Techniques and Tools for Hybrid Systems Reachability Analysis*

Erika Ábrahám

RWTH Aachen University, Germany

Hybrid systems are systems with combined discrete and continuous behaviour, typical examples being physical systems controlled by discrete controllers. Such systems can be found in various fields such as aviation, control engineering, medicine, or the wide field of cyber-physical systems.

The increasing relevance of hybrid systems, especially systems which interact with humans, requires careful design and proper *safety verification* techniques. Whereas the verification of purely continuous or purely discrete systems are already well-established research areas, the combination of discrete and continuous behaviours brings additional challenges for formal methods.

Logical formalisations are used in theorem-proving-based tools like KeYmaera [12], Ariadne [4], or the Isabelle/HOL-based tool described in [10]. Other tools like dReach [11], iSAT-ODE [6] and HSolver [13] also use logical characterisations but in combination with interval arithmetic and SMT solving. The tool C2E2 [5] uses validated numerical simulation; Bernstein expansion is implemented in [15]. This variety is complemented by approximation methods like hybridization, linearisation and abstraction techniques to increase the applicability of hybrid systems verification.

In this tutorial we focus on *flowpipe-construction-based* techniques and their implementation. As the reachability problem for hybrid systems is in general undecidable, flowpipe-construction-based reachability analysis techniques usually compute *over-approximations* of the set of reachable states of hybrid systems: starting from some initial sets, their time trajectories (*flowpipes*) and successors along discrete transitions (*jump successors*) are over-approximated in an iterative manner. Some tools in this area are Cora [1], Flow* [3], HyCreate [9], HySon [2], SoapBox [8], and SpaceEx [7].

The development of such tools is effortful, as datatypes for the underlying state set representations need to be implemented first. Our free and open-source C++ library HyPro [14] provides implementations for the most prominent state set representations, with the aim to offer assistance for the rapid implementation of new algorithms by encapsulating all representation-related issues and allowing the developers to focus on higher-level algorithmic aspects.

In this tutorial we give an introduction to hybrid systems, and to flowpipe-construction-based algorithms for computing their reachable state sets. After discussing theoretical aspects, we shortly describe available tools and introduce

* This work was supported by the German Research Council (DFG) in the context of the HyPro project.

in more detail our HyPro library, explain its functionalities, and give some examples to demonstrate its usage.

References

1. Althoff, M., Dolan, J.M.: Online verification of automated road vehicles using reachability analysis. IEEE Transaction on Robotics 30(4), 903–918 (2014)
2. Bouissou, O., Chapoutot, A., Mimram, S.: Computing flowpipe of nonlinear hybrid systems with numerical methods. CoRR abs/1306.2305 (2013), http://arxiv.org/abs/1306.2305
3. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: An analyzer for non-linear hybrid systems. In: Proc. of CAV'13. LNCS, vol. 8044, pp. 258–263. Springer (2013)
4. Collins, P., Bresolin, D., Geretti, L., Villa, T.: Computing the evolution of hybrid systems using rigorous function calculus. In: Proc. of ADHS'12. pp. 284–290. IFAC-PapersOnLine (2012)
5. Duggirala, P., Mitra, S., Viswanathan, M., Potok, M.: C2E2: A verification tool for Stateflow models. In: Proc. of TACAS'15, LNCS, vol. 9035, pp. 68–82. Springer (2015)
6. Eggers, A.: Direct Handling of Ordinary Differential Equations in Constraint-solving-based Analysis of Hybrid Systems. Ph.D. thesis, Universität Oldenburg, Germany (2014)
7. Frehse, G., Guernic, C.L., Donzé, A., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: SpaceEx: Scalable verification of hybrid systems. In: Proc. of CAV'11. LNCS, vol. 6806, pp. 379–395. Springer (2011)
8. Hagemann, W., Möhlmann, E., Rakow, A.: Verifying a PI controller using SoapBox and Stabhyli: Experiences on establishing properties for a steering controller. In: Proc. of ARCH'14. EPiC Series in Computer Science, vol. 34. EasyChair (2014)
9. HyCreate: A tool for overapproximating reachability of hybrid automata, http://stanleybak.com/projects/hycreate/hycreate.html
10. Immler, F.: Tool presentation: Isabelle/hol for reachability analysis of continuous systems. In: Proc. of ARCH14-15. EPiC Series in Computer Science, vol. 34, pp. 180–187. EasyChair (2015)
11. Kong, S., Gao, S., Chen, W., Clarke, E.M.: dReach: δ-reachability analysis for hybrid systems. In: Proc. of TACAS'15. LNCS, vol. 9035, pp. 200–205. Springer (2015)
12. Platzer, A., Quesel, J.: KeYmaera: A hybrid theorem prover for hybrid systems (system description). In: Proc. of IJCAR'08. LNCS, vol. 5195, pp. 171–178. Springer (2008)
13. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Proc. of HSCC'05. LNCS, vol. 3414, pp. 573–589. Springer (2005)
14. Schupp, S., Ábrahám, E., Ben Makhlouf, I., Kowalewski, S.: HyPro: A C++ library for state set representations for hybrid systems reachability analysis. In: Proc. of NFM'17. LNCS, vol. 10227, pp. 288–294. Springer International Publishing (2017)
15. Testylier, R., Dang, T.: NLTOOLBOX: A library for reachability computation of nonlinear dynamical systems. In: Proc. of the 11th Int. Symposium on Automated Technology for Verification and Analysis (ATVA'13). LNCS, vol. 8172, pp. 469–473. Springer (2013)