A Benchmark Suite for Hybrid Systems Reachability Analysis

Xin Chen[1], Stefan Schupp[1(✉)], Ibtissem Ben Makhlouf[1], Erika Ábrahám[1], Goran Frehse[2], and Stefan Kowalewski[1]

[1] RWTH Aachen University, Aachen, Germany
[2] Verimag, Gières, France
stefan.schupp@cs.rwth-aachen.de

Abstract. Since about two decades, formal methods for continuous and hybrid systems enjoy increasing interest in the research community. A wide range of analysis techniques were developed and implemented in powerful tools. However, the lack of appropriate benchmarks make the testing, evaluation and comparison of those tools difficult. To support these processes and to ease exchange and repeatability, we present a manifold benchmark suite for the reachability analysis of hybrid systems. Detailed model descriptions, classification schemes, and experimental evaluations help to find the right models for a given purpose.

1 Introduction

Recent advances in algorithms have turned reachability analysis into a powerful method for continuous and hybrid systems. Techniques are available that can compute approximations of the reachable states for systems with linear dynamics and more than 200 variables [11,16], and for complex non-linear dynamics [5,6,12]. Without any claim for completeness, some prominent tools based on different techniques are SpaceEx [11], Flow* [6], dReach [12], KeYmaera [17], iSAT [10], HSolver [18], HyCreate [14], Ariadne [7] and Cora [1]. Since in general the reachability problem is undecidable for hybrid systems, and even the one-step successors can only be computed approximately, experimental results are essential for validating algorithms, detecting their shortcomings, and identifying where further research is necessary.

Experiments in reachability require not only algorithms, but also models of systems and specifications that are to be verified. Such benchmarks are not easy to come by, in particular when looking for high-dimensional systems. Research papers typically include a small number of proprietary benchmarks, or modified versions of benchmarks published in other papers. A notable exception is a small collection of benchmarks in [8], and the benchmark collection of the ARCH

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

K. Havelund et al. (Eds.): NFM 2015, LNCS 9058, pp. 408–414, 2015.
DOI: 10.1007/978-3-319-17524-9_29

workshop series, which is tailored to industrial applications [2]. Using just a small number of benchmarks for test and evaluation comprehends the risk to tune tools to be efficient for certain application types only.

In this paper, we present a manifold collection of benchmarks for evaluating tools and algorithms for hybrid systems reachability, and to the best of our knowledge it is the first of this kind. It consists of system models along with property specifications, includes detailed descriptions, references to prior work, input files and exemplary results for some tools. Apart from making the benchmarks readily available in a unified form, the benchmark collection intends to make the following contributions:

Classification: The benchmarks originate from a variety of domains and serve a variety of purposes, e.g., testing scalability with respect to the number of variables or locations. Identifying a benchmark that suits a particular tool and helps to evaluate a certain property is non-trivial. The collection is organized by the model type (continuous/hybrid, linear/non-linear), which roughly corresponds to the kind of tool to which it is applicable. Within each class, benchmarks are listed by complexity (scalability, number of variables, locations, transitions). We intend to identify further attributes that help to find benchmarks with certain requested properties.

Specification: To ensure comparability of results between different tools, the specification needs to be unambiguous and formal. We provide such formal model specifications for all included benchmarks. Note that not all benchmarks easily lend themselves to specifications in the typical form of a given set of "bad states". For example, some benchmarks for testing the accuracy of approximations give quantitative results. Finding a unified form for specifying systems as well as their specifications is one of the long-term goals of the collection.

Evaluation criteria: Measuring the efficiency of algorithms can be done by measuring the running time and memory requirements of tools implementing them. Though comparing such measurements for different technologies is not objective, because the results are machine- and implementation-dependent, considering a larger experimental setting with a wider range of benchmarks allows to implicitly incorporate also other aspects such as accuracy, scalability and convergence rate (which in general influence running time and memory consumption).

Identifying challenges: Though state-of-the-art hybrid systems reachability analysis tools are impressively successful and can solve a wide range of interesting problems, they are still rarely applied outside their own community. Driving research directions towards the needs of other scientific areas and application domains would push this process forwards. Therefore, one of our long-term goals is to identify benchmarks suitable for this purpose, even if current tools do not exhibit sufficient functionalities yet.

Clearly, some of the above points will need to evolve while our benchmark suite grows and feedback from experiments becomes available.

The remainder of the paper is organized as follows: In the next section, we provide a brief overview of the benchmark collection. In Section 3, we show and discuss results for three tools on the benchmark suite, and conclude the paper in Section 4. The complete benchmark collection is available at [4].

2 The Benchmark Suite

Our benchmark suite currently covers 28 benchmarks. The included benchmarks are selected to cover different levels of expressivity in their components.

- We provide both pure *continuous* benchmarks as well as *hybrid* models.
- The continuous dynamics is described by either *linear* or *non-linear* ordinary differential equations.
- A further classification is provided according to the *number of variables* and, for hybrid behavior, the *number of locations* and the *number of discrete transitions*. One of the benchmarks is *scalable*, allowing the generation of high-dimensional models.
- The hybrid models specify *transition guards* varying in their form from half-spaces or hyperplanes over linear conditions up to non-linear ones.
- *Reset conditions* can be absent or described by linear terms.
- *Invariants* are boxes in some benchmarks and polyhedra in others.
- Reachability analysis is hindered by *Zeno behavior*, which is present in some of the models.

Our collection of linear benchmarks includes well-known smaller models such as the *bouncing ball* or the *two tank system*, as well as less known benchmarks, such as the *vehicle platoon* [3]. For the sake of completeness and for testing purposes we have decided to include also small but frequently referenced benchmarks. For the future it would be nice to have an even larger collection of small benchmarks that set traps for the reachability analysis through various model properties such as instability, Zeno behavior or deadlocks.

The non-linear models in our collection include benchmarks from different research fields such as mechanics, biology or electrical engineering. We have managed to extract benchmarks such as the *non-holonomic integrator* [13], the *spiking neurons* [15], *glycemic control* [9], or the *non-linear transmission line circuits* [19] from external sources, thus enhancing the collection by relevant, non-artificial benchmarks which are of interest in the previously mentioned fields and are now open to the formal methods community. Such non-artificial models are important for driving tool development towards being capable to solve real-world problems of different types.

The web page presentation [4] lists all benchmarks along with their property specifications, classified into linear continuous, non-linear continuous, linear hybrid, and non-linear hybrid models. For each model we list also measures regarding their size. We explain each of the benchmarks in our collection on its own web page, reference originating literature, provide a model description for downloading in SpaceEx and/or Flow* input format, and show example plottings of the reachable state set generated by those tools. In the future we plan to provide such information also for other tools.

Table 1. Linear hybrid benchmark results. Legends: **var**: #variables, **unsafe**: unsafe conditions, ***t***: running time in secs, $\boldsymbol{\delta}$: time-step size, ***k***: Taylor model order, **T.O.**: > 900 secs, **fail**: fail to prove the safety with $\delta \geq$ 1e-14.

benchmarks	**var**	**unsafe**	SpaceEx					Flow*		
			STC		**LGG**					
			t (s)		t (s)		δ	t (s)	δ	k
			box	oct	box	oct				
bouncing ball	2	$v \geq 10.7$	0.13	0.15	**0.02**	0.05	0.1	0.3	0.1	5
two tank system	2	$x_2 \leq -0.76$	*fail*	0.15	*fail*	**0.05**	0.01	5.3	0.01	10
rod reactor	3	location: shutdown	0.63	2.16	**0.09**	0.26	0.1	2.3	0.1	5
cruise control	3	$v \leq -2$	*fail*	*fail*	**0.08**	0.24	0.1	3.5	0.1	5
5-D lin. switch	5	$x_1 \leq -1.2$	0.06	0.79	**0.01**	9.23	0.01	1.8	0.01	15
3 vehicle platoon	9	$e_1 \geq 1.7$	0.19	9.4	**0.11**	10.28	0.1	20.3	0.02	10
filt. oscillator 4	6	$y \geq 0.5$	0.11	1.22	**0.09**	1.1	0.1	1.4	0.05	8
filt. oscillator 8	10	$y \geq 0.5$	0.32	13.9	**0.15**	10.5	0.1	4.1	0.05	8
filt. oscillator 16	18	$y \geq 0.5$	1.1	280	**0.2**	201	0.1	13.7	0.05	8
filt. oscillator 32	34	$y \geq 0.5$	**3.28**	*T.O.*	3.8	*T.O.*	0.1	70	0.05	8
5 vehicle platoon	16	$e_1 \in [-0.5, -0.2]$	**0.1**	1.54	*fail*	*fail*	1e-14	3.6	0.5	16
10 vehicle platoon	31	$e_1 \in [-0.5, -0.2]$	**0.29**	6.38	*fail*	*fail*	1e-14	44	0.5	20

3 Experimental Results

In this section we demonstrate the advantages of our benchmark suite by using hybrid models for the comparison of the tools SpaceEx, Flow* and dReach. Since different tools are devoted to different problem types, we distinguish between linear and non-linear hybrid benchmarks. All experiments were run on a Intel Core I-7 quad-core CPU with 4.0 GHz and 16 GB memory.

3.1 Linear Hybrid Benchmarks

SpaceEx [11] and Flow* [6] are two established tools for the reachability analysis of hybrid systems. SpaceEx is well-suited to analyze linear hybrid systems, whereas Flow* is specialized in non-linear systems but with a recent enhancement for dealing with linear systems.

SpaceEx has two scenarios. One of them is based on the LGG algorithm using support functions. The second one is the STC scenario, a recent enhancement of the LGG algorithm that produces fewer convex sets for a given accuracy and computes more precise images of discrete transitions. In SpaceEx, flowpipes are over-approximated by boxes or octagons, both of which are computed based on the same support functions. On the other hand, Flow* only uses Taylor Models for over-approximations.

In Table 3, we specify for each benchmark an unsafe condition, and use both of the tools to prove the safety of the system. For SpaceEx, we consider both of box and octagon, because the overall accuracy of octagons are better than that of boxes in general. From the table, it can be seen that the performance of the tools gradually becomes worse when the benchmark scale grows. On the

Table 2. Non-linear hybrid benchmark results. Legends: **var**: #variables, **unsafe**: unsafe conditions, $\boldsymbol{\delta}$: time-step size, $\boldsymbol{k}$ **in** Flow*: Taylor-model order, $\boldsymbol{t}$: running time in secs, $\boldsymbol{N}$: #subdivisions on the initial set, $\boldsymbol{k}$ **in** dReach: unrolling depth of bounded model checking, $\boldsymbol{p}$: value of numerical perturbation, **T.O.**: > 3600 secs.

			Flow*			dReach			
benchmark	var	unsafe	δ	k	t (s)	N	k	p	t (s)
non-holonomic integrator	3	$x \geq 3$	0.01	$5 \sim 8$	**201**	$1 \sim 10$	≤ 1	0.001	*T.O.*
spiking neuron I	2	$u \leq -25$	0.02	$4 \sim 6$	**367**	≥ 100	≤ 15	0.0001	*fail*
spiking neuron II	2	$u \geq 250$	0.02	$4 \sim 6$	**70**	$1 \sim 10$	≤ 15	0.001	*T.O.*
glycemic control I	3	$G \leq -2$	0.05	$2 \sim 5$	64	5	≤ 2	0.01	**1.1**
glycemic control II	3	$G \leq -2$	0.05	$2 \sim 5$	**95**	$1 \sim 5$	≤ 2	0.01	*T.O.*
glycemic control III	3	$G \leq -2$	0.05	$2 \sim 5$	**46**	$1 \sim 5$	≤ 1	0.01	*T.O.*
line circuit n = 2	2	$v_1 \geq 0.21$	0.01	$3 \sim 6$	2.3	1	≤ 2	0.01	**0.2**
line circuit n = 4	4	$v_1 \geq 0.21$	0.01	$3 \sim 6$	48	4	≤ 2	0.01	**9.6**
line circuit n = 6	6	$v_1 \geq 0.21$	$0.0002 \sim 0.02$	4	**243**	4	≤ 2	0.01	*T.O.*

other hand, some of the safety properties can not be proved (with "fail" in the table) due to the inaccuracy. Hence, the linear benchmarks from our collection are well-suited to evaluate tools in the aspects of accuracy and scalability.

3.2 Non-linear Hybrid Benchmarks

Since SpaceEx cannot work with non-linear models, we evaluate the performance of Flow* [6] and dReach [12] on the non-linear models in our benchmark suite. The main motivation to choose these tools is that Flow* is a typical safety verification tool based on flowpipe computation, while dReach is based on bounded model checking using constraint solving techniques. Thus we expect them to perform differently on different benchmarks.

We selected 12 non-linear benchmark instances from our benchmark suite for this experiment. The experimental results are listed in Table 2. The purpose of each experiment is to prove the safety. Since dReach cannot integrate large initial sets, for each benchmark, we divide the initial set into N parts in each dimension. Then for n variables, there are N^n subdivisions. Unlike the linear cases, the dynamics defined by a non-linear ODE can be very hard to handle. It can be seen that Flow* outperforms dReach on hard dynamics, while dReach works better when the dynamics is moderate. Therefore, our collection of non-linear benchmarks may provide a reasonable evaluation of a tool in not only scalability but also the ability to deal with hard dynamics.

4 Conclusion

The presented benchmark suite is an important first step to support the testing, evaluation and comparison of hybrid systems reachability analysis tools. Next steps will cover the extension with further benchmarks, including models with more expressive power like, e.g., continuous dynamics involving transcendental functions, urgent locations and transitions, or non-convex location invariants and transition guards. We will also investigate further classification criteria, with special interest in providing measures for the hardness of the problems. These steps are not only helpful for finding appropriate benchmarks and for evaluating tools, but also for the identification of interesting future research directions towards challenging unsolved problems.

References

1. Althoff, M., Dolan, J.M.: Online verification of automated road vehicles using reachability analysis. IEEE Trans. on Robotics **30**(4), 903–918 (2014)
2. Althoff, M., Frehse, G.: Benchmarks of the Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH) (2014). http://cps-vo.org/group/ARCH/benchmarks
3. Ben Makhlouf, I., Diab, H., Kowalewski, S.: Safety verification of a controlled cooperative platoon under loss of communication using zonotopes. In: Proc. of ADHS 2012, pp. 333–338. IFAC-PapersOnLine (2012)
4. Benchmarks of continuous and hybrid systems. http://ths.rwth-aachen.de/research/hypro/benchmarks-of-continuous-and-hybrid-systems/
5. Bouissou, O., Chapoutot, A., Djoudi, A.: Enclosing temporal evolution of dynamical systems using numerical methods. In: Brat, G., Rungta, N., Venet, A. (eds.) NFM 2013. LNCS, vol. 7871, pp. 108–123. Springer, Heidelberg (2013)
6. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: an analyzer for non-linear hybrid systems. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 258–263. Springer, Heidelberg (2013)
7. Collins, P., Bresolin, D., Geretti, L., Villa, T.: Computing the evolution of hybrid systems using rigorous function calculus. In: Proc. of ADHS 2012, pp. 284–290. IFAC-PapersOnLine (2012)
8. Fehnker, A., Ivančić, F.: Benchmarks for hybrid systems verification. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 326–341. Springer, Heidelberg (2004)
9. Fisher, M.: A semiclosed-loop algorithm for the control of blood glucose levels in diabetics. IEEE Trans. on Biomedical Engineering **38**(1), 57–61 (1991)
10. Fränzle, M., Herde, C., Ratschan, S., Schubert, T., Teige, T.: Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. Journal on Satisfiability, Boolean Modeling and Computation **1**, 209–236 (2007)
11. Frehse, G., et al.: SpaceEx: scalable verification of hybrid systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 379–395. Springer, Heidelberg (2011)
12. Gao, S.: Computable Analysis, Decision Procedures, and Hybrid Automata: A New Framework for the Formal Verification of Cyber-Physical Systems. Ph.D. thesis, Carnegie Mellon University (2012)
13. Hespanha, J., Morse, A.: Stabilization of nonholonomic integrators via logic-based switching. Automatica **35**(3), 385–393 (1999)
14. HyCreate: A tool for overapproximating reachability of hybrid automata. http://stanleybak.com/projects/hycreate/hycreate.html
15. Izhikevich, E.: Dynamical Systems in Neuroscience. MIT Press (2007)
16. Le Guernic, C., Girard, A.: Reachability analysis of linear systems using support functions. Nonlinear Analysis: Hybrid Systems **4**(2), 250–262 (2010)

17. Platzer, A., Quesel, J.-D.: KeYmaera: a hybrid theorem prover for hybrid systems (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS, vol. 5195, pp. 171–178. Springer, Heidelberg (2008)
18. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 573–589. Springer, Heidelberg (2005)
19. Rewienski, M., White, J.: A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems **22**(2), 155–170 (2003)