The d/dt Tool for Verification of Hybrid Systems

Eugene Asarin, Thao Dang, and Oded Maler

VERIMAG, 2, av de Vignate, Centre Equation, 38610 Gières, France.
$\{asarin, tdang, maler\}$@imag.fr

Abstract. In this paper we describe the tool **d/dt** which provides automatic safety verification of hybrid systems with linear continuous dynamics with uncertain input. The verification procedure is based on a method for overapproximating reachable sets by orthogonal polyhedra. The tool also allows to synthesize a controller which switches the system between continuous modes in order to satisfy a safety specification.

1 Introduction

Hybrid systems are dynamical systems whose state space is based on discrete variables, evolving by taking transitions, and continuous variables evolving according to some differential equation that depends on the discrete variables. When hybrid systems were first introduced to the verification community [16], they were viewed as an extension of timed systems in the sense that while in the latter all the continuous variables are clocks with a uniform slope in every discrete state, the former admit variables that may evolve according to an arbitrary continuous dynamics. The algorithmic verification approach for hybrid systems [2], developed together with that for timed systems, is based on the following principles:

1. Global configurations of a system consist of tuples of the form $(q, \mathbf{x})$ where q is a discrete state and $\mathbf{x} \in \mathsf{R}^n$ is a point in the continuous state space. Sets of reachable configurations can be written as unions of tuples of the form (q, F) where F is a subset of R^n. Such tuples are called symbolic states.
2. On such symbolic states one can define successor (or predecessor) operators that decompose into *transition-successor* and *time-successor*.
3. Equipped with a representation scheme for symbolic states and an implementation of these operators, one can apply standard verification algorithms.

The decidability results for timed automata [1] are based on the fact that there is a *finite* class of simple subsets of R^n which is closed under the above operations. The elements of this class, which are roughly sets definable by finite Boolean combinations of inequalities of the form $x_i \leq k$ or $x_i - x_j \leq k$ for an integer k, admit an efficient representation using difference bound matrices to which the time-successor operator can be easily applied (the computation of transition-successors is usually reduced to applying Boolean operations and does not pose major additional problems). This approach has been implemented in timed automata verification tools such as Kronos [19] and Uppaal [15]. The tool HyTech [13] took this idea further and applied to the more general class of "linear hybrid automata", in which the slope of every continuous variable is constant (but arbitrary) in every discrete state. Although the continuous dynamics in every discrete state is trivial, the combination with discrete transitions (and changes of the slope) makes the verification problem undecidable, apart from some special cases [12]. However, one can still compute the time-successors of a convex polyhedron using linear algebra, but the class of such polyhedra is infinite and the verification algorithms are not guaranteed to terminate. The tool presented in this paper attempts to go further and apply reachability-based verification techniques to systems where the continuous dynamics is defined by a differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$. Under such dynamics, the time-successors of a set usually form curved objects and, except for some special cases [17], cannot be computed exactly. Instead, we take an approximation approach, that is, for a given initial polyhedron F we compute another polyhedron which contains all the time-successors of F. In [11] we presented a method to do so for arbitrary f, while in this tool we concentrate on the class of systems with linear continuous dynamics. Our reachability techniques can be seen as an extension of numerical integration from points to polyhedral sets and they are of interest even for purely continuous systems.

An outline of the paper is as follows. Section 2 is devoted to a brief description of our reachability techniques, the kernel algorithms of **d/dt**, and their application to safety verification and switching controller synthesis for hybrid systems (see [3, 4] for more details). In Section 3 we present the tool **d/dt** and some case studies treated using the tool.

2 Reachability Techniques

We use hybrid automata [2] as a modeling formalism for hybrid systems. In the model we consider, continuous dynamics are *linear with uncertain, bounded input* of the form $f(\mathbf{x}) = A\mathbf{x} + B\mathbf{u}$ where $\mathbf{u}$ ranges inside a convex polyhedron. All the *staying conditions* (or "invariant") and *transition guards* are specified as conjunctions of linear inequalities. The *resets* associated with discrete transitions are *affine, set-valued maps* of the form $R(\mathbf{x}) = D\mathbf{x} + J$ where D is a matrix and J is a convex polyhedron.

To *prove a safety property* (never reaching a set of bad states), we overapproximate the reachable set of the system on a step-by-step basis and check for its intersection with the set of bad states. Given a time step r, each iteration k computes a convex polyhedron overapproximating the set of states reachable during the time interval $[kr, (k+1)r]$. While convex polyhedra allow to obtain reasonably good approximations for each time interval, the accumulation of reachable states often form a highly non-convex set. Therefore, to store reachable sets we use non-convex orthogonal polyhedra[1] [9] since they can be manipulated more

[1] Orthogonal polyhedra can be described as unions of closed full-dimensional hyper-rectangles.

easily than unions of arbitrary polyhedra. We consider first systems with continuous dynamics of the form $\dot{\mathbf{x}} = A\mathbf{x}$. Given an initial convex polyhedron F, the set F_r of reachable states at time r is the convex hull of the successors at time r of the vertices of F. The states reachable during the time interval $[0, r]$ are approximated by the convex hull $C = conv(F \cup F_r)$ which is enlarged by an appropriate amount to ensure overapproximation. Then, C is overapproximated by an orthogonal polyhedron. To handle staying conditions, we need to intersect F_r with the staying set and start the next iteration from the resulting polyhedron. For continuous dynamics with uncertain input $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}$, to compute F_r, we simulate the evolution of the faces of F. Since the dynamics of the normal vector of a face, governed by the adjoint system, is independent of the input, we use the Maximum Principle from optimal control to find the input that steers this face "farthest" allowing to cover all possible reachable states.

Concerning *switching controller synthesis*, the problem we consider is to find a controller that switches the system between continuous modes to avoid some bad states. The synthesis of such a controller is based on the *maximal invariant set* (i.e. the set of "winning" states). To compute this set, we make use of the *one-step predecessor* operator π. Given a set $\mathcal{F}$ of safe states, $\pi(\mathcal{F})$ consists of states from which the system *either* can stay in $\mathcal{F}$ indefinitely without switching *or* can stay in $\mathcal{F}$ for some time and then make a transition to another state which is still in $\mathcal{F}$. Intuitively, from all states which are not in $\pi(\mathcal{F})$ the system will leave $\mathcal{F}$ after not more than one switching; therefore, by iteratively removing all such states from $\mathcal{F}$ until convergence, we obtain an underapproximation of the maximal invariant set. Note that the operator π can be computed using our reachability operators for hybrid automata.

3 The tool d/dt

We present first the implementation of the tool and then some applications. One important component of our reachability algorithms are procedures for manipulating convex and orthogonal polyhedra. For common operations on convex polyhedra (e.g. Boolean operations, membership testing), we combine two standard libraries cdd and Qhull (which allows to better handle degenerate cases) and use the library Cubes [9] for operations on orthogonal polyhedra. In addition, we implemented the geometric operations specific to our approach (e.g. orthogonal approximation, intersection detection). For run-time visualization, the tool provides an option to interface with the 3D viewer Geomview. Given an input hybrid automaton, a safety specification, and optionally some user-defined approximation parameters (e.g. time step, granularity of orthogonal approximations), the tool can work in the following three modes: *reachability* (compute an overapproximation of the reachable set); *verification* (check whether the system can reach the bad set); *controller synthesis* (synthesize a safety switching controller by computing an underapproximation of the maximal invariant set).

We have successfully applied the tool to some case studies inspired from real-life applications. The first case study involves the verification of a longitudinal controller for platoons on automated highways. This controller consists of several modes with different control laws and we first proved the absence of collision for single modes [6] and also for a model with mode switching. For the latter model which has 4 continuous variables and 2 discrete states, the computation took 5 minutes on a Pentium 2. The second case study concerns a control method for under-actuated mechanical systems and its application to a double pendulum modeling the leg of a biped robot. As part of the design process we use the tool to find switching sequences which can drive the system to a desired periodic orbit [5]. We have also used the tool to analyze a model of bacterial quorum-sensing systems, more precisely, to determine whether the system can reach a equilibrium point corresponding to a steady state of luminescence [7].

4 Conclusion and Related Work

We have presented the tool **d/dt** for safety verification and switching controller synthesis for hybrid systems with linear differential inclusions. To our knowledge, at current time **d/dt** is the only existing tool that supports switching controller synthesis for hybrid systems. This work contributes a novel approach to the analysis and control of hybrid systems by exploiting the ideas of algorithmic methodology in computer science. In order to increase the applicability of **d/dt**, we intend to integrate in the tool our reachability technique for nonlinear continuous dynamics [11]. In addition, as verification is often expensive, we are considering an extension of the tool to include the analysis in a simulation fashion, that is, instead of exploring the whole state space, one can guide the reachability computation according to some strategy taking into account the property to be proved.

It is not easy to compare our tool to other tools in the domain for the following reason. The hybrid systems research is still young and, moreover, due to the complexity of the problem and the approximate nature of the solution, it is still hard to define performance measures and to compare tools according to standard benchmarks. In the remainder of the paper we discuss only the relationship between **d/dt** and two other tools: CheckMate [10] and VeriShift [8]. The reader is refered to [18] for a survey on hybrid systems verification tools. CheckMate can verify polyhedral invariant hybrid automata (where all the guard sets lie on the boundary of the staying sets and the reset maps are the identity). Unlike our tool, CheckMate takes an indirect approach, that is, it computes a finite-state abstraction of the original system using approximate reachability analysis and then verify the resulting discrete model. The reachability algorithm of CheckMate for linear continuous dynamics is similar to ours, but it is not easy to extend to systems with uncertain input. The tool VeriShift is designed to perform bounded time verification on hybrid automata with linear differential inclusions. It employs the ellipsoidal techniques [14] to approximate reachable sets. Note that in these tools reachable sets are represented in a non-canonical way (as unions of convex polyhedra or ellipsoids), which limits their applicability to high dimensional systems. The tool **d/dt** has been designed with generality in mind, and hence the problem of representing polyhedra of arbitrary dimension has been tackled and solved before the development of the rest of the algorithms. Therefore, one positive feature of **d/dt** is that it extends easily to more general systems (in terms of the complexity of dynamics and the dimensionality).

References

1. R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science* 126, 183–235, 1994.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine, The Algorithmic Analysis of Hybrid Systems, *Theoretical Computer Science* 138, 3–34, 1995.
3. E. Asarin, O. Bournez, T. Dang and O. Maler, Reachability Analysis of Piecewise-Linear Dynamical Systems, *HSCC*, 20-31 LNCS 1790, Springer, 2000.
4. E. Asarin, O. Bournez, T. Dang, O. Maler and A. Pnueli, Effective Synthesis of Switching Controllers for Linear Systems, *Proc. of the IEEE*, July, 2000.
5. E. Asarin, S. Bansal, B. Espiau, T. Dang and O. Maler, On Hybrid Control of Under-actuated Mechanical Systems, *HSCC*, 77-88 LNCS 2034, Springer, 2001.
6. E. Asarin, T. Dang and O. Maler, **d/dt**, a Tool for Reachability Analysis of Continuous and Hybrid Systems, *Proc. IFAC Nonlinear Control Systems*, 20-31, 2001.
7. C. Belta, J. Schug, T. Dang, V. Kumar, G.J. Pappas, H. Rubin and P. Dunlap, Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium *Vibrio Fisheri*, *Proc. 40th IEEE Conf. on Decision and Control*, 2001.
8. O. Botchkarev and S. Tripakis, Verification of Hybrid Systems with Linear Differential Inclusions Using Ellipsoidal Approximations, *HSCC*, 73-88 LNCS 1790, Springer, 2000.
9. O. Bournez, O. Maler and A. Pnueli, Orthogonal Polyhedra: Representation and Computation, *HSCC*, 46-60 LNCS 1569, Springer, 1999.
10. A. Chutinan and B.H. Krogh, Verification of Polyhedral Invariant Hybrid Automata Using Polygonal Flow Pipe Approximations, *HSCC*, 76-90 LNCS 1569, Springer, 1999.
11. T. Dang and O. Maler, Reachability Analysis via Face Lifting, *HSCC*, 96-109 LNCS 1386, Springer, 1998.
12. T.A. Henzinger, P.W. Kopke, A. Puri and P. Varaiya, What's decidable about hybrid automata?, *J. of Computer and System Sciences* 57, 94-124, 1998.
13. T.A. Henzinger, P.-H. Ho and H. Wong-Toi, HyTech: A Model Checker for Hybrid Systems, *Software Tools for Technology Transfer* 1, 110-122, 1997.
14. A. Kurzhanski and P. Varaiya, Ellipsoidal Techniques for Reachability Analysis, *HSCC*, 202-214 LNCS 1790, Springer, 2000.
15. K. Larsen, P. Pettersson and W. Yi, Uppaal in a nutshell, *Software Tools for Technology Transfert* 1-1, 1997.
16. O. Maler, Z. Manna and A. Pnueli, From Timed to Hybrid Systems, *Real-Time: Theory in Practice*, 447-484 LNCS 600, Springer, 1992.
17. G. Pappas, G. Lafferriere and S. Yovine, A New Class of Decidable Hybrid Systems, *HSCC*, 7-12 LNCS 1569, Springer, 1999.
18. B.I. Silva, O. Stursberg, B.H. Krogh and S. Engell, An Assessment of the Current Status of Algorithmic Approaches to the Verification of Hybrid Systems, *Proc. IEEE Conf. on Decision and Control*, 2867-2874, 2001.
19. S. Yovine, Kronos: A Verification Tool for Real-time Systems, *Software Tools for Technology Transfer* 1-1, 123-133, 1997.