Work in Progress: Reachability Analysis for Time-triggered Hybrid Systems, The Platoon Benchmark

François Bidet
LIX, École polytechnique, CNRS
Université Paris-Saclay
91128 Palaiseau, France
francois.bidet@polytechnique.edu

Éric Goubault
LIX, École polytechnique, CNRS
Université Paris-Saclay
91128 Palaiseau, France
goubault@lix.polytechnique.fr

Sylvie Putot
LIX, École polytechnique, CNRS
Université Paris-Saclay
91128 Palaiseau, France
putot@lix.polytechnique.fr

Abstract

This article presents an extension of the method of [1] to time-triggered hybrid systems, providing over- and under-approximations of the set of reachable states. Our results on the vehicles platoon benchmark [2] compare favorably to the state of the art tools Flow* and SpaceEx, with more precise over-approximations. Moreover, we provide a measure of the approximation's accuracy using the ratio of under- to over-approximation.

1 Introduction and presentation of the platoon benchmark

Different algorithms have been proposed to compute safe approximations of reachable states of hybrid systems, subject to uncertain initial conditions, parameters, and environment. These reachable states are key in analyzing the behavior of systems, for instance for proving that a region of the state-space will eventually be reached, while avoiding some unsafe set of states. In this short paper, we experimentally compare the behavior of different reachability tools on a benchmark that was specifically chosen to study their behavior in case of time-triggered transitions. We chose as benchmark the vehicle platoon proposed in the ARCH workshop [2] for applied verification of hybrid systems. We will compare the results obtained on our small Taylor methods based reachability prototype with state of the art reachability tools Flow* and SpaceEx.

The model represents a platoon of three vehicles following a leader, a vehicle which can be controlled by a human, on a one-dimensional road. As illustrated in Figure 1, each vehicle i is localized with respect to its predecessor by a distance $d_i = d_{ref,i} + e_i$ with $d_{ref,i}$ the reference distance and e_i the spacing error. Because the reference distance $d_{ref,i}$ is constant, the dynamic is independent of $d_{ref,i}$, each vehicle i is described by its acceleration a_i, its spacing error e_i and the variation of this error $\dot{e}_i$. The system is driven by the leader and described by its acceleration a_L.

The behavior of the system can be described as a simple hybrid system with two modes : the first one, q_c (see Figure 2b), in which each vehicle has access to information about the whole system, by communicating with each other, and the second one, q_n (see Figure 2c), in which each vehicle has just access to its own state, because of a loss of communication. The switch between the two modes is time-triggered (i.e. each guard only depends on time and not on state variables), as shown in Figure 2a.

The goal of this benchmark is to find the minimum reference distance between vehicles such that we can ensure that for all i, $d_i > 0$, which is equivalent to $e_i > -d_{ref,i}$. If $e_i \leq -d_{ref,i}$ then $d_i \leq 0$ and vehicule i collides with vehicle $i-1$ (or the leader if $i = 1$).

In: O. Hasan, S. Tahar, U. Siddique (eds.): Proceedings of the Workshop Formal Verification of Physical Systems (FVPS), Hagenberg, Austria, 17-Aug-2018, published at http://ceur-ws.org

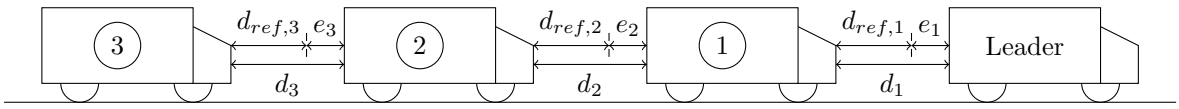
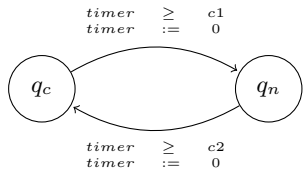


Figure 1: Platoon Benchmark description



(a) Platoon automaton

Dynamics
$\dot{X} = A_c X$
Invariant
$timer \leq c_1$

(b) Mode q_c

Dynamics
$\dot{X} = A_n X$
Invariant
$timer \leq c_2$

(c) Mode q_n

Figure 2: Platoon modes and automaton

Unlike in article [2], we choose here a full loss of communication in mode q_n, which means that all vehicles lose communication. It results in a slightly different matrix A_n. Also, compared to the results given in [2], we will investigate several transitions between the two modes (with and without communication) of the system.[1]

2 Approaches and tools for reachability analysis

We developed a prototype extending the method of [1] for reachability analysis of uncertain continuous systems, to the case of time-triggered hybrid systems, that is hybrid systems that change modes at certain specified times.

To compute the continuous state in one mode of the system, we use a combination of affine arithmetic [3] and Taylor models [4], as in [1]:

affine arithmetic: Each uncertain quantity $x \in \mathbb{R}^n$ is represented by $x = x_0 + \sum_{i=1}^{p} x_i \epsilon_i$ with for all $i = 0, \ldots, p$ $x_i \in \mathbb{R}^n$ and $\epsilon_i \in [-1, 1]$. Unlike interval arithmetic, it allows to encode dependencies between different quantities, e.g. let $x = 1 + 2\epsilon_1 \in [-1, 3]$, then $x - x = 1 - 1 + (2 - 2)\epsilon_1 = 0$ in affine arithmetic, whereas $[-1, 3] - [-1, 3] = [-4, 4]$.

Taylor model: A Taylor model of function x of $t \in [0, T]$ is a pair (p, I) where p is a polynomial in t and I is a set called remainder (typically an interval) such that $\forall t \in [0, T], x(t) \in p(t) + I$.

We improved the method of [1] by adding the possibility for the analyzer to adapt the time step by the naive algorithm: if no over-approximation is found with the interval Picard operator [4], we reduce the time step by half, otherwise we enlarge the time step by 10%. Because our prototype is based on Taylor models, the time is an implicit variable in our system, and the state at a given time instant can be precisely bounded by evaluating the Taylor model at that time. Time-triggered transitions can thus be evaluated without any loss of precision.

Moreover, in addition to over-approximations, our prototype is able to compute under-approximations (as will be demonstrated in Section 3.3) following the algorithm of [1] : we thus compute an over-approximation of the solution $[\boldsymbol{z}](t, \widetilde{z_0})$ for a particular initial condition $\widetilde{z_0} \in \boldsymbol{z_0}$ and an over-approximation of the Jacobian matrix of the solution $[\boldsymbol{J}](t, \boldsymbol{z_0})$ at each time t. Similarly, the interpretation of guards for time-triggered systems does not lose any precision.

The prototype has been programmed in C++ with the library "aaflib" [5] for the affine forms and the library "FADBAD++" [6] for the derivations (Taylor models and Jacobian matrix).

Other reachability tools experimented here

Flow*[7] also uses Taylor Models [8], but there are some significant differences with our approach. First, the Taylor models are not built and evaluated in affine arithmetic like in our approach. Moreover, the physical time of the system is not directly accessible to evaluate transitions' guards. We have to add a "timer" variable to know how much time we passed in each mode, as shown in Figure 2a. The relation between state variables

[1] The models and sources of experiments are downloadable on `http://www.lix.polytechnique.fr/Labo/Francois.Bidet/`

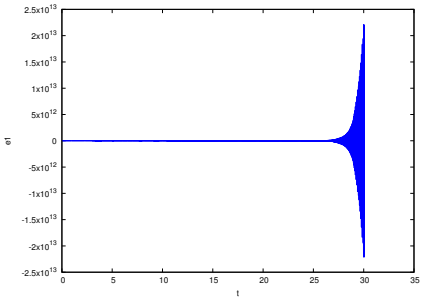
and system time are thus not as direct as in our approach, and transitions become non-deterministic, inducing over-approximations.

SpaceEx [9, 10] uses different (sub-)polyhedra to over-approximate states combined with different approaches to model time dependence. Like for Flow*, dependence of the system to physical time is not directly expressed, and accuracy is lost in time-triggered transitions. Also, contrarily to Flow* and our approach, it is restricted to affine systems.

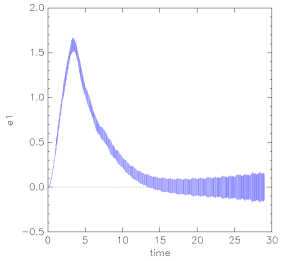
3 Experiments[2]

3.1 Transient, but certain acceleration

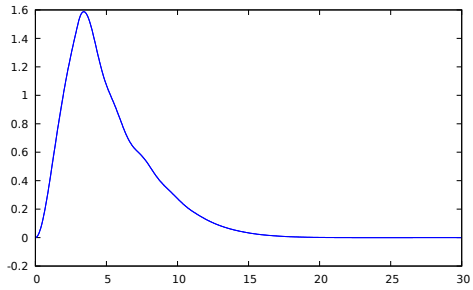
The original benchmark [2] presents some results for a velocity step of 1 m/s. Here, we chose to add a transient acceleration for the leader of 1 $m.s^{-2}$ during 3 seconds (a velocity variation of 3 $m.s^{-1}$ in total). We suppose that vehicles lose communication during 1 second every 2 seconds (i.e. $c_1 = 1$ and $c_2 = 1$ in Figure 2). As shown



(a) Flow*: time horizon = 30 s (computation time = 292 seconds)



(b) SpaceEx: time horizon = 30 s (computation time = 22 seconds)



(c) Our Prototype: time horizon = 30 s (computation time 34 seconds)

Figure 3: e_1 as a function of time for a transient acceleration of 3 seconds

in Figure 3a, Flow* produces an over-approximation which clearly diverges. SpaceEx (Figure 3b) produces an over-approximation wich width increases with time. Our prototype (Figure 3c) on the contrary is able to prove that the system stabilizes to a zero error e_1: the width of the over-approximation is of the order of 10^{-9} at time 30 s.

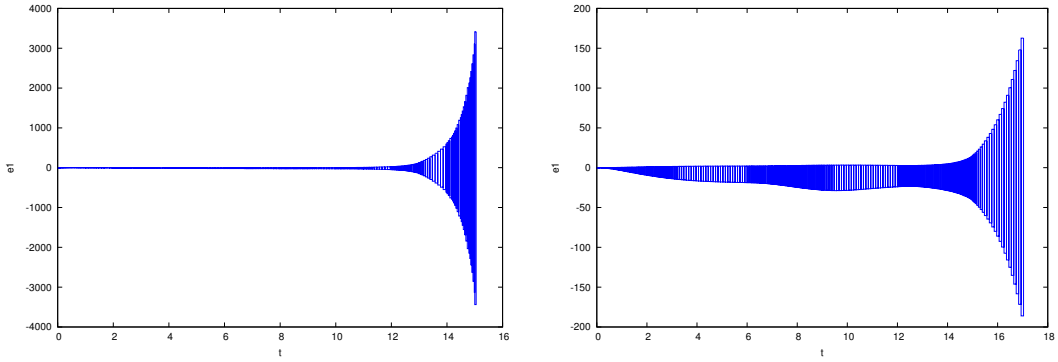
3.2 Constant, but uncertain acceleration

We now set $c_1 = c_2 = c$ (same duration of communication and loss of communication) and we run multiple analyzes with various values for c, in $[1, 10]$. We also assume that the leader acceleration a_L is uncertain but constant in $[-9, 1]$. We fix initial state variables to 0 (no spacing error and no acceleration), except for a_L that is set to be the interval $[-9, 1]$. As mentioned in section 1, each vehicle is described using 3 variables (the spacing error e_i from the reference distance, the derivative of this error $\dot{e}_i$ and acceleration of the vehicule a_i). We only present results for e_1 as a function of time in this section, but we obtain similar results with e_2 and e_3.

Settings and results with Flow*

We use an adaptative time step between 10^{-5} and 10^{-1} seconds, a remainder estimation of 10^{-8}, a cutoff threshold of 10^{-12} and a precision of 100 bits (for the MPFR library). These parameter values were obtained after several tests, in order to make the simulation successful (by reducing the time step) and reduce divergence (by adapting the cutoff threshold and remainder estimation). To reduce divergence of the over-approximation, we increased the Taylor Model order up to 16. Results of simulations however still diverge after about 14 seconds as shown in Figure 4. Interestingly, the analysis takes more time with order 4 Taylor Models (we stopped it after 10 minutes) than order 16 for which results for same time horizon 17 were obtained in 106 seconds. We expect the analysis is refining a lot the time step in order to try to control the increasing approximation error.

[2] All experiments can be downloaded on `http://www.lix.polytechnique.fr/Labo/Francois.Bidet/`



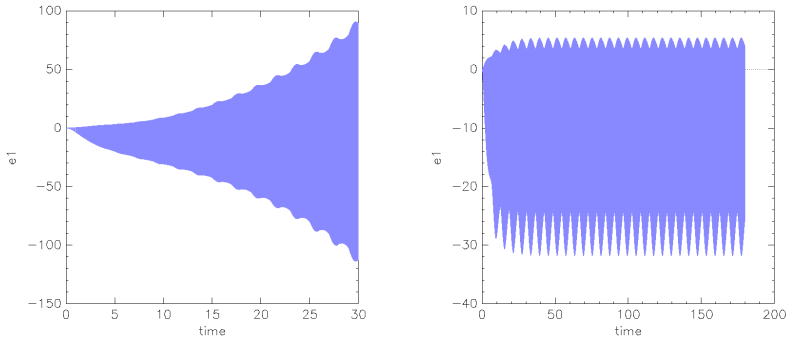
(a) $c = 1$, time horizon = 15 s (computation time 106 seconds)

(b) $c = 3$, time horizon = 17 s (computation time = 106 seconds)

Figure 4: Flow*: e_1 as a function of time

Settings and results with SpaceEx

We used the STC Support Function with octagons, without set aggregation, an absolute flowpipe tolerance of 1 and a relative of 0, a local time horizon of 5 s (for $c < 5$), a maximum number of iterations of 30, a relative error of 10^{-12} and an absolute error of 10^{-15}. We reduced the absolute flowpipe tolerance to 10^{-2} without significative difference. We also used uniformly distributed directions instead of octagons (with 12 directions) but had to stop the simulation after about 10 minutes of computation (with less than 12 directions, we obtained an error). Figure 5 shows that the analysis behaves well for $c = 3$, but diverges for fast switching, that is with



(a) $c = 1$, time horizon = 30 s (computation time = 123 seconds)

(b) $c = 3$, time horizon = 180 s (computation time = 394 seconds)

Figure 5: SpaceEx: e_1 as a function of time

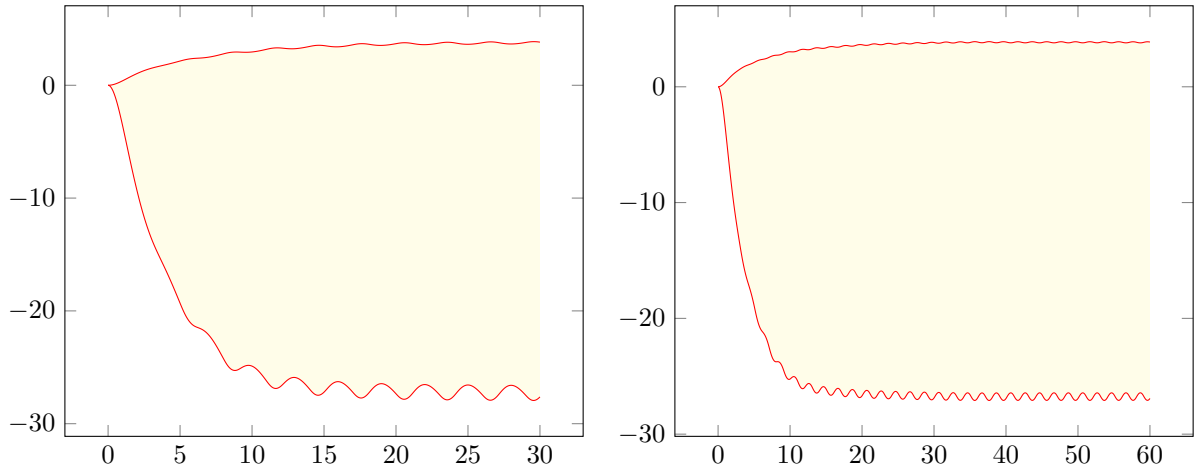
$c = 1$, confirming that some accuracy is lost in transitions.

Comparing accuracy and efficiency with our prototype

As shown in Figure 6, our prototype was able to compute an over-approximation of e_1 without significant divergence even when c is small, e.g. when $c = 1$. We used Taylor models of order 4, an adaptative time step between 10^{-5} s and 10^{-1} s, and a cutoff threshold of 10^{-5} (for each affine form, we regroup all noise terms with coefficient smaller than this threshold into a single noise term). The two analyzes represented in Figure 6 seem to reach an almost periodic behavior, which is the expected behavior.

For a switching time period $c = 3$, our prototype's over-approximation is slightly more accurate than that of SpaceEx but both are comparable. But when the switching frequency is higher ($c = 1$), our prototype is clearly more accurate than SpaceEx and Flow*. As already hinted, these tools lose precision when localizing and taking the time-triggered transitions.

A time comparison is difficult because we used order 4 Taylor models in our prototype and order 16 Taylor model in Flow* (but the latter is faster than its own order 4 analysis with same other parameters). But we can note that for $c = 1$, our analysis takes 36 seconds for time horizon 30, while Flow* takes 106 seconds for time



(a) $c = 1$, time horizon = 30 s (computation time 36 seconds)

(b) $c = 3$, time horizon = 180 s (computation time = 142 seconds)

Figure 6: Our prototype: e_1 as a function of time

horizon 15, for less accurate results. SpaceEx is running on a virtual machine with only 4Go of RAM against 8Go for our machine. In this setting, our prototype is between two and three times faster on this example.

3.3 Under-approximation

In addition to the over-approximation, our prototype produces an under-approximation of the reachable states (see the case of purely continuous systems in [1]). An under-approximation (or inner-approximation) corresponds to a set of states that are sure to be reached from some initial value.

The under-approximation is useful as a complement of an over-approximation in many applications. It can be used to obtain a measure of precision of the approximation (e.g. the ratio of the under- to the over-approximations). It can also be used to invalidate some properties. Over-approximations ensure that if they do not intersect a set of "bad" states, then the systems' executions are safe. In the case of the platoon benchmark, such "bad" states are when the distances between two consecutive vehicles become negative ($e_i < -d_{ref,i}$). If the under-approximation intersects the set of bad states, then we are bound to run into an erroneous state, for some initial state. We can see on Figure 7b that if we fix $d_{ref,3} \geq 11$, we will never have a collision between vehicles 2 and 3 in the next 60 seconds following the acceleration of the leader $a_L \in [-9, 1]$ (over-approximation). Conversely, if we fix $d_{ref,3} = 10$, we are sure that there is a value of the acceleration of the leader $a_L \in [-9, 1]$ for which we will have a collision between vehicles 2 and 3 (under-approximation at time 7 seconds).

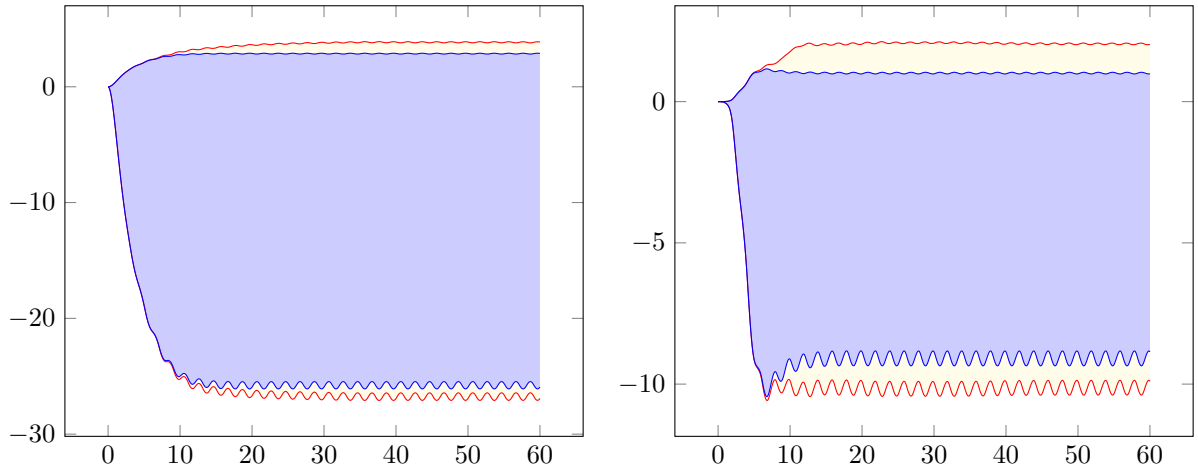
As in [1], we can also use the ratio of the under-approximation's width to the over-approximation's width as an indicator of the accuracy of our approximations. The higher that ratio, the better the approximations are. If the ratio is equal to 1, then we have the exact set of reachable states. For instance, in Figure 7a, the ratio decreases from 1 (the set of initial states is known) to about 0.935 where it levels off. So the under-approximation of e_1 contains more than 93.5% of the real set of reachable states at the end of the simulation.

4 Conclusion and future work

Our prototype is able to compute over-approximation of reachable states of time-triggered system faster and with more accuracy than Flow* and SpaceEx for a challenging benchmark (the platoon benchmark). This is because we use a combination of affine arithmetic and Taylor models, with adaptative time steps, and because we use the time value as part of our Taylor models and not as an extra state variable, making guards for such time-triggered hybrid systems much simpler to interpret. The under-approximation allows us to refute hypotheses and to define an accuracy measure of our approximation as a ratio of under- to over-approximation's width.

We do not yet handle problems due to floating-point arithmetic. They potentially result in small errors on the over- and under-approximations' widths. This can be taken care of by carefully rounding (in the right direction) the affine forms coefficients, this will be included in the prototype in the near future.

Another limitation of our prototype is the exact time of transition guards. The next step of our development is the analysis of arbitrary switching, e.g. guards like $t \in [t_1, t_2]$ or $t \geq t_1$. Once done, we will be able to compare



(a) e_1 as a function of time

(b) e_3 as a function of time

Figure 7: Our Prototype: over- and under-approximation (for $c = 1$; computation time 239 seconds)

our performance with other platoon models (arbitrary loss of communication or loss at nondeterministic times) presented in [11].

References

[1] S. Putot E. Goubault. Forward inner-approximated reachability of non-linear continuous systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, HSCC '17, pages 1–10, New York, NY, USA, 2017. ACM.

[2] I. B. Makhlouf and S. Kowalewski. Networked cooperative platoon of vehicles for testing methods and verification tools. In *ARCH*, volume 34 of *EPiC Series in Computing*, pages 37–42, 2015.

[3] L. H. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37(1):147–158, Dec 2004.

[4] K. Makino M. Berz. Verified integration of odes and flows using differential algebraic methods on high-order taylor models. *Reliable Computing*, 4(4):361–369, Nov 1998.

[5] aaflib - an affine arithmetic c++ library. `http://aaflib.sourceforge.net`.

[6] O. Stauning and C. Bendtsen. Fadbad++ - flexible automatic differentiation using templates and operator overloading in c++. `http://www.fadbad.com/fadbad.html`.

[7] X. Chen and S. Sankaranarayanan. Decomposed reachability analysis for nonlinear systems. In *RTSS*, pages 13–24, 2016.

[8] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *RTSS*, pages 183–192, Dec 2012.

[9] C. Le Guernic G. Frehse, R. Kateja. Flowpipe approximation and clustering in space-time. In *HSCC*, pages 203–212. ACM, 2013.

[10] G. Frehse A. Donzé. Modular, hierarchical models of control systems in spaceex. In *ECC*, Zurich, Switzerland, 2013.

[11] M. Althoff, S. Bak, D. Cattaruzza, X. Chen, G. Frehse, R. Ray, and S. Schupp. Arch-comp17 category report: Continuous and hybrid systems with linear continuous dynamics. In *ARCH*, volume 48 of *EPiC Series in Computing*, pages 143–159, 2017.