Information and Computation 257 (2017) 114–138

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Tracking smooth trajectories in linear hybrid systems ☆

Massimo Benerecetti, Marco Faella *

Dept. of Electrical Engineering and Information Technology, Università di Napoli "Federico II", Via Claudio 21, 80125 Napoli, Italy

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 5 July 2017
Available online 31 October 2017

Keywords:
Hybrid automata
Reachability
Controller synthesis

ABSTRACT

We analyze the properties of smooth trajectories subject to a constant differential inclusion which constrains the first derivative to belong to a given convex polyhedron. We present the first exact symbolic algorithm that computes the set of points from which there is a trajectory that reaches a given polyhedron while avoiding another (possibly non-convex) polyhedron. We prove that this set of points remains the same if the smoothness constraint is replaced by a weaker differentiability constraint, but not if it is replaced by almost everywhere differentiability. We discuss the connection with (Linear) Hybrid Automata and in particular the relationship with the classical algorithm for reachability analysis for Linear Hybrid Automata.

1. Introduction

Hybrid Automata are a mathematical abstraction of systems that feature both discrete and continuous dynamics. Linear Hybrid Automata (LHAs) [1] were introduced as a computationally tractable model of hybrid systems that still allows for non-trivial dynamics. In particular, LHAs can approximate complex dynamics up to an arbitrary precision [2].

In an LHA, discrete dynamics is represented by a finite set of control modes called *locations*, while the continuous dynamics is embodied by a finite set of real-valued variables. In each location, the continuous dynamics is constrained by a differential inclusion of the type $\dot{x} \in F$, where $\dot{x}$ is the vector of the time-derivatives of all the variables in the system, and $F \subseteq \mathbb{R}^n$ is a convex polyhedron. The main decision problem that was considered for LHAs is *reachability*, i.e., given two system configurations, say an initial state and an error state, establish whether there is a system behavior that leads from the first to the second. A more complex task consists in verifying whether a given LHA can be modified (i.e., *controlled*) in such a way that a given error configuration (or region) is *not* reached by any behavior. This problem can be called *safety control* and is analogous to a game with a safety objective. Both problems require an algorithm for the following sub-problem, which applies to a single discrete location: given a region G (for *goal*) and a region A (for *avoid*) of system configurations, find the set of points from which there is a trajectory that reaches G while avoiding A at all times. We denote this set by $RWA(G, A)$ for *reach while avoiding*. In reachability problems, the goal region G can be thought of as comprising error states, and the avoidance region A is the complement of the *invariant* of the automaton, which is the set of configurations that make physical sense for the system. Hence, $RWA(G, A)$ is the set of states that reach an error state while remaining in the invariant. In a safety control problem, the goal region G is taken to be a set of uncontrollable states (such as, states outside the safe region) and A is a set of controllable states (included in the invariant). Then, $RWA(G, A)$ identifies the region in which the environment can reach an error state while avoiding the good, controllable states.

☆ This work was partially supported by the INdAM-GNCS project "Logica e Automi per il Model-Checking Intervallare".
* Corresponding author.
E-mail addresses: massimo.benerecetti@unina.it (M. Benerecetti), m.faella@unina.it (M. Faella).

https://doi.org/10.1016/j.ic.2017.10.004
0890-5401/

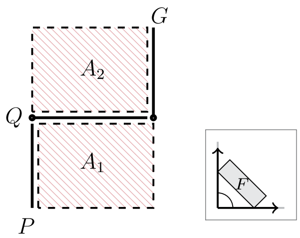

Fig. 1. Can the points in P reach the target region G while remaining in $P \cup Q \cup G$? The flow constraint F is shown on the r.h.s.

The *RWA* operator is recognized as a central tool in the analysis of various kinds of hybrid systems: it corresponds to the *Reach* operator in Tomlin et al. [3] and *Unavoid_Pre* in Balluchi et al. [4]; it was also used in the synthesis of controllers for reachability objectives [5,6].

Computing reach-while-avoiding The algorithmic computation of $RWA(G, A)$ is simple when A is co-convex, corresponding to the case of reachability analysis for LHAs with convex invariants. In that case, *RWA* can be expressed in the first-order theory of reals and computed using a constant number of basic polyhedra operations.

When A is not co-convex, one may adapt the procedure that is presented in one of the early papers on LHAs, in the context of reachability analysis in presence of non-convex invariants [1]. The idea of the algorithm is simple: consider a partition of the non-convex invariant I into a finite set of convex polyhedra $P_1, \dots, P_n$; then, split the location with invariant I into n different locations, each with convex invariant P_i; finally, connect these new locations with virtual transitions corresponding to the boundaries between two adjacent convex polyhedra P_i, P_j. Because of the added virtual transitions, this approach naturally leads to trajectories that are *almost everywhere (a.e.) differentiable*.

Consider for example the situation depicted in Fig. 1. Assume that the invariant for the current location is $P \cup Q \cup G$ and the goal is to reach G. Dashed lines identify topologically open sides of polyhedra. The flow constraint F is also depicted in the figure: it allows trajectories to move in a range of directions going from straight right to straight up, and it forbids stopping (i.e., it does not include the origin).

The above procedure splits the invariant into three convex polyhedra, and then performs a backward reachability analysis which starts from the goal G and progressively enlarges the set of "good" states W by including the states that can reach W while remaining in one of the convex parts of the invariant.

In our example, the points in the line segment Q can reach the target by moving straight to the right, while remaining in one convex part of the invariant and, similarly, points in the line segment P can reach the extreme point of Q by moving straight up. Hence, both Q and P end up on the final solution. On the other hand, no differentiable trajectory can start in a point of P and reach G while remaining in $P \cup Q \cup G$.[1]

In some scenarios, such as the one we describe below, restricting the system trajectories to be differentiable, or even smooth (i.e., differentiable an arbitrary number of times at all times), may be desirable to ensure that certain physical constraints are satisfied. In this paper, we present an exact algorithm for computing $RWA(G, A)$ for general polyhedra G and A with respect to smooth trajectories. Moreover, we prove that differentiable and smooth trajectories lead to equivalent notions of *RWA*.

Applications The difference between RWA under smooth trajectories and under a.e. differentiable ones for LHAs only surfaces when the avoidance region A is not topologically closed. To see how this case may be relevant to applications, consider the example in Fig. 2(a), where multiple robots (such as Kiva Systems[2]) must visit a target region G at the same time. The robots can move between two free roaming areas connected by two linear, intersecting tracks, and, for safety reasons, must always keep a minimum distance between each other. In this scenario, a topologically open avoidance region can be used to model the two intersecting tracks, as shown in Fig. 3. Requiring smoothness (or even plain differentiability) of the allowed trajectories ensures that robots cannot switch track at the intersection. The process of moving along a track could also be abstracted away by a discrete mode change (i.e., robots would seem to jump from one end of the track to the other end), but this prevents the explicit modeling of some phenomena, such as the possible collisions between two robots traveling on different tracks.

Fig. 2(b) shows a different interpretation of the same scenario, in which some ships have to reach the target region G, using two narrow intersecting channels. The small width of the channels compared to the size of the ships prevents the ships from turning at any point, including the point where the channels intersect. Once again, the topologically open avoidance region shown in Fig. 3 is a convenient modeling technique for this scenario.

[1] By rotating the polyhedra in Fig. 1 (including the flow constraint F) by 45°, it becomes apparent that the issue also occurs with *rectangular* flow constraints. However, Rectangular Hybrid Automata and Games [7] do not exhibit the above issue, due to the presence of multiple restrictions, such as the fact that guards and invariants are convex and topologically closed.

[2] A commercial robotic platform for warehouse automation: http://www.kivasystems.com.