mathcal{U} \pi_3)) \wedge (\neg\pi_2 \mathcal{U} \pi_0) \wedge (\neg\pi_0 \mathcal{U} (\pi_1 \vee \pi_3)) \tag{27}$$

A FSA $\mathcal{A}$ that accepts the language satisfying formula Φ is given in Fig. 4a. The regions of interests and the corresponding partition are given in Fig. 4b. The upper time bound to satisfy the specification is set to $T = 2.5$, the minimum edge length is set to $d = 0.2$ and the robustness parameter for optimization problems (19) and (20) is set to $\epsilon = 0.2$.

To illustrate the main results of the paper, we use two approaches to generate a control strategy. In the first experiment, only control-to-facet problems are considered, hence a deterministic transition system is used. As discussed in the paper, the resulting product automaton is also deterministic and it is sufficient to use a shortest path algorithm instead of Alg. 1. In the second approach, both control-to-facet and control-to-set-of-facets problems are considered. Hence, the resulting transition system and product automaton are non-deterministic, and Alg. 1 is applied.

We use $(Q_0^d, \mathcal{A}_d^C)$ and $(Q_0^{nd}, \mathcal{A}_{nd}^C)$ to denote the control strategies as defined in Sec. 5 for the partition schemes resulted from the iterative refinement described in Sec.6 for the first and second approach, respectively. We use $\mathbb{X}_0^d$ and $\mathbb{X}_0^{nd}$ to denote the corresponding sets of initial states of system(26), respectively. These sets, together with sample trajectories of the closed loop systems, are shown in Fig. 5. The volume of $\mathbb{X}_0^d$ is 5.25 and the volume of $\mathbb{X}_0^{nd}$ is 7.62. A control-to-facet problem on a rectangle $R_2(a,b) \subseteq [-2,-0.2] \times [-2,-0.2]$ does not have a solution for facets F^{e_1} and F^{e_2} because of the strong drift in that region. However, rectangles in the same region have solutions to control-to-set-of-facets problem for $\mathcal{E} = \{F^{e_1}, F^{e_2}\}$. Consequently, rectangles in that region is only covered by $\mathbb{X}_0^{nd}$ as the construction of $\mathbb{X}_0^{nd}$ considers non-determinism.



(a)

(b)

Fig. 5. The yellow regions in (a) and (b) represent $\mathbb{X}_0^d$ and $\mathbb{X}_0^{nd}$, respectively. Some simulated satisfying trajectories of the corresponding closed-loop systems are shown (the initial states are marked by circles).

8. CONCLUSION

We studied a time-constrained control problem for a continuous-time multi-affine system from a specification given as a syntactically co-safe LTL formula over a set of predicates in its state variables. Our approach was based on finding an optimal control strategy on the product between an abstraction of the system and an automaton enforcing the satisfaction of the specification. The abstraction was a weighted transition system constructed by solving facet reachability problems on a rectangular partition of the state space of the original system. We proposed an iterative refinement procedure via a random optimization algorithm to increase the set of admissible initial states.

REFERENCES

E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *Hybrid Systems: Computation and Control*, pages 19–30. Springer, 2009.

C. Belta. On controlling aircraft and underwater vehicles. In *IEEE International Conference on Robotics and Automation*, volume 5, pages 4905 – 4910, 2004.

C. Belta and L.C.G.J.M. Habets. Control of a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control*, 51(11):1749 –1759, 2006.

A. Bhatia, L. E. Kavraki, and Moshe Y. Vardi. Motion planning with hybrid dynamics and temporal goals. In *IEEE Conference on Decision and Control*, pages 1108–1115, 2010.

H. de Jong. Modeling and simulation of genetic regulatory systems. *J. Comput. Biol.*, 9(1):69–105, 2002.

H. Kress Gazit, G. Fainekos, and G. J. Pappas. Where's Waldo? Sensor-based temporal logic motion planning. In *IEEE International Conference on Robotics and Automation*, 2007.

A. Girard. Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Hybrid Systems: Computation and Control*, pages 111–120. ACM, 2010a.

A. Girard. Synthesis using approximately bisimilar abstractions: time-optimal control problems. In *IEEE Conference on Decision and Control*, pages 5893 –5898, 2010b.

L.C.G.J.M. Habets, M. Kloetzer, and C. Belta. Control of rectangular multi-affine hybrid systems. In *IEEE Conference on Decision and Control*, pages 2619 –2624, 2006.

M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53 (1):287 –297, 2008a.

M. Kloetzer and C. Belta. Dealing with nondeterminism in symbolic control. In *Hybrid Systems: Computation and Control*, pages 287–300. Springer-Verlag, 2008b.

O. Kupferman and M. Y. Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19:291–314, 2001.

T. Latvala. Efficient model checking of safety properties. In *In Model Checking Software. 10th International SPIN Workshop*, pages 74–88. Springer, 2003.

A. Lotka. *Elements of physical biology.* Dover Publications, Inc., New York, 1925.

M. Mazo and P. Tabuada. Symbolic approximate time-optimal control. *Systems and Control Letters*, 60(4):256 – 263, 2011.

H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag, 1990.

P. Tabuada and G. Pappas. Model checking LTL over controllable linear systems is decidable. In *Lecture Notes in Computer Science.* Springer-Verlag, 2003.

I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, pages 317 – 325, 2003.

V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

T. Wolfgang. Infinite games and verification. In *Computer Aided Verification*, pages 58–65. Springer Berlin / Heidelberg, 2002.

T. Wongpiromsarn, U. Topcu, and R. M. Murray. Receding horizon temporal logic planning for dynamical systems. In *IEEE Conference on Decision and Control*, pages 5997–6004, 2009.