Automatic synthesis of switching controllers for linear hybrid systems: Reachability control

Massimo Benerecetti and Marco Faella

Università di Napoli "Federico II", Italy

Abstract. We consider the problem of computing the controllable region of a Linear Hybrid Automaton with controllable and uncontrollable transitions, w.r.t. a reachability objective. We provide an algorithm for the finite-horizon version of the problem, based on computing the set of states that must reach a given non-convex polyhedron while avoiding another one, subject to a polyhedral constraint on the slope of the trajectory.

1 Introduction

Hybrid systems are non-linear dynamic systems, characterized by the presence of continuous and discrete variables. Hybrid automata [5] are the most common syntactic variety of hybrid system: a finite set of locations, similar to the states of a finite automaton, represents the value of the discrete variables. The current location, together with the current value of the (continuous) variables, form the instantaneous description of the system. Change of location happens via discrete transitions, and the evolution of the variables is governed by differential inclusions attached to each location. In a Linear Hybrid Automaton (LHA), the allowed differential inclusions are of the type $\dot{\mathbf{x}} \in P$, where $\dot{\mathbf{x}}$ is the vector of the first derivatives of all variables and $P \subseteq \mathbb{R}^n$ is a convex polyhedron. Notice that differential inclusions are non-deterministic, allowing for infinitely many solutions with the same starting conditions.

We study LHAs whose discrete transitions are partitioned into controllable and uncontrollable ones, giving rise to a 2-player model called Linear Hybrid Game (LHG): on one side the controller, which can only issue controllable transitions; on the other side the environment, which can choose the trajectory of the variables and can take uncontrollable transitions whenever they are enabled.

As control goal, we consider reachability, i.e., the objective of reaching a given set of target states. It is easy to show that the reachability control problem is undecidable, being harder than the standard reachability verification (i.e., 1-player reachability) for LHAs [6] . We present the first exact algorithm for 1-step controllability under a reachability objective, namely reaching the target region with at most one discrete transition. In turn, this provides an exact algorithm for bounded-horizon reachability control (i.e., reaching the target within a fixed number of discrete steps), and a semi-algorithm[1] for the infinite-horizon case. This extended abstract summarizes the results of [2].

[1] In other words, a procedure that may or may not terminate, and that provides the correct answer whenever it terminates.

Although the control goal we examine, as a language of infinite traces, is the dual of safety, the corresponding synthesis problem is not, because our game model is asymmetric: choice of continuous-time trajectories is uncontrollable. Hence, it is not possible to solve the control problem with reachability goal T by exchanging the roles of the two players and then solving the safety control problem with goal $\overline{T}$ (i.e., the complement of T).

On the one hand, the 1-step safety control problem can be solved by computing the may-reach-while-avoiding operator $\mathit{RWA}^{\mathrm{m}}$ [3, 4]. Given two sets of states U and V, $\mathit{RWA}^{\mathrm{m}}(U, V)$ collects the states from which there exists a system trajectory that reaches U while avoiding V at all times. On the other hand, the 1-step *reachability* control problem requires a different operator, called must-reach-while-avoiding and denoted by $\mathit{RWA}^{\mathrm{M}}(U, V)$. As suggested by its name, such operator computes the set of states from which *all* system trajectories reach U while avoiding V. The main technical result of this paper is that the first operator can be used to compute the latter, once a suitable over-approximation of $\mathit{RWA}^{\mathrm{M}}(U, V)$ is available (Theorem 2). Moreover, we present an effective way to obtain such an over-approximation.

To the best of our knowledge, the reachability control goal was never considered for LHGs. In the full paper [2], we present the proofs for all results and an experimental evaluation based on an extension to the tool SpaceEx.

2 The model: Linear hybrid games

A *convex polyhedron* is a subset of $\mathbb{R}^n$ that is the intersection of a finite number of strict and non-strict affine half-spaces. A *polyhedron* is the union of a finite number of convex polyhedra. Given an ordered set $X = \{x_1, \ldots, x_n\}$ of variables, a *valuation* is a function $v : X \to \mathbb{R}$. Let $\mathit{Val}(X)$ denote the set of valuations over X. There is an obvious bijection between $\mathit{Val}(X)$ and $\mathbb{R}^n$, allowing us to extend the notion of (convex) polyhedron to sets of valuations. We denote by $\mathit{CPoly}(X)$ (resp., $\mathit{Poly}(X)$) the set of convex polyhedra (resp., polyhedra) on X. Let A be a set of valuations, states or points in $\mathbb{R}^n$, we denote by $\overline{A}$ its complement.

We use $\dot{X}$ to denote the set $\{\dot{x}_1, \ldots, \dot{x}_n\}$ of dotted variables, used to represent the first derivatives, and X' to denote the set $\{x'_1, \ldots, x'_n\}$ of primed variables, used to represent the new values of variables after a transition. A *trajectory* over X is a function $f : \mathbb{R}^{\geq 0} \to \mathit{Val}(X)$ that is differentiable but for a finite subset of $\mathbb{R}^{\geq 0}$.

Definition 1. *A* Linear Hybrid Game (LHG) $(\mathit{Loc}, X, \mathit{Edg}_{\mathrm{c}}, \mathit{Edg}_{\mathrm{u}}, \mathit{Flow}, \mathit{Inv}, \mathit{Init})$ *consists of the following components:* (i) *a finite set Loc of* locations*;* (ii) *a finite set* $X = \{x_1, \ldots, x_n\}$ *of real-valued* variables*;* (iii) *two sets* $\mathit{Edg}_{\mathrm{c}}, \mathit{Edg}_{\mathrm{u}} \subseteq \mathit{Loc} \times \mathit{Poly}(X \cup X') \times \mathit{Loc}$ *of* controllable *and* uncontrollable transitions*, respectively;* (iv) *a mapping* $\mathit{Flow} : \mathit{Loc} \to \mathit{CPoly}(\dot{X})$*, called the* flow constraint*;* (v) *a mapping* $\mathit{Inv} : \mathit{Loc} \to \mathit{Poly}(X)$*, called the* invariant*;* (vi) *a mapping* $\mathit{Init} : \mathit{Loc} \to \mathit{Poly}(X)$*, contained in the invariant, defining the* initial states *of the automaton.*

A *state* is a pair $\langle l, v \rangle$ of a location l and a valuation $v \in \mathit{Val}(X)$. Transitions describe instantaneous changes of location, in the course of which the variables may change their value. Each transition $(l, J, l') \in \mathit{Edg}_{\mathrm{c}} \cup \mathit{Edg}_{\mathrm{u}}$ consists of a *source location* l, a *target location* l', and a *jump relation* $J \in \mathit{Poly}(X \cup X')$, that specifies how the variables may change their value during the transition. The projection of J on X contains the valuations for which the transition is enabled, a.k.a. a *guard*. The flow constraint attributes to each location a set of valuations over the first derivatives of the variables, which determines how variables can change over time. We let $\mathit{InvS} = \bigcup_{l \in \mathit{Loc}} \{l\} \times \mathit{Inv}(l)$ and $\mathit{InitS} = \bigcup_{l \in \mathit{Loc}} \{l\} \times \mathit{Init}(l)$. Notice that InvS and InitS are sets of states. Given a set of states A and a location l, we denote by $A\lfloor_l$ the projection of A on l.

The behavior of an LHG is based on two types of steps: *discrete* steps correspond to the *Edg* component, and produce an instantaneous change in both the location and the variable valuation; *timed* steps describe the change of the variables over time in accordance with the *Flow* component. A *joint step* is either a timed step of infinite duration, or a timed step of finite duration followed by a discrete step. The controller influences the system through a *strategy*. Strategies assign to each controllable transition a possibly non-convex polyhedron, which is contained in the jump relation of the transition. The intended meaning is that the strategy restricts controllable transitions so that they can be taken from a given subset of their original guard and they lead to a given subset of their original set of destinations. Formal definitions are included in the full paper [2].

Reachability control problem. Given an LHG and a set of states $T \subseteq \mathit{InvS}$, the *reachability control problem* asks whether there exists a strategy σ such that, for all initial states $s \in \mathit{InitS}$, all runs that start from s and are consistent with σ reach some state in T. We call the above σ a *winning strategy*.

3 Solving the reachability control problem

The following theorem states the general procedure for solving the reachability control problem, based on the *controllable predecessor operator for reachability* $\mathit{CPre}^{\mathrm{R}}$. For a set of states A, the set $\mathit{CPre}^{\mathrm{R}}(A)$ contains all states from which the controller can ensure that the system reaches A within the next joint step. In discrete games, the CPre operator used for solving reachability games is the same as the one used for the safety goal [7]. In both cases, when the operator is applied to a set of states T, it returns the set of states from which Player 1 can force the game into T in one step. In hybrid games, the situation is different: a joint step represents a complex behavior, extending over a (possibly) non-zero time interval. While the CPre for reachability only requires T to be visited once during such interval, CPre for safety requires that the entire behavior constantly remains in T. Hence, we present a novel algorithm for computing $\mathit{CPre}^{\mathrm{R}}$.

The following theorem states that the least fixpoint of the operator $\tau(X) \triangleq T \cup \mathit{CPre}^{\mathrm{R}}(X)$ provides a solution of the reachability control problem. Intuitively, each application of τ extends (backward), by adding a single joint step, all the runs compatible with some winning strategy for the controller.

Theorem 1. *Let T be a polyhedron and $W^* = \mu W \,.\, T \cup \mathit{CPre}^{\mathrm{R}}(W)$, where μ denotes the least fixpoint. If the fixpoint is obtained in a finite number of iterations then the answer to the reachability control problem for target set T is positive if and only if $\mathit{InitS} \subseteq W^*$.*

Computing the predecessor operator. In order to compute the predecessor operator, we introduce the *Must Reach While Avoiding* operator, denoted by $\mathit{RWA}^{\mathrm{M}}$. Given a location l and two sets of variable valuations U and V, $\mathit{RWA}_l^{\mathrm{M}}(U, V)$ contains the set of valuations from which all continuous trajectories of the system reach U while avoiding V [2].

We can now reformulate $\mathit{CPre}^{\mathrm{R}}$ by means of the operator $\mathit{RWA}_l^{\mathrm{M}}$. Let B_l be the set of states of location l, where the environment can take a discrete transition leading outside A and, similarly, C_l be the set of states of l, where the controller can take a discrete transition leading to A. A state s of location l belongs to $\mathit{CPre}^{\mathrm{R}}(A)$ if the controller can force the system into A within one joint step, no matter what the environment does. This occurs if, for every possible trajectory chosen by the environment, one of the following conditions holds: (i) the system reaches $A\lfloor_l$ while avoiding $B_l \setminus A\lfloor_l$, thus without giving the environment any chance to take an action leading outside A; (ii) the system reaches a point in $C_l \setminus B_l$, from where the controller can force the system into A, while avoiding $B_l \setminus A\lfloor_l$; or (iii) the trajectory exits from the invariant $\mathit{Inv}(l)$ meanwhile avoiding $B_l \setminus A\lfloor_l$. In this last case, the semantics ensures that, before the trajectory exits from the invariant, the environment will take some discrete transition, which can only lead to A (see *well-formedness* in the full paper).

The following lemma formalizes the above intuition. We say that a set of states A is *polyhedral* if for all $l \in \mathit{Loc}$, the projection $A\lfloor_l$ is a polyhedron.

Lemma 1. *If $A \subseteq \mathit{InvS}$ is polyhedral, the following holds:*

$$\mathit{CPre}^{\mathrm{R}}(A) = \mathit{InvS} \cap \bigcup_{l \in \mathit{Loc}} \{l\} \times \mathit{RWA}_l^{\mathrm{M}}\big(A\lfloor_l \cup C_l \setminus B_l \cup \overline{\mathit{Inv}(l)}, B_l \setminus A\lfloor_l\big).$$

Computing $\mathit{RWA}^{\mathrm{M}}$. We show how to compute $\mathit{RWA}^{\mathrm{M}}$ based on the operator which is used to solve *safety* control problems: the *May Reach While Avoiding* operator $\mathit{RWA}_l^{\mathrm{m}}(U, V)$, returning the set of states from which *there exists* a trajectory that reaches U while avoiding V. In safety control problems, $\mathit{RWA}^{\mathrm{m}}$ is used to compute the states from which the environment may reach an unsafe state (in U) while avoiding the states from which the controller can take a transition to a safe state (in V). Notice that $\mathit{RWA}^{\mathrm{m}}$ is a classical operator, known as *Reach* [8], *Unavoid_Pre* [1], and *flow_avoid* [9]. We recently gave the first sound and complete algorithm for computing it on LHGs [4].

In the rest of this section, we consider a fixed location $l \in \mathit{Loc}$ and we omit the l subscript whenever possible. For a polyhedron G and $p \in G$, we say that p is *t-bounded in* G if all admissible trajectories starting from p eventually exit from G. We denote by $\textit{t-bnd}(G)$ the set of points of G that are t-bounded in it, and we say that G is *t-bounded* if all points $p \in G$ are t-bounded in G.

[2] In the temporal logic CTL, we have $\mathit{RWA}^{\mathrm{M}}(U, V) \equiv \forall \overline{V} \; \mathcal{U} \; (U \wedge \overline{V})$.

We now show how to relate $\mathit{RWA}^{\mathrm{M}}$ and $\mathit{RWA}^{\mathrm{m}}$, by exploiting the following idea. First, notice that all points in U belong to $\mathit{RWA}^{\mathrm{M}}(U, V)$ by definition. Now, the content of $\mathit{RWA}^{\mathrm{M}}(U, V)$ can be partitioned into two regions: the first region is U; the second region must be t-bounded, because each point in the second region must eventually reach U. If we can find a polyhedron Over that over-approximates $\mathit{RWA}^{\mathrm{M}}(U, V)$ and such that $\mathit{Over} \setminus U$ is t-bounded, we can use $\mathit{RWA}^{\mathrm{m}}$ to refine it. Precisely, we can use $\mathit{RWA}^{\mathrm{m}}$ to identify and remove the points of Over that may leave Over without hitting U first.

If $\mathit{Over} \setminus U$ is not t-bounded, the above technique does not work, because $\mathit{RWA}^{\mathrm{m}}$ cannot identify (and remove) the points that may remain forever in Over without ever reaching U. This idea is formalized by the following result.

Theorem 2. *For all disjoint polyhedra U and V, such that V is convex, let Over be a polyhedron such that:* (i) $\mathit{RWA}^{\mathrm{M}}(U, V) \subseteq \mathit{Over} \subseteq \overline{V}$ *and* (ii) $\mathit{Over} \setminus U$ *is t-bounded. Then,* $\mathit{RWA}^{\mathrm{M}}(U, V) = \mathit{Over} \setminus \mathit{RWA}^{\mathrm{m}}(\overline{\mathit{Over}}, U)$.

It can be proved that the set $\mathit{Over} \triangleq U \cup \mathit{t\text{-}bnd}(\overline{U} \cap \overline{V})$ is computable and satisfies the conditions of Theorem 2. The full paper [2] also provides an alternative, more efficient, version for Over. In conclusion, the results sketched above provide an effective solution to the bounded-horizon reachability control problem and a semi-algorithm for the infinite-horizon version.

References

1. A. Balluchi, L. Benvenuti, T. Villa, H. Wong-Toi, and A. Sangiovanni-Vincentelli, "Controller synthesis for hybrid systems with a lower bound on event separation," *Int. J. of Control*, vol. 76, no. 12, pp. 1171–1200, 2003.
2. M. Benerecetti and M. Faella, "Automatic synthesis of switching controllers for linear hybrid systems: Reachability control," *ACM Trans. on Embedded Computing Systems*, vol. 16, no. 4, 2017.
3. M. Benerecetti, M. Faella, and S. Minopoli, "Revisiting synthesis of switching controllers for linear hybrid systems," in *Proc. of the 50th IEEE Conf. on Decision and Control.* IEEE, 2011.
4. ——, "Automatic synthesis of switching controllers for linear hybrid systems: Safety control," *Theoretical Computer Science*, vol. 493, pp. 116–138, 2013.
5. T. Henzinger, "The theory of hybrid automata," in *11th IEEE Symp. Logic in Comp. Sci.*, 1996, pp. 278–292.
6. T. Henzinger, P. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *J. of Computer and System Sciences*, vol. 57, no. 1, pp. 94 – 124, 1998.
7. O. Maler, "Control from computer science," *Annual Reviews in Control*, vol. 26, no. 2, pp. 175–187, 2002.
8. C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proc. of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.
9. H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *36th IEEE Conf. on Decision and Control.* San Diego, CA: IEEE, 1997, pp. 4607 – 4612.