EPiC Series in Computing

Volume 54, 2018, Pages 53–70

ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems


# ARCH-COMP18 Category Report: Continuous and Hybrid Systems with Nonlinear Dynamics

Fabian Immler[1], Matthias Althoff[1], Xin Chen[2], Chuchu Fan[3], Goran Frehse[4], Niklas Kochdumper[1], Yangge Li[3], Sayan Mitra[3], Mahendra Singh Tomar[1], and Majid Zamani[1]

[1] Technische Universität München, Munich, Germany
{althoff,immler}@in.tum.de,{niklas.kochdumper,mahendrasingh.tomar,zamani}@tum.de
[2] University of Dayton, Dayton, OH, United States
xchen4@udayton.edu
[3] University of Illinois at Urbana-Champaign, Champaign, IL, United States
{mitras,cfan10,li213}@illinois.edu
[4] Univ. Grenoble Alpes, Grenoble, France
goran.frehse@univ-grenoble-alpes.fr

**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 2018. In this year, six tools CORA, CORA/SX, C2E2, Flow*, Isabelle/HOL, and SymReach (in alphabetic order) participated. They are applied to solve reachability analysis problems on four benchmarks problems, one of them with hybrid dynamics. We do not rank the tools based on the results, but show the current status and discover the potential advantages of different tools.

## 1 Introduction

**Disclaimer** The presented report of the ARCH friendly competition for *continuous and hybrid systems with nonlinear 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 at gitlab.com/goranf/ARCH-COMP.

In this report, we summarize the results of the second ARCH friendly competition on the reachability analysis of continuous and hybrid systems with nonlinear dynamics. 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 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, six tools CORA, CORA/SX, C2E2, Flow*, Isabelle/HOL, and SymReach participate in solving the safety problems defined on three continuous and one hybrid benchmark. The continuous benchmarks are the Van der Pol oscillator, the Laub-Loomis model, and a controlled quadrotor model. The hybrid benchmark models a space rendezvous.

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`.

**CORA/SX** CORA/SX is a port of the basic zonotope reachability algorithm from the CORA Matlab toolbox to SpaceEx. There are some differences between CORA and CORA/SX. We believe they have only a minor effect on the results in this report, and we will summarize them briefly. CORA/SX varies slightly in that some matrix computations (which approximate the input over one time step) use SpaceEx code instead of an overapproximation that is based on intervals. CORA/SX uses its own proprietary symbolic differentiation to compute the Jacobian and Hessian matrices. Affine arithmetic based on the library AAFlib [18] is used to obtain interval bounds on the linearization error.

**C2E2.** C2E2 (Compare-Execute-Check-Engine) [15, 16] is a tool for verifying bounded-time invariant properties for hybrid system with both linear or nonlinear dynamics, and discrete transitions with guards and resets. The tool implements a *simulation-based approach* for overapproximating the reachable states. The input hybrid automata and the unsafe set has to be represented in an XML format. The new version of C2E2 used for these experiments (to be released in Fall 2018) comes with a model editor that can compose hybrid automata and a built-in plotter. C2E2 and related publications are available from `https://publish.illinois.edu/c2e2-tool/`.

**Flow*.** The tool Flow* [12] 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 [8], 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 [11], 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 [13]). In order to produce guaranteed results, the tool represents reals by their interval enclosures such that all roundoff errors are taken into account. In the future, we plan to improve the estimation of roundoff errors so as to make the tool more numerically stable. Flow* is available at `flowstar.org`.

**Isabelle/HOL-ODE-Numerics.** HOL-ODE-Numerics [19, 20] 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. Isabelle/HOL is available at `https://isabelle.in.tum.de`, HOL-ODE-Numerics is part of the Archive of Formal Proofs `http://isa-afp.org/entries/Ordinary_Differential_Equations.shtml`.

**SymReach.** SymReach is a tool (under development) for the computation of an overapproximation of the reachable set of continuous time nonlinear systems. It is a C++ implementation of the procedure [1, 5] that is also included in CORA, with the hope that computation time may come out to be smaller. It utilizes on-the-fly linearization using first order Taylor series and its Lagrange remainder. To compute the reachable set for each step, an interval called the applied error (AE) is assumed to enclose the linearization error. The computed error (CE) on the obtained tentative reachable set is then compared with the AE. The set is accepted if the CE is a subset of the AE, otherwise the initial set for the step is split into two and the process is repeated for each of the two newly created sets. SymReach is available at `https://github.com/mahendrasinghtomar/SymReach`.

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 | 2.3 | MATLAB | $M_{CORA}$ |
| CORA/SX | 0.6 | C++ | $M_{SpaceEx}$ |
| C2E2 | 38.5 | C++ | $M_{C2E2}$ |
| Flow* | 1.5 | C++ | $M_{Flow*}$ |
| Isabelle/HOL | 1.5 | SML | $M_{Isabelle}$ |
| SymReach | 17.14 | C++ | $M_{SymReach}$ |

# 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 $[0, 7]$.

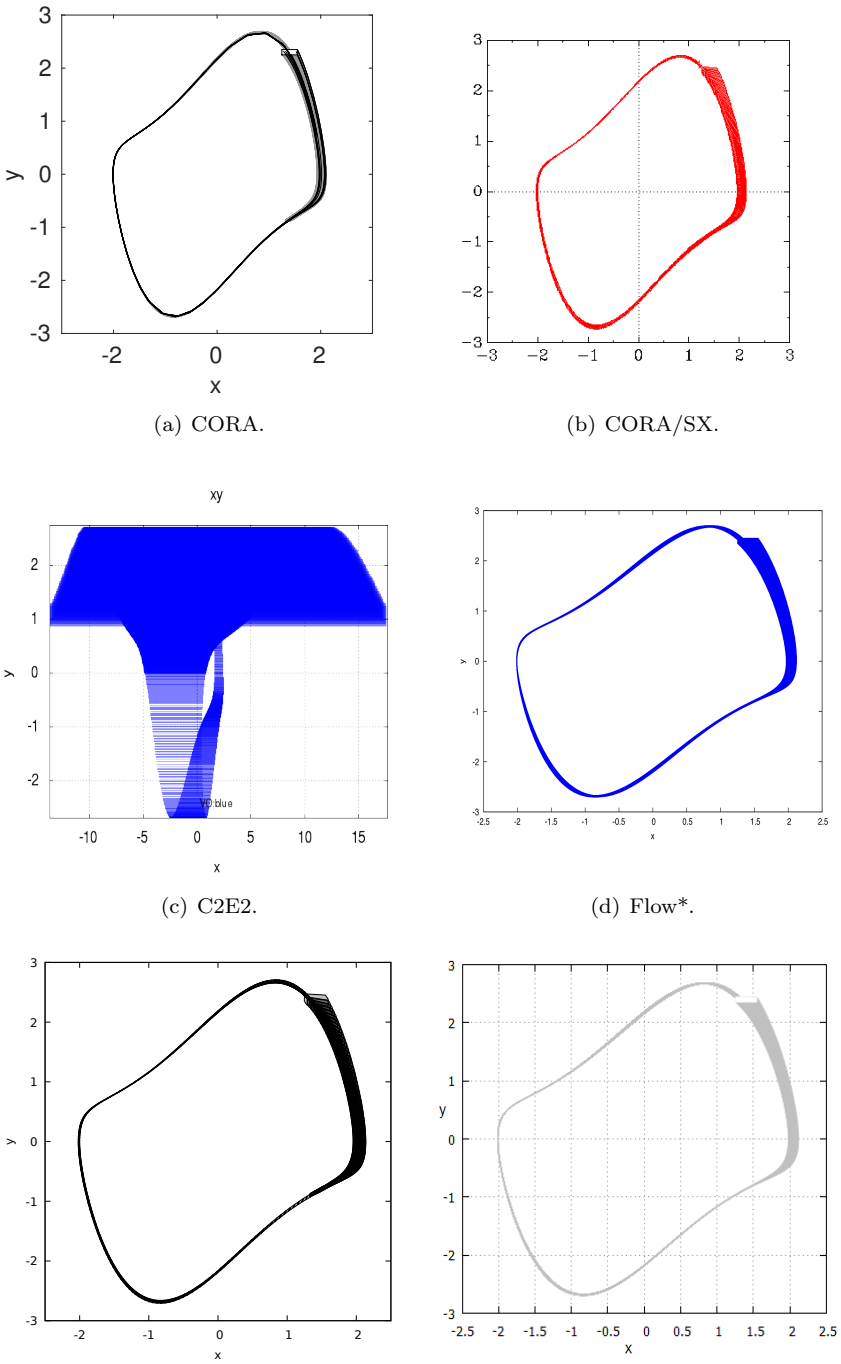
### 3.1.3 Results

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 CORA/SX.** A pseudo invariant at $x = 1.2$ was introduced manually. The time step size is 0.01 and the zonotope order is chosen as 20.

**Setting for C2E2.** C2E2 proved the safety of the model by using time step size 0.01. The K value is chosen to be 1000. The reachtube for variable $x$ is over bloated due to the lack of constraints on variable $x$. The reachtube overapproximation is enough for proving the safety constraints on variable $y$. Note that the result for C2E2 is not optimal since C2E2 is currently been updated.




(a) CORA.

(b) CORA/SX.

(c) C2E2.

(d) Flow*.

(e) Isabelle/HOL.

(f) SymReach.

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

**Setting for Flow*.** Flow* uses the step size 0.04, the TM order 6, 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(d). The approximation quality can be better evaluated based on the remainder size of the last TM flowpipe (see [10]). In this task, the overapproximation error for the last flowpipe is bounded by only 0.01214.

**Setting for Isabelle/HOL.** Maximum Zonotope order is set to 20, Reachability analysis is carried out with an (absolute and relative) error tolerance of $2^{-14}$. A pseudo-invariant is added at $x = 1.5$.

**Setting for SymReach** Step size is set to 0.01, zonotope order limited to 20, $L_{max} = 0.01$ (if CE $\not\subset [-L_{max}, L_{max}]$ then set split).

## 3.2 Laub-Loomis Benchmark

### 3.2.1 Model

The Laub-Loomis model is presented in [21] for studying a class of enzymatic activities. The dynamics can be defined by the following ODE with 7 variables.

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

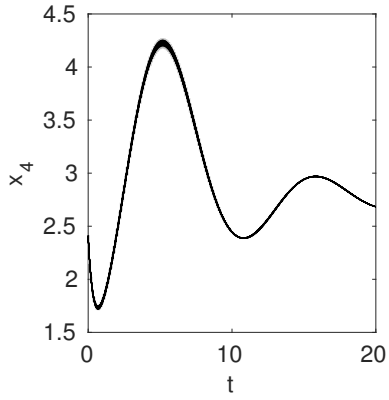
The system is asymptotically stable and the equilibrium is the origin.

### 3.2.2 Specification

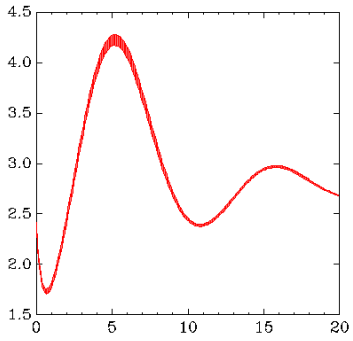
The initial sets are defined according to the one used in [22]. They are boxes centered at $x_1(0) = 1.2$, $x_2(0) = 1.05$, $x_3(0) = 1.5$, $x_4(0) = 2.4$, $x_5(0) = 1$, $x_6(0) = 0.1$, $x_7(0) = 0.45$. The width of the initial set is vital to the difficulty of the reachability analysis job. The larger the initial set the harder the reachability analysis. In the paper, we consider the initial box of the radius $W = 0.01$ and $W = 0.1$, i.e., the range of the box in the $i$th dimension is defined by the interval $[x_i(0) - W, x_i(0) + W]$. 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

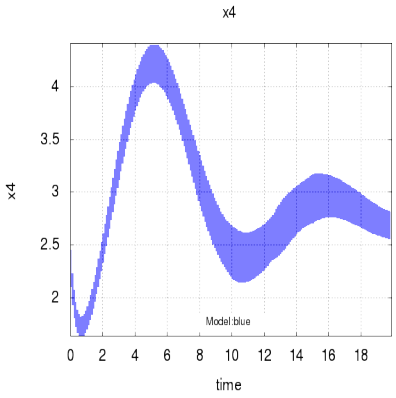
The computation results of the tools are given in Table 2. It can be seen that enlarging the initial set size can greatly make the reachability analysis task harder. The tool settings are given as below. Since the safety condition is only related to the variable $x_4$, we present the plots of projections of the overapproximations in the $t$-$x_4$ plane such that $t$ is the time variable. Figure 2 shows the results for the smaller initial set ($W = 0.01$), Figure 3 for the larger one ($W = 0.1$).



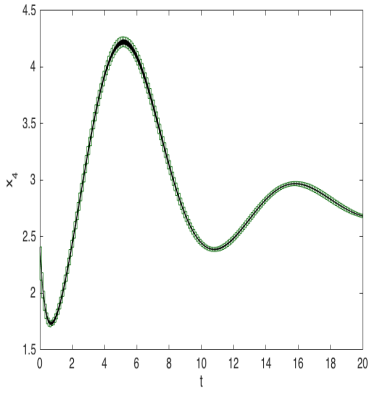

(a) CORA



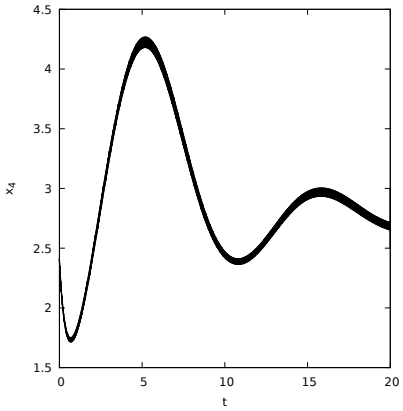
(b) CORA/SX



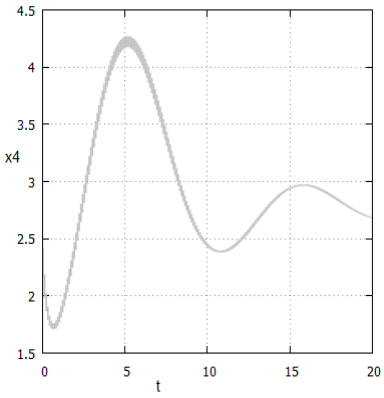

(c) C2E2




(d) Flow*




(e) Isabelle/HOL

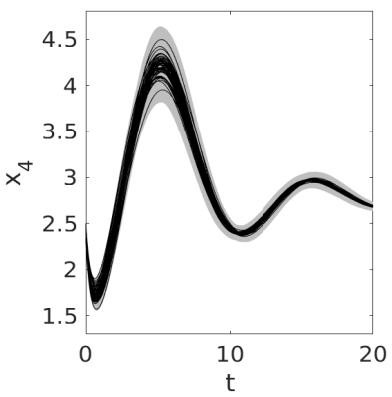



(f) SymReach

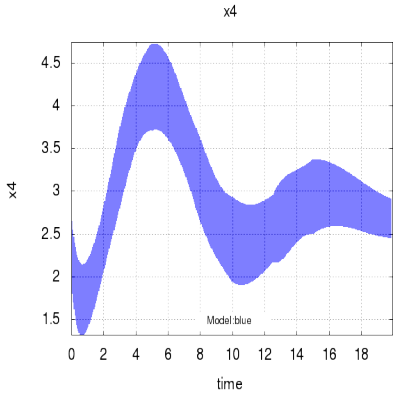
Figure 2: Reachable set overapproximations for the Laub-Loomis model ($W = 0.01$). CORA and Flow* show numerical simulations in black.

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

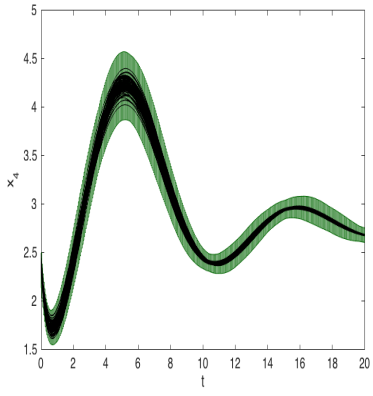
| | **computation time in [s]** | | **platform** | |
|---|---|---|---|---|
| **tool** | $W = 0.01$ | $W = 0.1$ | **language** | **machine** |
| CORA | 0.82 | 63 | MATLAB | $\mathrm{M}_{\mathrm{CORA}}$ |
| CORA/SX | 0.85 | – | C++ | $\mathrm{M}_{\mathrm{SpaceEx}}$ |
| C2E2 | 0.12 | 0.67 | C++ | $\mathrm{M}_{\mathrm{C2E2}}$ |
| Flow* | 4.5 | 12.7 | C++ | $\mathrm{M}_{\mathrm{Flow*}}$ |
| Isabelle/HOL | 10 | – | SML | $\mathrm{M}_{\mathrm{Isabelle}}$ |
| SymReach | 1.93 | – | C++ | $\mathrm{M}_{\mathrm{SymReach}}$ |




(a) CORA




(b) C2E2




(c) Flow*

Figure 3: Reachable set overapproximations for the Laub-Loomis model ($W = 0.1$). CORA and Flow* show numerical simulations in black.

**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 [6] 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 CORA/SX.** Since CORA/SX does not currently support splitting the initial states, only the small initial set was handled. The time step is 0.1 and the zonotope order is 50.

**Setting for C2E2.** For the small initial set, C2E2 uses step size 0.1 and K value 50. For the large initial set, C2E2 uses step size 0.05 and K value 100. Note that the result for C2E2 are not optimal due to the updating of C2E2. Note that the result for C2E2 is not optimal since C2E2 is currently been updated.

**Setting for Flow*.** For the small initial set, Flow* uses the step size 0.1, the TM order 4, the cutoff threshold $10^{-7}$ and the precision 53 for floating-point numbers. The TM flowpipe remainders are kept symbolically every 50 steps. Besides the safety set that is defined by $x_4 \leq 4.5$, the tool even proves that $x_4 \leq 4.3$ is safe. On the other hand, for the large initial set, we use the same setting except that the stepsize is reduced to 0.05, and the remainders are kept symbolically every 100 steps. Besides the given safety property, the tool proves a smaller safe set which is defined by $x_4 \leq 4.6$. The plots of the flowpipes are shown in Figure 2(d) and 3(c). 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 overapproximation error of the last flowpipe is only 0.01044 which is determined by the $x_4$-dimension, while for the large initial set, the corresponding maximum error is only 0.06016.

**Setting for Isabelle/HOL.** Maximum Zonotope order is set to 60. Reachability analysis is carried out with an (absolute and relative) error tolerance of $2^{-12}$. Compared to last year, we dropped the analysis of the larger initial set, since this required many subdivisions and is too inefficient.

**Setting for SymReach.** Step size is set to 0.1, zonotope order limited to 40, $L_{max} = 0.05$.

## 3.3 Quadrotor Benchmark

### 3.3.1 Model

We study the dynamics of a quadrotor as derived in [7, 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 $m = M + 4M_{rotor}$.

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 [7, 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

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

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 4 and Table 3. We give the tool settings below.

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

| **tool** | **computation time in [s]** | **language** | **machine** |
|---|---|---|---|
| CORA | 5.2 | MATLAB | $M_{CORA}$ |
| CORA/SX | 1.5 | C++ | $M_{SpaceEx}$ |
| Flow* | 5.9 | C++ | $M_{Flow*}$ |
| Isabelle/HOL | 30 | SML | $M_{Isabelle}$ |
| SymReach | 2.96 | C++ | $M_{SymReach}$ |

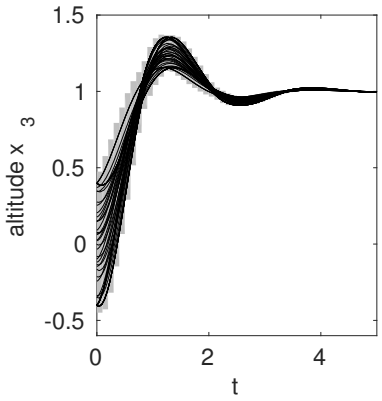
**Setting for CORA.** CORA uses the step size 0.1 and the zonotope order 50. The computation is carried out using the approach in [6], which conservatively linearizes the system dynamics for each consecutive time interval by adding the linearization error as an uncertain input. The linearization error 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 CORA/SX.** CORA/SX uses the step size 0.05 and the zonotope order 50. Note that the plot of CORA/SX in Fig. 4(b) does not have a staircaise form like CORA and SymReach, because time needs to be added as a state variable for CORA/SX to be able plot over time.

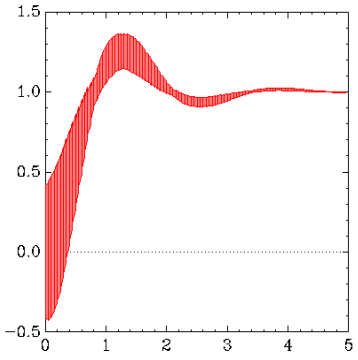
**Setting for Flow*.** Flow* uses the step size 0.025, the TM order 4, the cutoff threshold $10^{-6}$ and the precision 53 for floating-point numbers. The TM flowpipe remainders are kept symbolically every 20 steps. All floating-point roundoff errors are included in the overapproximation. Figure 4(c) illustrates the interval overapproixmations for the flowpipes. To better evaluate the approximation error, we provide the maximum overapproximation error of the last flowpipe which is below 0.0003103. Besides, the altitude at $t = 5$ is below 1 according to the computed flowpipes.

**Setting for Isabelle/HOL.** Maximum Zonotope order is set to 25. Reachability analysis is carried out with an (absolute and relative) error tolerance of $2^{-10}$.

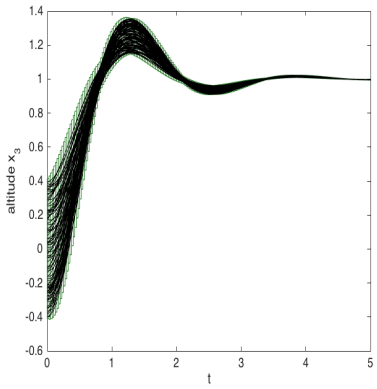
**Setting for SymReach.** Step size is set to 0.1, zonotope order limited to 5, $L_{max} = 0.05$.



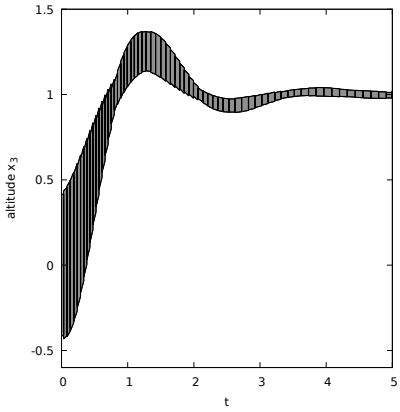

(a) CORA.



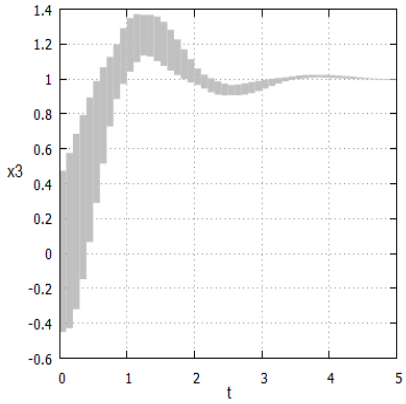
(b) CORA/SX.




(c) Flow*.




(d) Isabelle/HOL.




(e) SymReach.

Figure 4: Reachable set overapproximations for the quadrotor model.

## 3.4 Space Rendezvous Benchmark

### 3.4.1 Model

Spacecraft rendezvous is a perfect use case for formal verification of hybrid systems with nonlinear dynamics since mission failure can cost lives and is extremely expensive. This benchmark is taken from [9]. A version of this benchmark with linearized dynamics is verified in the ARCH-COMP category *Continuous and Hybrid Systems with Linear Continuous Dynamics*. The nonlinear dynamic equations describe the two-dimensional, planar motion of the spacecraft on an orbital plane towards a space station:

$$\begin{cases} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v_x} &= n^2 x + 2nv_y + \frac{\mu}{r^2} - \frac{\mu}{r_c^3}(r+x) + \frac{u_x}{m_c} \\ \dot{v_y} &= n^2 y - 2nv_x - \frac{\mu}{r_c^3} y + \frac{u_y}{m_c} \end{cases}$$

The model consists of position (relative to the target) $x, y$ [m], time $t$ [min], as well as horizontal and vertical velocity $v_x, v_y$ [m / min]. The parameters are $\mu = 3.986 \times 10^{14} \times 60^2$ [m$^3$ / min$^2$], $r = 42164 \times 10^3$ [m], $m_c = 500$ [kg], $n = \sqrt{\frac{\mu}{r^3}}$ and $r_c = \sqrt{(r+x)^2 + y^2}$.

The hybrid nature of this benchmark origins from a switched controller. In particular, the modes are *approaching* ($x \in [-1000, -100]$ [m]), *rendezvous attempt* ($x \geq -100$ [m]), and *aborting* (time $t \geq 120$ [min]). The linear feedback controllers for the different modes are defined as $\binom{u_x}{u_y} = K_1 \underline{x}$ for mode *approaching*, and $\binom{u_x}{u_y} = K_2 \underline{x}$ for mode *rendezvous attempt*, where $\underline{x} = \begin{pmatrix} x & y & v_x & v_y \end{pmatrix}^T$ is the vector of system states. The feedback matrices $K_i$ were determined with an LQR-approach applied to the linearized system dynamics, which resulted in the following numerical values:

$$K_1 = \begin{pmatrix} -28.8287 & 0.1005 & -1449.9754 & 0.0046 \\ -0.087 & -33.2562 & 0.00462 & -1451.5013 \end{pmatrix}$$

$$K_2 = \begin{pmatrix} -288.0288 & 0.1312 & -9614.9898 & 0 \\ -0.1312 & -288 & 0 & -9614.9883 \end{pmatrix}$$

In the mode *aborting* the system is uncontrolled $\binom{u_x}{u_y} = \binom{0}{0}$.

### 3.4.2 Specification

The spacecraft starts from the initial set $x \in [-925, -875]$ [m], $y \in [-425, -375]$ [m], $v_x = 0$ [m/min] and $v_y = 0$ [m/min]. For the considered time horizon of $t \in [0, 200]$ [min], the following specifications have to be satisfied:

- **Line-of-sight:** In mode *rendezvous attempt*, the spacecraft has to stay inside line-of-sight cone $\mathcal{L} = \{\binom{x}{y} \mid (x \geq -100) \wedge (y \geq x \tan(30°)) \wedge (-y \geq x \tan(30°))\}$.
- **Collision avoidance:** In mode *aborting*, the spacecraft has to avoid a collision with the target, which is modeled as a box $\mathcal{B}$ with 0.2m edge length and the center placed at the origin.
- **Velocity constraint:** In mode *rendezvous attempt*, the absolute velocity has to stay below 3.3 [m/min]: $\sqrt{v_x^2 + v_y^2} \leq 3.3$ [m/min].

Table 4: Results of the spacecraft rendezvous model. Details of the platforms are described in Section A.

| **tool** | **computation time in [s]** | **language** | **machine** |
|---|---|---|---|
| CORA | 14.8 | MATLAB | $M_{CORA}$ |
| C2E2 | 29.18 | C++ | $M_{C2E2}$ |
| Flow* | 18.7 | C++ | $M_{Flow*}$ |
| Isabelle/HOL | 395 | SML | $M_{Isabelle}$ |

**Remark on velocity constraint** In the original benchmark [9], the constraint on the velocity was set to 0.05 m/s, but it can be shown (by a counterexample) that this constraint cannot be satisfied. We therefore use (just like the ARCH-COMP category *Continuous and Hybrid Systems with Linear Continuous Dynamics*) the relaxed constraint 0.055 [m/s] = 3.3 [m/min].

### 3.4.3 Results

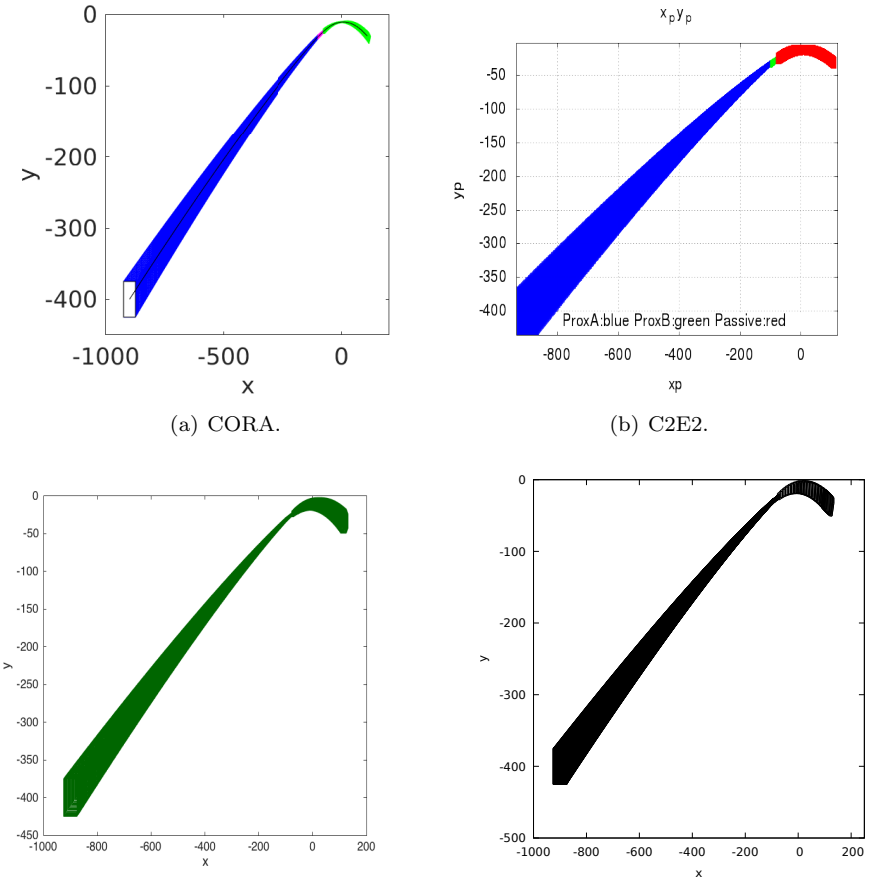
The results of the reachability computation for the spacecraft rendezvous model are given in Figure 5 and Table 4, with the tool settings below.

**Setting for CORA.** CORA was run with a time step size of 0.2 [min] for the modes *approaching* and *aborting*, and with a time step size of 0.05 [min] for mode *rendezvous attempt*. The intersections with the guard sets were calculated with the method of Girard, which was introduced in [17]. In order to find suitable orthogonal directions, the last zonotope that did not intersect the guard set is projected onto the hyperplane that represents the guard set. Then, Principal Component Analysis is applied to the generators of the projected zonotope.

**Setting for C2E2.** C2E2 can verify this model when the abortion mode is linearized. Since C2E2 cannot take nonlinear unsafe set specifications, the thrust constraint is safely underapproximated by an octagon. Moreover, since C2E2 cannot handle unsafe set specifications with logical conjunction, each constraint is checked separately and the sum of run time for each individual constraint represents the total run time for verifying the model. The time step used to solve the model is 0.1 and the K value used is 2000. Note that the result for C2E2 is not optimal since C2E2 is currently been updated.

**Setting for Flow*.** The model can be directly verified by Flow* with the following setting for parameters: the TM order is fixed by 5, the stepsize is adaptively selected in the range from 0.001 to 0.5, the remainder estimation is the interval $[-10^{-3}, 10^{-3}]$ in all dimensions, and the cutoff threshold is $10^{-6}$. Besides, we use the precision 100. We simply aggregate the intersections after each jump by a box instead of a parallelotope which is more time-costly to compute but more accurate, since it is already sufficient to prove the property.

**Setting for Isabelle/HOL.** Isabelle/HOL does not support hybrid systems automatically. One can, however, compute the reachable sets for each mode seperately. The intersection with the guard set is computed with the method of Girard [17], simply using axis-aligned orthogonal directions, which results in box enclosures. We verify that the transition to mode *rendezvous attempt* occurs at $t \in [108.66, 111.71]$, $x = -100$, $y \in [-35.04, -28.43]$, $v_x \in [1.99365, 2.00644]$,




(a) CORA. (b) C2E2.

(c) Flow*. (d) Isabelle/HOL.

Figure 5: Reachable set of the spacecraft position in the $x$-$y$-plane. CORA shows simulations in black. CORA and C2E2 use different colors to encode different modes of the hybrid system. C2E2 with abortion mode linearized.

$v_y \in [0.6489, 0.8130]$. Starting from this box, the transition to mode *aborting* occurs at $t = 120$, $x \in [-78.02, 71, 20]$, $y \in [-27.34, -20.23]$, $v_x \in [2.135, 2.341]$, $v_y \in [0.603, 0.825]$. From there we compute the reachable set until time $t = 200$, which satisfies $y < -1$ [m]. In those computations, maximum zonotope order is set to 50 and we use absolute and relative error tolerance of $2^{-10}$.

# 4 Conclusion and Outlook

In last year's competition [14], we promised to include hybrid benchmarks and hoped that more tools would join the competition. Indeed, three new tools (C2E2, CORA/SX, SymReach) participated. Hybrid dynamics is included in the new benchmark Space Rendezvous (section 3.4). A linearized version of the Space Rendezvous benchmark was used in the ARCH-COMP category *Continuous and Hybrid Systems with Linear Continuous Dynamics*.

Triggered by the participation in this competition, individual tools made progress and can now solve benchmarks that were previously out of reach:

- This is the first time **C2E2** participates in the competition. The results for C2E2 are not optimal because a new version of C2E2 is about to be released with the experience gained during the competition.
- The tool **Flow*** has been improved in the following aspects since the last competition. Firstly, we optimized some interval computation procedure to overall improve the performance. Secondly, the C++ API of the tool is under development, we currently exposed all functions for handling LTI and LTV dynamics as well as linear constraints. It provides not only a much more flexible way to handle different systems, but also a possibility to cooperate with program analysis tools to handle more complex controlled systems. Besides, Flow* will also support the formal verification of continuous systems with neural network controllers.
- Compared to last year, **Isabelle/HOL** includes affine arithmetic approximations for some transcendental functions $(\sin, \cos, \exp, \sqrt{\ })$ and can therefore now solve the Quadrotor benchmark.

A direct outcome of this competition is that one has a means of comparing the available tools for nonlinear reachability analysis. An indirect outcome is that individual tools made progress and more tools and algorithms are available. The direct and indirect outcomes of this competition serve the whole community interested in developing and applying tools for reachability analysis of nonlinear systems: We can claim that this competition helps to advance and promote the state-of-the-art of tools for nonlinear reachability analysis.

# 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 i7-7820HQ CPU @ 2.90GHz x 4
- Memory: 32 GB
- Average CPU Mark on www.cpubenchmark.net: 9409 (full), 2070 (single thread)

## A.2 $\mathbf{M}_{C2E2}$

- Processor: Intel Core i5-3470 CPU @ 3.20GHz x 4
- Memory: 8 GB
- Average CPU Mark on www.cpubenchmark.net: 6680 (full), 1915 (single thread)

### A.3 $\mathbf{M_{Flow*}}$

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.4 $\mathbf{M_{Isabelle}}$

- Processor: Intel Core i7-8750H CPU @ 2.20GHz x 6
- Memory: 16 GB 2666 MHz DDR4
- Average CPU Mark on www.cpubenchmark.net: 12,516 (full), 2368 (single thread)

### A.5 $\mathbf{M}_{SpaceEx}$

- Processor: Intel Core i7-7920HQ CPU @ 3.1GHz x 4
- Memory: 16 GB
- Average CPU Mark on www.cpubenchmark.net: 10230 (full), 2161 (single thread)

### A.6 $\mathbf{M_{SymReach}}$

- Processor: Intel Core Duo T2250 CPU @ 1.73GHz x 2
- Memory: 1.5 GB
- Average CPU Mark on www.cpubenchmark.net: 758 (full), 547 (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, C. Le Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Hybrid Systems: Computation and Control*, pages 93–102, 2011.

[6] 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.

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

[8] 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.

[9] N. Chan and S. Mitra. Verifying safety of an autonomous spacecraft rendezvous mission. In *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems, collocated with Cyber-Physical Systems Week (CPSWeek) on April 17, 2017 in Pittsburgh, PA, USA*, pages 20–32, 2017.

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

[11] 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.

[12] 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.

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

[14] Xin Chen, Matthias Althoff, and Fabian Immler. ARCH-COMP17 category report: Continuous systems with nonlinear dynamics. In Goran Frehse and Matthias Althoff, editors, *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems*, volume 48 of *EPiC Series in Computing*, pages 160–169. EasyChair, 2017.

[15] P.S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2e2: A verification tool for stateflow models. In *TACAS*, 2015.

[16] C. Fan, B. Qi, S. Mitra, M. Viswanathan, and P.S. Duggirala. Automatic reachability analysis for nonlinear hybrid models with c2e2. In *Computer Aided Verification*, pages 531–538, Cham, 2016. Springer International Publishing.

[17] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proc. of Hybrid Systems: Computation and Control*, LNCS 4981, pages 215–228. Springer, 2008.

[18] Darius Grabowski, Markus Olbrich, and Erich Barke. Analog circuit simulation using range arithmetics. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 762–767. IEEE Computer Society Press, 2008.

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

[20] F. Immler and J. Hölzl. Ordinary differential equations. *Archive of Formal Proofs*, July 2018. http://isa-afp.org/entries/Ordinary_Differential_Equations.shtml, Formal proof development.

[21] 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.

[22] 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.