ARCH-COMP17 Category Report: Continuous Systems with Nonlinear Dynamics

Conference Paper · April 2017

DOI: 10.29007/v6g4

3 authors, including:

Matthias Althoff
Technische Universität München
169 PUBLICATIONS **2,810** CITATIONS

SEE PROFILE

Fabian Immler
Carnegie Mellon University
19 PUBLICATIONS **200** CITATIONS

SEE PROFILE

EPiC Series in Computing

Volume 48, 2017, Pages 160–169

ARCH17. 4th International Workshop on Applied
Verification of Continuous and Hybrid Systems

EPiC Computing

ARCH-COMP17 Category Report: Continuous Systems with Nonlinear Dynamics

Xin Chen[1], Matthias Althoff[2], and Fabian Immler[2]

[1] University of Colorado, Boulder, CO, United States
xinchen@colorado.edu

[2] Technische Universität München, Department of Informatics, Munich, Germany
{althoff,immler}@in.tum.de

Abstract

We present the results of a friendly competition for formal verification of continuous and hybrid systems with nonlinear continuous dynamics. The friendly competition took place as part of the workshop Applied Verification for Continuous and Hybrid Systems (ARCH) in 2017. This year, three tools CORA, Flow* and Isabelle/HOL (in alphabetic order) participated. They are applied to solve the reachability analysis problems on three benchmarks which have 2, 7 and 12 variables respectively. We do not rank the tools based on the results, but show the current status and discover the potential advantages of different tools. Besides, the computational settings presented here provide a guide to use the tools although they might not be optimal.

1 Introduction

Disclaimer The presented report of the ARCH friendly competition for *continuous and hybrid systems with linear continuous dynamics* aims at providing a landscape of the current capabilities of verification tools. We would like to stress that each tool has unique strengths—not all of the specificities can be highlighted within a single report. To reach a consensus in what benchmarks are used, some compromises had to be made so that some tools may benefit more from the presented choice than others. The obtained results have been verified by an independent repeatability evaluation. To establish further trustworthiness of the results, the code with which the results have been obtained is publicly available.

In this report, we summarize the results of the first ARCH friendly competition on the reachability analysis of nonlinear continuous systems. Given a system defined by a nonlinear Ordinary Differential Equation (ODE) $\dot{\vec{x}} = f(\vec{x}, t)$ along with an initial condition $\vec{x} \in X_0$ as well as an unsafe set U, we apply the participating tools to prove that there is no state reachable

G. Frehse and M. Althoff (eds.), ARCH17 (EPiC Series in Computing, vol. 48), pp. 160–169

contained in U over a bounded time horizon. The techniques for solving such a problem are usually very sensitive to not only the nonlinearity of the dynamics but also the size of the initial set. This is also one of the main reasons why most of the tools require quite a lot of computational parameters.

In this report, three tools CORA, Flow*, and Isabelle/HOL participate in solving the safety problems defined on three nonlinear benchmarks which are the Van der Pol oscillator, the Laub-Loomis model, and a controlled quadrotor model. The benchmarks are selected based on the discussions of the tool authors. Since the experimental results are produced on different platforms, we provide Section A for the hardware details.

2 Participating Tools

CORA The tool *COntinuous Reachability Analyzer* (CORA) [3, 4] realizes techniques for reachability analysis with a special focus on developing scalable solutions for verifying hybrid systems with nonlinear continuous dynamics and/or nonlinear differential-algebraic equations. A further focus is on considering uncertain parameters and system inputs. Due to the modular design of CORA, much functionality can be used for other purposes that require resource-efficient representations of multi-dimensional sets and operations on them. CORA is implemented as an object-oriented MATLAB code. The modular design of CORA makes it possible to use the capabilities of the various set representations for other purposes besides reachability analysis. CORA is available at `http://www6.in.tum.de/Main/SoftwareCORA`.

Flow*. The tool Flow* [10] uses an adapted Taylor Model (TM) integration method to compute reachable set overapproximations for nonlinear continuous and hybrid systems. Similar to the original method proposed in [7], an ODE solution, i.e., a function over the initial set as well as the time variable, over a bounded time interval is overapproximated by a TM in Flow*, and it therefore forms an overapproximation of the reachable set there. We also call this TM a TM flowpipe. For the discrete jumps of hybrid systems, Flow* uses the techniques of domain contraction and range overapproximation to compute flowpipe/guard intersections [9], and then aggregates them by a box or parallelotope. Besides, in order to reduce the accumulation of overestimation during an integration job, the tool can symbolically represent the remainders of the previous N flowpipes for some $N > 0$ (see [11]). Flow* is available at `flowstar.org`.

Isabelle/HOL-ODE-Numerics. HOL-ODE-Numerics [12, 13] is a collection of rigorous numerical algorithms for continuous systems. It is based on Runge-Kutta methods implemented with affine arithmetic. The distinctive feature is that all algorithms are formally verified in the interactive theorem prover Isabelle/HOL: everything from single roundoff errors to the global approximation scheme is proved correct with respect to a formalization of ODEs in Isabelle/HOL. The resulting code is therefore highly trustworthy. It does, however, not feature many optimizations or the most sophisticated algorithms. We therefore do not expect competitive performance figures. Nevertheless, the tool should exhibit reasonable performance: it should scale (modulo possibly large constant factors) like “regular” tools implementing similar algorithms.

3 Benchmarks

3.1 Van der Pol Oscillator

3.1.1 Model

The Van der Pol oscillator was introduced by the Dutch physicist Balthasar van der Pol. It can be defined by the following ODE with 2 variables.

$$\begin{cases} \dot{x} &= y \\ \dot{y} &= y - x - x^2 y \end{cases}$$

The system has a stable limit cycle however shows complicated behavior.

3.1.2 Specification

We consider the initial condition $x(0) \in [1.25, 1.55]$, $y(0) \in [2.35, 2.45]$ which is used in [1]. The unsafe set is given by $y \geq 2.75$ for the time horizon $[0, 7]$.

3.1.3 Results

The time costs of the participating tools on the Van der Pol oscillator benchmark are given in Table 1, and the plots of the overapproximation sets are presented in Figure 1. We also provide the computational settings of the tools as below.

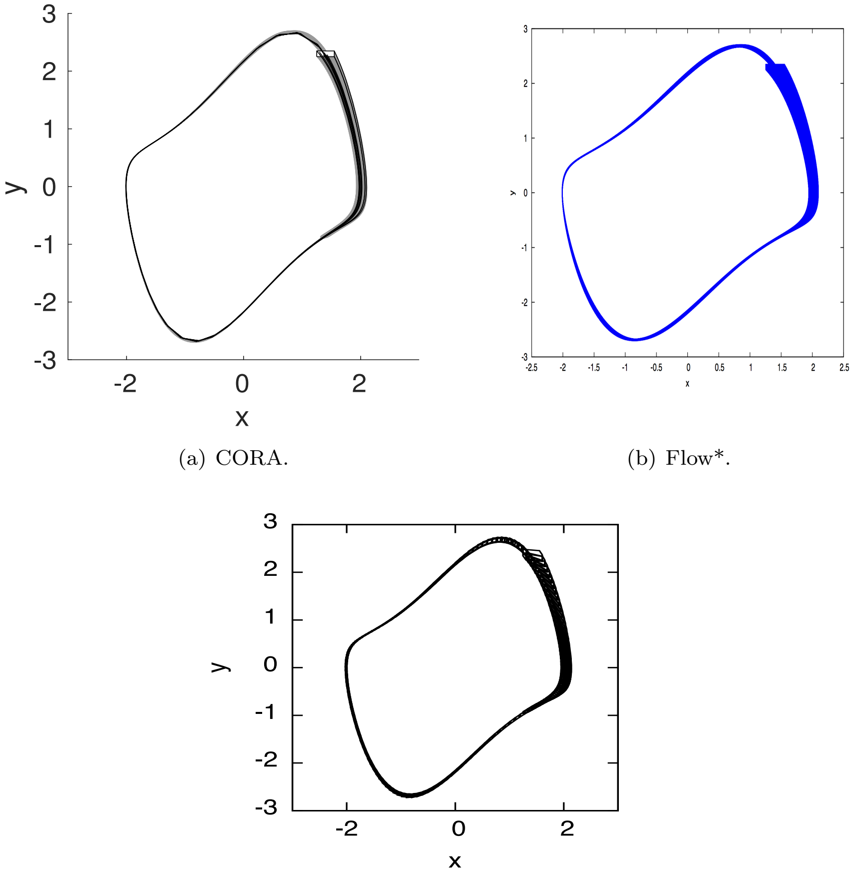
Setting for CORA. CORA has introduced a pseudo invariant at $x = 1.5$. Further, CORA uses the time step size 0.01 and the zonotope order is chosen as 20.

Setting for Flow*. Flow* uses the step size 0.02, the TM order 5, the cutoff threshold 10^{-6}, and the precision 53 for floating-point numbers. The TM flowpipe remainders are kept symbolically every 100 steps. All floating-point roundoff errors are included in the overapproximations. Since there are only 2 state variables, the tool plots a grid overapproximation for the flowpipes, see Figure 1(b). The approximation quality can be better evaluated based on the remainder size of the last TM flowpipe (see [8]). In this task, the maximum width of that remainder is below 0.02434.

Setting for Isabelle/HOL. Maximum Zonotope order is set to 20, Reachability analysis is carried out with an (absolute and relative) error tolerance of 2^{-12}. A pseudo-invariant is added at $x = 1.5$.

Table 1: Results of the Van der Pol Oscillator. Details of the platforms are described in Section A.

tool	computation time in [s]	language	machine
CORA	8	MATLAB	$\mathrm{M}_{\mathrm{CORA}}$
Flow*	2	C++	$\mathrm{M}_{\mathrm{Flow*}}$
Isabelle/HOL	3	SML	$\mathrm{M}_{\mathrm{Isabelle}}$



(a) CORA.

(b) Flow*.

(c) Isabelle/HOL.

Figure 1: Reachable set overapproximations for the Van der Pol oscillator.

3.2 Laub-Loomis Benchmark

3.2.1 Model

The Laub-Loomis model is presented in [14] for studying a class of enzymatic activities. The dynamics can be defined by the following ODE with 7 variables.

$$\left\{\begin{array}{lcl} \dot{x}_1 & = & 1.4x_3 - 0.9x_1 \\ \dot{x}_2 & = & 2.5x_5 - 1.5x_2 \\ \dot{x}_3 & = & 0.6x_7 - 0.8x_2x_3 \\ \dot{x}_4 & = & 2 - 1.3x_3x_4 \\ \dot{x}_5 & = & 0.7x_1 - x_4x_5 \\ \dot{x}_6 & = & 0.3x_1 - 3.1x_6 \\ \dot{x}_7 & = & 1.8x_6 - 1.5x_2x_7 \end{array}\right.$$

The system is asymptotically stable and the equilibrium is the origin.

3.2.2 Specification

The initial sets are defined according to the one used in [15]. They are boxes centered at $x_1(0) = 1.2$, $x_2(0) = 1.05$, $x_3(0) = 1.5$, $x_4(0) = 2.4$, $x_5(0) = 1$, $x_6(0) = 0.1$, $x_7(0) = 0.45$. The width of the initial set is vital to the difficulty of the reachability analysis job. The larger the initial set the harder the reachability analysis. In the paper, we consider the initial box of the widths $W = 0.01$ and $W = 0.1$. For the smaller initial box, we consider the unsafe region defined by $x_4 \geq 4.5$, while for the larger one, the unsafe set is defined by $x_4 \geq 5$. The time horizon for both of the cases is $[0, 20]$.

3.2.3 Results

The computation results of the tools are given in Table 2. Since the safety condition is only related to the variable x_4, we present the plots of projections of the overapproximations in the t-x_4 plane such that t is the time variable. It can be seen that enlarging the initial set size can greatly make the reachability analysis task harder. The tool settings are given as below.

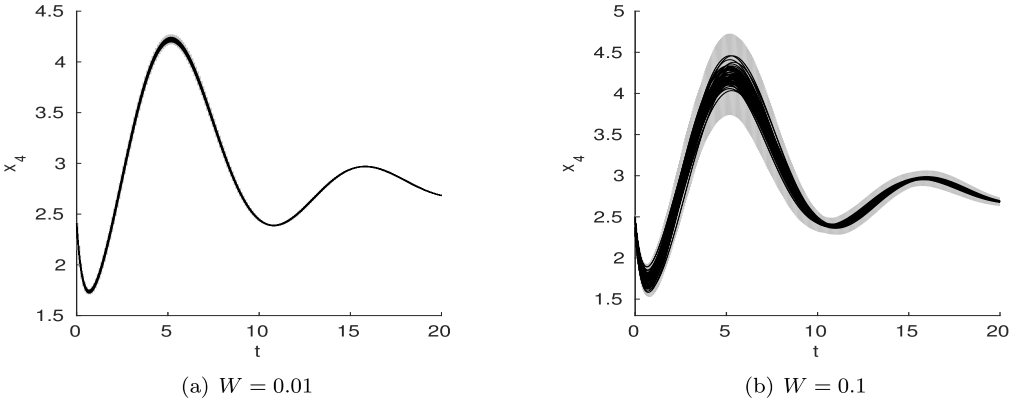
Setting for CORA. Depending on whether the smaller or larger initial sets are used, different algorithms in CORA are applied. For the smaller initial set, the faster but less accurate algorithm presented in [5] is executed. For the larger initial set, the more accurate but slower algorithm from [2] is used. CORA uses a step size of 0.1 for the small initial set and a step size of 0.05 for the large initial set. The maximum zonotope order for both initial sets is chosen as 50.

Setting for Flow*. For the small initial set, Flow* uses the step size 0.05, the TM order 4, the cutoff threshold 10^{-6} and the precision 100 for floating-point numbers. The TM flowpipe remainders are kept symbolically every 50 steps. On the other hand, for the large initial set, Flow* uses the same setting except that it keeps the remainders symbolically every 200 steps. All floating-point roundoff errors are included in the overapproximations. The plots of the flowpipes are shown in Figure 3. Notice that they are only the interval overapproximations of the flowpipes, the exact flowpipes are much more accurate, since for the small initial set, the maximum width of the last flowpipe remainder is only 0.02004 which is determined by the x_4-dimension, while for the large initial set, the maximum width is only 0.07634.

Setting for Isabelle/HOL. Maximum Zonotope order is set to 60 for the smaller initial set and 100 for the larger one. Reachability analysis is carried out with an (absolute and relative) error tolerance of 2^{-12} for the smaller initial set and 2^{-14} for the larger one.

Table 2: Results of the Laub-Loomis model. Details of the platforms are described in Section A.

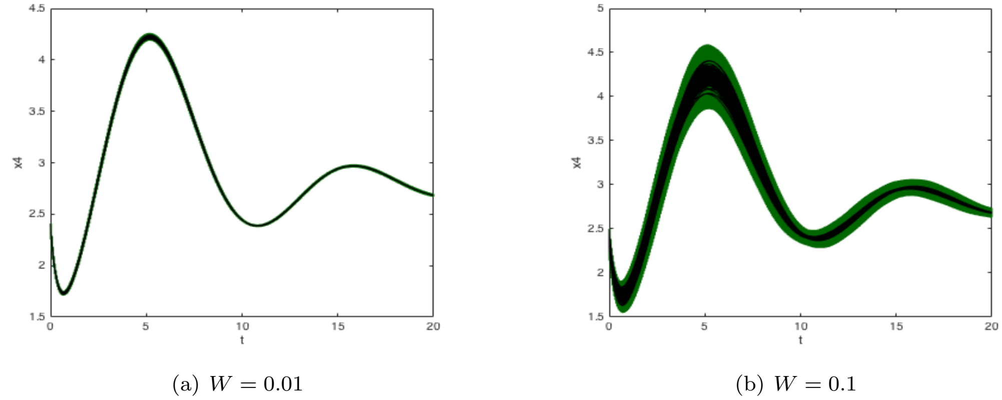
	computation time in [s]		**platform**	
tool	$W = 0.01$	$W = 0.1$	**language**	**machine**
CORA	3	63	MATLAB	$\mathrm{M}_{\mathrm{CORA}}$
Flow*	8	18	C++	$\mathrm{M}_{\mathrm{Flow*}}$
Isabelle/HOL	90	1200	SML	$\mathrm{M}_{\mathrm{Isabelle}}$



(a) $W = 0.01$

(b) $W = 0.1$

Figure 2: Reachable set overapproximations for the Laub-Loomis model computed by CORA. Numerical simulations are in black.



(a) $W = 0.01$

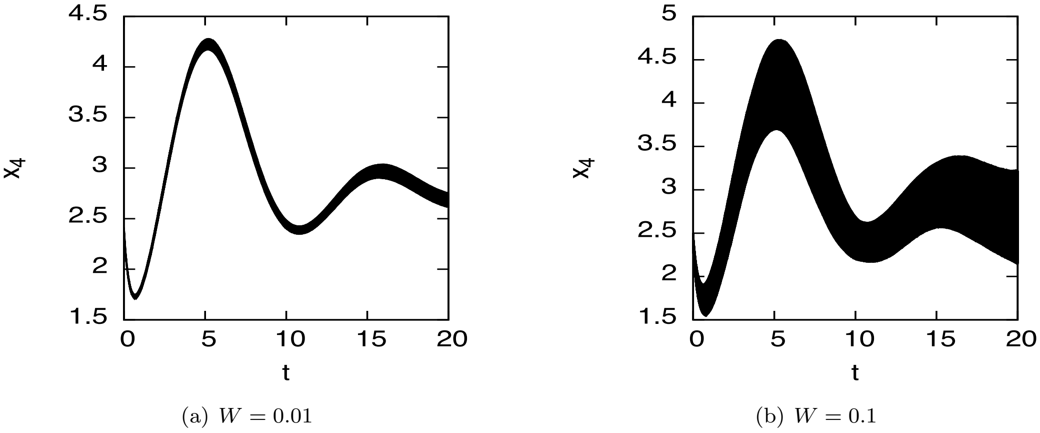
(b) $W = 0.1$

Figure 3: Reachable set overapproximations for the Laub-Loomis model computed by Flow*. Numerical simulations are in black.

3.3 Quadrotor Benchmark

3.3.1 Model

We study the dynamics of a quadrotor as derived in [6, eq. (16) - (19)]. Let us first introduce the variables required to describe the model: The inertial (north) position x_1, the inertial (east) position x_2, the altitude x_3, the longitudinal velocity x_4, the lateral velocity x_5, the vertical velocity x_6, the roll angle x_7, the pitch angle x_8, the yaw angle x_9, the roll rate x_{10}, the pitch rate x_{11}, and the yaw rate x_{12}. We further require the following parameters: gravity constant $g = 9.81$ [m/s^2], radius of center mass $R = 0.1$ [m], distance of motors to center mass $l = 0.5$ [m], motor mass $M_{rotor} = 0.1$ [kg], center mass $M = 1$ [kg], and total mass $m = M + 4M_{rotor}$.



(a) $W = 0.01$

(b) $W = 0.1$

Figure 4: Reachable set overapproximations for the Laub-Loomis model computed by Isabelle/HOL.

From the above parameters we can compute the moments of inertia as

$$\begin{aligned} J_x =& \frac{2}{5}\, M\, R^2 + 2\, l^2\, M_{rotor}, \\ J_y =& J_x, \\ J_z =& \frac{2}{5}\, M\, R^2 + 4\, l^2\, M_{rotor}. \end{aligned}$$

Finally, we can write the set of ordinary differential equations for the quadrotor according to [6, eq. (16) - (19)]:

$$\left\{ \begin{array}{lcl} \dot{x}_1 &=& \cos(x_8)\cos(x_9)x_4 + \Big(\sin(x_7)\sin(x_8)\cos(x_9) - \cos(x_7)\sin(x_9)\Big)x_5 \\ & & +\Big(\cos(x_7)\sin(x_8)\cos(x_9) + \sin(x_7)\sin(x_9)\Big)x_6 \\ \dot{x}_2 &=& \cos(x_8)\sin(x_9)x_4 + \Big(\sin(x_7)\sin(x_8)\sin(x_9) + \cos(x_7)\cos(x_9)\Big)x_5 \\ & & +\Big(\cos(x_7)\sin(x_8)\sin(x_9) - \sin(x_7)\cos(x_9)\Big)x_6 \\ \dot{x}_3 &=& \sin(x_8)x_4 - \sin(x_7)\cos(x_8)x_5 - \cos(x_7)\cos(x_8)x_6 \\ \dot{x}_4 &=& x_{12}x_5 - x_{11}x_6 - g\sin(x_8) \\ \dot{x}_5 &=& x_{10}x_6 - x_{12}x_4 + g\cos(x_8)\sin(x_7) \\ \dot{x}_6 &=& x_{11}x_4 - x_{10}x_5 + g\cos(x_8)\cos(x_7) - \frac{F}{m} \\ \dot{x}_7 &=& x_{10} + \sin(x_7)\tan(x_8)x_{11} + \cos(x_7)\tan(x_8)x_{12} \\ \dot{x}_8 &=& \cos(x_7)x_{11} - \sin(x_7)x_{12} \\ \dot{x}_9 &=& \frac{\sin(x_7)}{\cos(x_8)}x_{11} + \frac{\cos(x_7)}{\cos(x_8)}x_{12} \\ \dot{x}_{10} &=& \frac{J_y - J_z}{J_x}x_{11}x_{12} + \frac{1}{J_x}\tau_\phi \\ \dot{x}_{11} &=& \frac{J_z - J_x}{J_y}x_{10}x_{12} + \frac{1}{J_y}\tau_\theta \\ \dot{x}_{12} &=& \frac{J_x - J_y}{J_z}x_{10}x_{11} + \frac{1}{J_z}\tau_\psi \end{array} \right.$$

To check interesting control specifications, we stabilize the quadrotor using simple PD controllers for height, roll, and pitch. The inputs to the controller are the desired values for height, roll, and pitch u_1, u_2, and u_3, respectively. The equations of the controllers are

$$\begin{aligned} F &= m\,g - 10(x_3 - u_1) + 3x_6 && \text{(height control)},\\ \tau_\phi &= -(x_7 - u_2) - x_{10} && \text{(roll control)},\\ \tau_\theta &= -(x_8 - u_3) - x_{11} && \text{(pitch control)}. \end{aligned}$$

We leave the heading uncontrolled so that we set $\tau_\psi = 0$.

3.3.2 Specification

The task is to change the height from 0 [m] to 1 [m] within 5 [s]. A goal region $[0.98, 1.02]$ of the height x_3 has to be reached within 5 [s] and the height has to stay below 1.4 for all times. After 1 [s] the height should stay above 0.9 [m]. The initial position of the quadrotor is uncertain in all directions within $[-0.4, 0.4]$ [m] and also the velocity is uncertain within $[-0.4, 0.4]$ [m/s] for all directions. All other values are initialized as 0.

3.3.3 Results

The results of the reachability computation for the quadrotor model are given in Figure 5 and Table 3. We give the settings for CORA and Flow* as below.

Setting for CORA. CORA uses the step size 0.1 and the zonotope order 50. The computation is carried out using the approach in [5], which conservatively linearizes the system dynamics for each consecutive time interval by adding the linearization error as an uncertain input. The linearization error is obtained using the Lagrange remainder, which are evaluated via interval arithmetic. This results in many function calls (especially for this example), whose overhead has been reduced since MATLAB R2015b. So the execution time for the quadrotor benchmark depends significantly on the MATLAB version (more than twice as fast since R2015b).

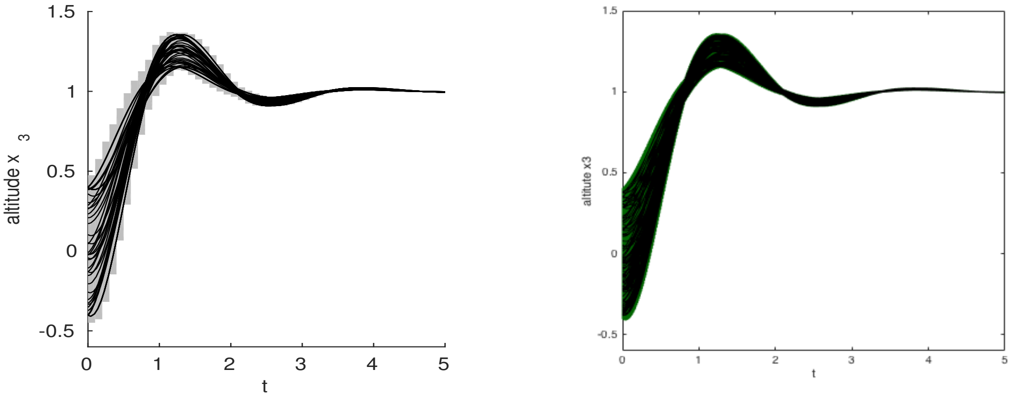
Setting for Flow*. Flow* uses the step size 0.01, the TM order 4, the cutoff threshold 10^{-6} and the precision 100 for floating-point numbers. The TM flowpipe remainders are kept symbolically every 20 steps. All floating-point roundoff errors are included in the overapproximation. Figure 5(b) illustrates the interval overapproixmations for the flowpipes. To better evaluate the approximation error, we provide the maximum remainder width of the last flowpipe and that is only 0.00128.

Table 3: Results of the quadrotor model. Details of the platforms are described in Section A.

tool	**computation time in** [s]	**language**	**machine**
CORA	11	MATLAB	$\mathrm{M}_{\mathrm{CORA}}$
Flow*	12	C++	$\mathrm{M}_{\mathrm{Flow*}}$
Isabelle/HOL	-	SML	$\mathrm{M}_{\mathrm{Isabelle}}$

4 Conclusion and Outlook

From the results of the competition on nonlinear systems, we can see that the techniques handling nonlinear dynamics often require more user-specified parameters than those for linear



(a) CORA.

(b) Flow*.

Figure 5: Reachable set overapproximations for the quadrotor model.

dynamics. all of the tools have their advantages on some examples, and the applicability of a tool to one example is usually quite sensitive to the computational setting in use. Therefore, it could be a promising direction for coming up with new techniques to find proper settings for a tool and optimize its performance for a given computation task.

In the next year, we hope that more tools could join the friendly competition and we will also collect more benchmarks which are not only continuous but also hybrid. We will also try to expose the advantages of different tools by examples. The reports of other categories can be found in the proceedings and on the ARCH website: cps-vo.org/group/ARCH.

5 Acknowledgments

The authors gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921.

A Specification of Used Machines

A.1 $\mathbf{M_{CORA}}$

- Processor: Intel Core i7-3520M CPU @ 2.90GHz x 4
- Memory: 7.6 GB
- Average CPU Mark on `www.cpubenchmark.net`: 4515 (full), 1785 (single thread)

A.2 $\mathbf{M_{Flow*}}$

Virtual machine on VMware Workstation 11 with a single core CPU and 4.0 GB memory. The operating systems is Ubuntu 16.04 LTS. The physical CPU is given as below.

- Processor: Intel Xeon E3-1245 V3 @ 3.4GHz x 4
- Average CPU Mark on `www.cpubenchmark.net`: 9545 (full), 2155 (single thread)

A.3 $\mathrm{M}_{\textbf{Isabelle}}$

- Processor: Intel Core i7-4960HQ CPU @ 2.60GHz x 4
- Memory: 16 GB 1600 MHz DDR3
- Average CPU Mark on www.cpubenchmark.net: 9770 (full), 2169 (single thread)

References

[1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars.* PhD thesis, Technischen Universität München, 2010.

[2] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Hybrid Systems: Computation and Control*, pages 173–182, 2013.

[3] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[4] M. Althoff and D. Grebenyuk. Implementation of interval arithmetic in CORA 2016. In *Proc. of the 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 91–105, 2016.

[5] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[6] Randal Beard. Quadrotor dynamics and control rev 0.1. Technical report, Brigham Young University, 2008.

[7] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4:361–369, 1998.

[8] X. Chen. *Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models.* PhD thesis, RWTH Aachen University, 2015.

[9] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *Proc. of RTSS'12*, pages 183–192. IEEE Computer Society, 2012.

[10] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proc. of CAV'13*, volume 8044 of *LNCS*, pages 258–263. Springer, 2013.

[11] X. Chen and S. Sankaranarayanan. Decomposed reachability analysis for nonlinear systems. In *Proc. of RTSS'16*, pages 13–24. IEEE Computer Society, 2016.

[12] F. Immler. Verified reachability analysis of continuous systems. In *Proc. of TACAS'15*, volume 9035 of *LNCS*, pages 37–51. Springer, 2015.

[13] F. Immler and J. Hlzl. Ordinary differential equations. *Archive of Formal Proofs*, April 2012. http://isa-afp.org/entries/Ordinary_Differential_Equations.shtml, Formal proof development.

[14] M. T. Laub and W. F. Loomis. A molecular network that produces spontaneous oscillations in excitable cells of dictyostelium. *Molecular Biology of the Cell*, 9:3521–3532, 1998.

[15] R. Testylier and T. Dang. Nltoolbox: A library for reachability computation of nonlinear dynamical systems. In *Proc. of ATVA'13*, volume 8172 of *LNCS*, pages 469–473. Springer, 2013.