

Recent Progress in Continuous and Hybrid Reachability Analysis

Eugene Asarin, Thao Dang, Goran Frehse, Antoine Girard, Colas Le Guernic and Oded Maler

Abstract—Set-based reachability analysis computes all possible states a system may attain, and in this sense provides knowledge about the system with a completeness, or coverage, that a finite number of simulation runs can not deliver. Due to its inherent complexity, the application of reachability analysis has been limited so far to simple systems, both in the continuous and the hybrid domain. In this paper we present recent advances that, in combination, significantly improve this applicability, and allow us to find better balance between computational cost and accuracy. The presentation covers, in a unified manner, a variety of methods handling increasingly complex types of continuous dynamics (constant derivative, linear, nonlinear). The improvements include new geometrical objects for representing sets, new approximation schemes, and more flexible combinations of graph-search algorithm and partition refinement. We report briefly some preliminary experiments that have enabled the analysis of systems previously beyond reach.

I. INTRODUCTION

Complex systems that involve computers that interact with a physical external environment consist of heterogeneous components that may include software, digital hardware, analog hardware, sensors and actuators. Mathematical models of such systems and of the external environment they are supposed to control are indispensable during the design phase, as they allow to explore the behavior of the system either analytically or by simulation.

Hybrid systems are the result of the marriage of the two most commonly used models of *dynamical systems*, namely continuous dynamical systems defined by *differential equations*, and discrete-event systems defined by *automata*. Continuous models are used extensively in the physical sciences while discrete ones are used for the abstract modeling of software, digital hardware, manufacturing systems etc. Hybrid systems research explores models that combine such discrete and continuous dynamics, and attempts to extend specific analysis methods developed for each type of systems toward methods that can analyze the behavior of a complete system, having both types of dynamics.

One promising approach that has emerged from hybrid systems research consists of a combination of ideas from algorithmic verification of discrete systems (*model checking*) and numerical simulation of continuous systems. This approach, which uses graph algorithms, numerical analysis and computational geometry, allows to compute (an approximation of) the set of *all trajectories* of the system,

starting from *all possible initial conditions*, and under *all admissible disturbances and variations in parameter values*. A successful analysis according to this method can replace infinitely many individual simulations and give additional insight on the properties of the system in question. One can view this approach as a compromise between clean analytical methods that give strong results but apply mostly to idealized and isolated subsystems, and simulation-based methods that can be applied, in principle, to arbitrary classes of systems, but their results cannot guarantee absolute confidence. Like any other proposed solution to the analysis and optimization of complex systems, this *reachability computation* suffers from the *curse of dimensionality*, and the analysis of systems with more than a few continuous variables is considered very hard. In this paper, we present recent progress in the reachability analysis of continuous and hybrid systems. Several results of our group are presented for different classes of systems of increasing complexity.

II. HYBRID SYSTEMS AND REACHABILITY ANALYSIS

The interaction of discrete events and continuous, time-driven dynamics can be efficiently modeled by a so-called *hybrid automaton* [1]. It consists of a graph in which each vertex, also called *location* or *mode*, is associated with a set of differential equations (or inclusions) that defines the time-driven evolution of the continuous variables. A *state* consists of a location and values for all the continuous variables. The edges of the graph, also called *transitions*, allow the system to jump between locations, thus changing the dynamics, and instantaneously modify the values of continuous variables according to a *jump relation*. The jumps may only take place when the values of the variables are within a certain range, called *guard*, associated with each transition. The system may only remain in a location as long as the variable values are in a range called *invariant* associated with the location.

Because of the switching between several modes, it may not be sufficient to regard the stationary behaviors of the continuous dynamics. Thus, classical results of control theory, which seldom deal with transient behaviors of dynamical systems, are not always sufficient for analyzing the complex dynamics of hybrid systems. An algorithmic approach, based on the computation of the reachable set, has emerged from hybrid systems research.

The reachable set consists of all the states that can be visited by a trajectory of the hybrid system starting in a specified set of initial states. Reachability analysis has often been motivated by safety verification, which consists in checking whether the intersection of the reachable set with a set of *bad states* is empty. When the reachable set of a

Eugene Asarin is with the LIAFA, Université Paris 7, 2 pl. Jussieu, 75251 Paris, Cedex 5, France Eugene.Asarin@liafa.jussieu.fr

Thao Dang, Goran Frehse, Antoine Girard, Colas Le Guernic and Oded Maler are with VERIMAG, 2 avenue de Vignate, 38610 Gières, France Firstname.Lastname@imag.fr

hybrid system is not exactly computable, we try to compute an overapproximation so that if it does not intersect the set of bad states, the hybrid system is guaranteed to be safe.

There is a vast literature on reachability analysis of hybrid systems. For systems such as timed automata or linear hybrid automata, where the continuous dynamics are given by linear constraints on the derivatives of the continuous variables, an exact reachability analysis is possible using standard linear algebra and algorithmic computations on polytopes [1], [6], [13], [22]. For systems with more complex continuous dynamics, several methods compute overapproximations of the reachable sets by combining numerical integration and computational geometry. These techniques use various representations for the reachable sets such as polytopes [5], [9], [16], [18], ellipsoids [8], [25], or level sets [31]. Finally, other approaches are based on computing simple discrete or hybrid abstractions of the complex continuous dynamics and performing the reachability analysis on the approximate model [2], [4], [13], [23], [28], [30]. The above methods are concerned with explicitly deriving the set of reachable states. To simply show that forbidden states are not reachable, implicit techniques, such as barrier certificates, can be advantageous [27].

III. LINEAR HYBRID AUTOMATA

A simple, yet surprisingly powerful, class of hybrid systems are those with *piecewise constant derivatives* (PCD) [6], also known as *linear hybrid automata* (LHA) [21]. In linear hybrid automata, the continuous dynamics are given as linear differential inclusions, i.e., conjunctions of *linear constraints*

$$a \cdot \dot{x} \leq b, \quad a \in \mathbb{Z}^n, b \in \mathbb{Z}, \quad (1)$$

over the time derivatives of the variables. All other sets (invariants, guards, jump relations, initial and final states) are defined by boolean combinations of linear constraints over the variables, and can be interpreted as collections of polyhedra. LHA readily model systems with simple dynamics such as timed protocols with drifting clocks, or production systems with tanks and buffers [3].

A. Fixed-point algorithm

We give a basic algorithm for computing the set of reachable states from a set of initial states I . Given a set of states Q , let $\text{Post}_c(Q)$ be the set of states reachable by letting time elapse starting from a state in Q . Let $\text{Post}_d(Q)$ be the set of states that result by taking a transition from a state in Q . Then the set of reachable states Reach is computed by the following fixed-point algorithm:

```

R := I; R' := ∅;
while R ≠ R' do
  R' := R;
  R := R ∪ Postc(R) ∪ Postd(R);
end while
Reach := R;

```

In the case of LHA, both Post-operators can be implemented using simple polyhedral computations [3]. If the fixed-point computation terminates, one obtains the exact reachable set of the LHA for unbounded time horizon. While termination often occurs in practice, it is not guaranteed and showing safety is undecidable for LHA [21]. The extension of the algorithm to hybrid automata with more complex dynamics is straightforward for systems for which one can compute continuous successors, or an overapproximation thereof. Several such classes are discussed later in this paper.

An advanced version of the above algorithm has been implemented in the tool PHAVer [13], which uses polyhedral computations based on the Parma Polyhedra Library [7] and exact arithmetic based on integers. PHAVer improves on earlier tools, namely HyTech [22], in that it employs unbounded number representations, only recomputes states where necessary, has a shared passed and waiting list, and optimizes the search order based on the topology of the transition graph.

B. Computing with exact arithmetic

For all except the most simple systems, the repeated application of the Post-operators leads to polyhedra of increasing complexity, which manifests itself in three ways: the size of the integer coefficients of the constraints (in the case of exact arithmetic computations), the number of constraints, and the number of polyhedra may grow rapidly and exceed practical limits. When the reachable set converges towards a polyhedron defined by non rational numbers or to a non polyhedral shape, the size of coefficients, respectively the number of constraints, grows without bounds and the reachability algorithm does not terminate.

Consequently, a key component in computing reachability is the ability to limit this complexity, i.e., to conservatively overapproximate polyhedra with smaller coefficients and less constraints. In theory, termination of the reachability algorithm is guaranteed under such limits, since only a finite number of constraints are possible. However, convergence might still be slow, and the final result may be an excessively large overapproximation. Nonetheless, in practice limiting the complexity is indispensable and its implementation in PHAVer has allowed us to analyze LHA of practical relevance that were previously intractable [13].

We briefly illustrate our technique to limit the number of bits using the polyhedron shown in Fig. 1(a), which features a constraint with 7 bits. The coefficients are rounded to 3 bits, which gives the slightly tilted constraint shown in (b). Linear programming is used to push the constraint towards the outside of the polyhedron, see (c), thus making the approximation conservative. Further rounding yields the final constraint, and the resulting polyhedron is outlined in (d).

For limiting the number of constraints, we have tried several techniques that are based on ranking the constraints according to some measure of importance, e.g., volume, slack or the angle between the constraints. In a *deconstruction* scheme, we rank the constraints and drop the least significant ones. In a *reconstruction*, a new polyhedron is built one by

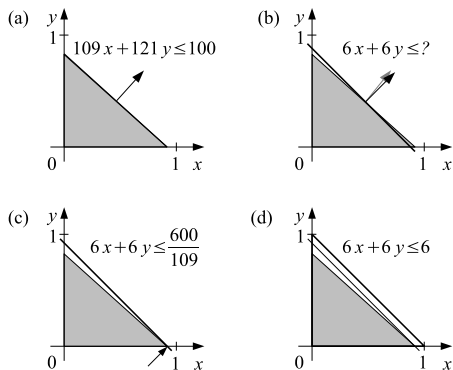


Fig. 1. Limiting the number of bits of a constraint

one with the most significant constraint of those not yet chosen. In experiments, an angle based reconstruction has shown to be orders of magnitudes faster than other methods, and acceptable in its degree of overapproximation as long as the number of constraints is chosen sufficiently high.

IV. LINEAR SYSTEMS

Hybrid systems whose continuous dynamics are given by a collection of linear differential equations is a widely used class of systems for which numerous reachability analysis techniques have been proposed [5], [8], [9], [18], [25], [26], [30]. Unlike the previous class of systems, the exact computation of the continuous successors is generally impossible and, therefore, overapproximation methods are needed. Let us consider a linear system of the form:

$$S : \dot{x} = Ax + Bu, \quad u \in U, \quad (2)$$

where U is a bounded convex set. Given a set of initial states I , and a positive time t , $\text{Reach}_{[0,t]}(S, I)$ denotes the set of states that are reachable on the interval $[0, t]$.

A. Time discretization methods

Several methods for computing $\text{Reach}_{[0,t]}(S, I)$ are based on time discretization: let $r = t/N$ be the time step, we compute a sequence of sets $\Omega_0, \dots, \Omega_N$ such that Ω_i is an overapproximation of all the states that are reachable from I within $[ir, (i+1)r]$ time.

Let us first consider an autonomous system (i.e. $\dot{x} = Ax$). The initial set of the sequence Ω_0 can be obtained by bloating the convex hull of the sets I and $e^{rA}I$ (see e.g. [5] or [9]). Then, the other elements of the sequence are computed using the recurrence relation $\Omega_{i+1} = e^{rA}\Omega_i$. The implementation of this algorithm requires to choose a representation for the sets $\Omega_0, \dots, \Omega_N$. In order to avoid additional approximations, this choice generally consists of a class of sets that is invariant under linear transformations such as polytopes [5], [9] or ellipsoids [8], [25].

A similar algorithm is possible for systems with inputs such as the one in (2). There are two main approaches to take into account the effect of inputs. The first one is based on the maximum principle [5], [25], [32], and computes at each time step and for each point of the boundary of Ω_i the input that results in the maximal successor Ω_{i+1} . The second

approach consists in forwarding the reachable set using the autonomous dynamics and then adding, using the Minkowski sum (denoted \oplus), a set that accounts for the influence of the inputs [18]. In this case, Ω_{i+1} is computed from Ω_i using a recurrence relation of the form

$$\Omega_{i+1} = e^{rA}\Omega_i \oplus V, \quad (3)$$

where the set V depends on the matrices A and B , the set of inputs U and the time step r . Both approaches guarantee the computation of an overapproximation of the reachable set $\text{Reach}_{[0,t]}(S, I)$ that converges as r tends to 0.

B. Scaling up reachability computations

In the recent years, much effort has been directed at developing scalable methods for reachability analysis [16], [18], [20], [29]. In the following, we summarize the contributions of [18] and [16] towards an efficient implementation of the recurrence relation (3).

1) *Reachability using zonotopes*: The choice of the representation of the sets $\Omega_0, \dots, \Omega_N$ is crucial for reachability computations and determines the balance between accuracy of the overapproximation and efficiency of the algorithm. For instance, the use of polytopes allows to compute an arbitrarily accurate approximation of the reachable set, but the use of the Minkowski sum in (3) will result in increasingly complex polytopes leading to intractable computations even for systems of relatively small dimension. The use of sets of bounded complexity such as ellipsoids or parallelotopes will allow an efficient implementation, but since these classes are not closed under the Minkowski sum, additional approximations will be needed at each step of the computations. The propagation of these approximations through the computations has generally a dramatic impact on the global approximation error of the reachable sets and is known as the *wrapping effect* [24].

These observations lead us to propose in [18] the use of zonotopes for the representation of the sets $\Omega_0, \dots, \Omega_N$. A zonotope Z is a polytope that can be represented as the Minkowski sum of segments:

$$Z = (u, \langle v_1, \dots, v_m \rangle) = \left\{ u + \sum_{i=1}^m \alpha_i v_i \mid \alpha_i \in [-1, 1] \right\}.$$

The vector u is called the *center* and the vectors v_1, \dots, v_m the *generators* of the zonotope. Zonotopes are closed under linear transformation and Minkowski sum. Moreover, the computation of these operations is extremely simple:

$$\begin{aligned} \Phi Z &= (\Phi u, \langle \Phi v_1, \dots, \Phi v_m \rangle), \\ Z \oplus Z' &= (u + u', \langle v_1, \dots, v_m, v'_1, \dots, v'_m \rangle). \end{aligned}$$

Consequently, an implementation of the recurrence (3) using zonotopes is very efficient even for high dimensional systems. However, let us remark that the number of generators of Ω_i increases linearly with the number of iterations. Hence, the linear transformation applied to Ω_i becomes more expensive at each step and the computations may become intractable for a large time horizon N . A solution

to this problem consists in adding a reduction operation at each step: $e^{rA}\Omega_i \oplus V$ is overapproximated by a zonotope with a predetermined number of generators m . This results in faster computations, but the wrapping effect inevitably appears for long time horizons. The parameter m allows to adjust the balance between accuracy and efficiency. This method, though presented for time-invariant systems, extends straightforwardly to time-varying systems.

2) *Efficient implementation for LTI systems:* When considering specifically linear time-invariant systems, an efficient implementation is possible based on the following observation [16]:

$$\Omega_{i+1} = e^{(i+1)rA}\Omega_0 \oplus e^{irA}V \oplus \left(e^{(i-1)rA}V \dots \oplus V \right).$$

Let us define the following auxiliary sequences of sets:

$$\begin{aligned} X_0 &= \Omega_0, & X_{i+1} &= e^{rA}X_i, \\ V_0 &= V, & V_{i+1} &= e^{rA}V_i, \\ S_0 &= \{0\}, & S_{i+1} &= S_i \oplus V_i. \end{aligned} \quad (4)$$

Then, it is clear that $\Omega_{i+1} = X_{i+1} \oplus S_{i+1}$. X_i is an overapproximation of the states reachable by the autonomous dynamics from the initial states I within $[ir, (i+1)r]$ time. S_i is an overapproximation of the states reachable by system (2) from the initial state 0 within $[ir, (i+1)r]$ time.

The computation of $\Omega_0, \dots, \Omega_N$ can be implemented very efficiently using the scheme given by (4) and zonotope representations. Indeed, the number of generators of the zonotopes to which the linear transformations are applied does not grow. Thus, the cost of an iteration is constant. This allows fast computations even for large time horizon.

However, the fact that for large i , Ω_i is a zonotope with a large number of generators can be problematic for operations other than linear transformations and Minkowski sum. For example, intersecting a zonotope with another set, which is needed for reachability analysis of hybrid systems, is intractable in high dimension. This can be handled by overapproximating the zonotopes. A variant of the previous algorithm is presented in [16] for the computation of the interval hull of $\Omega_0, \dots, \Omega_N$. The overapproximations do not propagate through the computation and therefore it is not subject to the wrapping effect. The projection of the overapproximation of the reachable set of a five dimensional linear system, computed with this method, is presented in Fig. 2. In practice, our algorithm can handle systems with about a hundred variables in a few seconds and using few MBytes [16].

V. ANALYSIS OF NONLINEAR SYSTEMS USING HYBRIDIZATION

When dealing with dynamics defined by nonlinear differential equations, the computation of the reachable set becomes much harder. Methods based on time discretization (see e.g. [11]) generally compute overapproximations that are often very large compared to the actual reachable set. To compute a more accurate approximation, we can split the state-space into small disjoint regions, compute a *simple* piecewise approximation of the system on the partition of

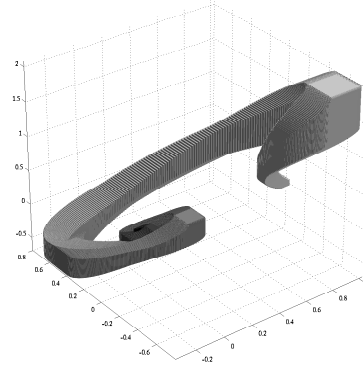


Fig. 2. Projection of an overapproximation of the reachable set $\text{Reach}_{[0,1]}(S, I)$ of a five dimensional linear system. The overapproximation consists of the union of interval hulls.

the state-space and perform the reachability analysis on the approximate hybrid model [4], [13], [23]. This approach is referred to as *hybridization*.

A. Approximation of reachable sets by hybridization

Let us consider a nonlinear system of the form:

$$S : \dot{x} = f(x), \quad (5)$$

where f is L -Lipschitz. The construction of an approximate system by hybridization consists of two steps. First, the state-space of the system is discretized into disjoint regions $(\mathcal{X}_k)_{k \in \mathcal{K}}$. We denote by Δ the diameter of the largest region. Then, in each element \mathcal{X}_k of the partition, the dynamics given by (5) is approximated by a differential equation of the form

$$\dot{x} = g_k(x) + u, \quad u \in U_k, \quad (6)$$

where the set of inputs U_k is such that $f(x) - g_k(x) \in U_k$ for all $x \in \mathcal{X}_k$. The piecewise system defined by this procedure defines H_Δ , our hybrid abstraction of S . Typically, the dynamics given by (6) is chosen much simpler than the one of S , e.g. , constant [13], [23] or linear [4]. This allows one to use methods for LHA or linear dynamics, such as those presented in the previous sections, for the reachability analysis of H_Δ . Note that any trajectory $x(t)$ of S is also a trajectory of H_Δ , since one may always choose the input $u(t) = f(x(t)) - g_k(x(t))$. This property insures that the reachable set of S is included in the one of H_Δ . The hybridization approach allows in principle to compute arbitrarily close overapproximations of the reachable set of a nonlinear system. Let $\mathcal{N}(\cdot, \delta)$ denote the δ -neighborhood and

$$\varepsilon(S, H_\Delta) = \sup_{k \in \mathcal{K}} \left(\sup_{x \in \mathcal{X}_k, u \in U_k} \|f(x) - g_k(x) - u\| \right),$$

then we can prove that [4]

$$\text{Reach}_{[0,t]}(H_\Delta, I) \subseteq \mathcal{N} \left(\text{Reach}_{[0,t]}(S, I), \varepsilon(S, H_\Delta) \frac{e^{Lt} - 1}{L} \right).$$

Since f is Lipschitz, $\varepsilon(S, H_\Delta)$ can be made arbitrarily small by using fine enough partitions of the state space, and the

reachable set of H_Δ arbitrarily close to that of S . In [4], we presented a method based on hybrid abstraction with linear continuous dynamics defined by the piecewise linear interpolant of the vector field f on a simplicial mesh of the state-space. For a mesh a size Δ , we showed that the approximation error of the reachable set was $O(\Delta^2)$ provided f is C^2 or $O(\Delta)$ otherwise. Note that in practice, the hybrid abstraction can be computed on the fly as the partition $(\mathcal{X}_k)_{k \in \mathcal{K}}$ only needs to be generated in the region of the state-space that is explored while computing the reachable set. The hybridization approach has been implemented in PHAVer to overapproximate linear systems using LHA, and was used to verify systems of up to four dimensions [13].

B. Forward/backward refinement of the partition

When only interested in verifying the reachability of a particular set of final (or unsafe) states F , even an optimal partitioning of the reachable states might result in small, costly partitions in irrelevant parts of the state space, i.e., where trajectories never even get close to the bad states. In [14], [15], similar to some approaches for discrete systems [19], we proposed an improved partitioning algorithm based on forward/backward refinement (f/b-refinement) that allows us to analyze systems of much higher complexity. A similar algorithm was proposed independently in [12].

Before we can introduce the algorithm, we need two operators: The *reverse* of a hybrid automaton H is the automaton H^{-1} obtained by reversing the transitions and jump relations and reversing the sign of the differential equations defining the continuous dynamics. The *restriction* of H to a set of states R is the automaton $H|_R$ obtained by intersecting invariants with R . The operators do not affect whether the bad states are reachable in the following sense: H reaches F from a set of initial states I if and only if H^{-1} reaches I from F . Given that $Reach(H, I) \subseteq R$, $H|_R$ is safe (i.e. does not reach F) if and only if H is safe. The refinement procedure is described by the following simple algorithm. The parameters Δ_{min} and Δ_{max} represent the minimum and maximum size of the partition of the state-space:

- 1) Initialize $\Delta = \Delta_{max}$, $R = \mathbb{R}^n$.
- 2) Compute $R = Reach(H_\Delta|_R, I \cap R)$. If $R \cap F = \emptyset$ return *safe*; else if $\Delta > \Delta_{min}$, decrease Δ and go to 3; otherwise return *inconclusive*.
- 3) Compute $R = Reach(H_\Delta^{-1}|_R, F \cap R)$. If $R \cap I = \emptyset$ return *safe*; else if $\Delta > \Delta_{min}$, decrease Δ and go to 2; otherwise return *inconclusive*.

Figure 3 illustrates the procedure. In (a), a forward reachability with a coarse partition yields a subset of bad, or “final”, states as reachable. Using the reversed automaton, reachability with a finer partition results in an even smaller intersection with the initial states. In another forward iteration, the reachable states do not intersect with the final states and we can conclude that H is safe.

F/b-refinement, implemented in PHAVer, was successful in verifying a voltage controlled oscillator circuit (VCO) that proved intractable with simple forward reachability [15].

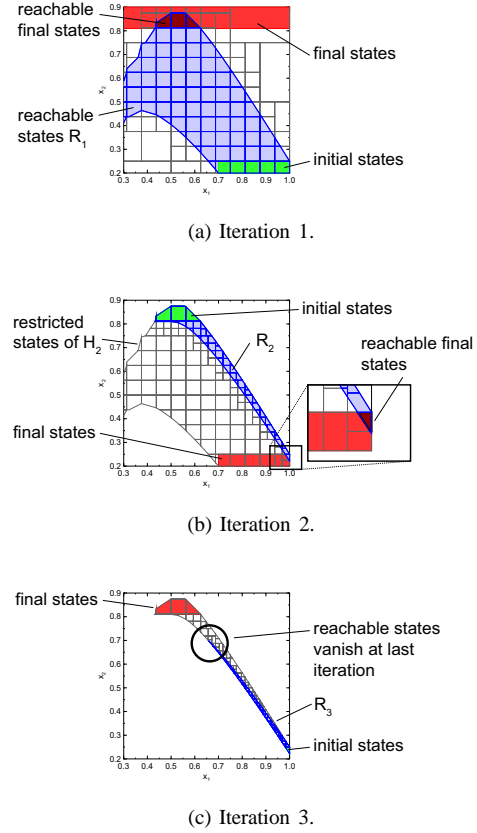


Fig. 3. Illustration of f/b-refinement

The verification goal is to show the existence of a cyclic invariant, i.e. to prove that a set of states oscillates. The proof strategy is to cut the cycle with a Poincaré plane, through which all trajectories in the vicinity of the cycle must pass exactly once. If there is a set of states on this plane whose trajectories land back inside the set after one cycle, the set, and all the reachable states from the set, is a cyclic invariant. The model of the VCO has 3 state variables, and is difficult to analyze because the system is only marginally stable in certain parts of the cycle. Figure 4 shows the forward reachable states in yellow, and the parts that land outside the initial states are clearly visible. The unsuccessful forward computation shown takes about 0.5h and 1GB RAM on a 2.8GHz Xeon running 32-bit Linux. The problem is solved using f/b-refinement by checking that the complement of the initial states on the Poincaré plane is not reachable. If this is the case, all trajectories passing through the plane must pass through the initial states. F/b-refinement with PHAVer terminates successfully after 11.5h and 1.7GB. Extrapolating from the partition size necessary to show invariance, a successful forward computation would have taken at least twice as long, and consumed about 30 times more memory. The final guaranteed limit cycle, also computed using f/b-refinement, is shown in green in Fig. 4.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented several methods for the reachability analysis of continuous and hybrid systems. The

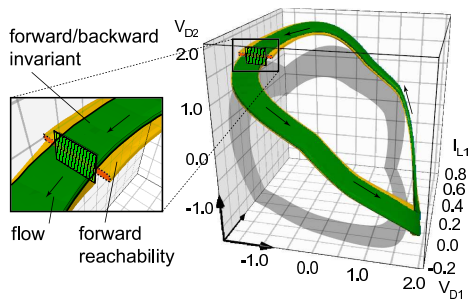


Fig. 4. Invariant for a VCO circuit from f/b-refinement versus forward reachability

complexity of reachability computations is confronted by using several types of overapproximation, and with set representations that are particularly amenable to the mathematical operators used. For linear hybrid automata, which have piecewise constant bounds on the derivatives, and for systems with linear dynamics, this has allowed us to verify systems of practical relevance and of a complexity previously beyond reach. Using hybridization approaches, we can apply these methods to continuous and hybrid systems of arbitrary dynamics. Improvements in the search mechanism such as forward/backward-refinement help us to keep the complexity of such hybridizations as low as possible.

Future work includes the development of a software tool for the reachability analysis of large-scale systems. The development of new reachability computation methods for more complex yet tractable continuous dynamics is also important. This allows us to, on one hand, enlarge the classes of hybrid systems that we can analyze directly, and on the other hand to approximate nonlinear hybrid systems using the hybridization approach more efficiently.

Along these lines, we have developed a method for systems with polynomial continuous dynamics, using Bézier techniques from computer-aided geometric design to represent the reachable sets [10]. This also suggests that other geometric modelling tools can be exploited in the algorithmic analysis of hybrid systems.

Another promising approach for complex continuous dynamics is the simulation-based reachability analysis [17]. It is based on so-called bisimulation metrics, which define topologies on the set of trajectories of a system. Then, provided we can compute a bisimulation metric for the system, it is possible to cover its reachable set with the neighborhoods of a finite number of its trajectories. This enables sound reachability analysis from a finite number of simulations of the system.

REFERENCES

- [1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995.
- [2] R. Alur, T. Dang, and F. Ivancic, "Reachability analysis of hybrid systems via predicate abstraction," in *HSCC'02*, vol. 2289 in LNCS. Springer, 2002, pp. 35–48.
- [3] R. Alur, T. A. Henzinger, and P.-H. Ho, "Automatic symbolic verification of embedded systems," *IEEE Trans. Soft. Eng.*, vol. 22, pp. 181–201, 1996.

- [4] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation." in *HSCC'03*, vol. 2623 in LNCS. Springer, 2003, pp. 20–35.
- [5] E. Asarin, T. Dang, O. Maler, and O. Bournez, "Approximate reachability analysis of piecewise-linear dynamical systems." in *HSCC'00*, vol. 1790 in LNCS. Springer, 2000, pp. 20–31.
- [6] E. Asarin, O. Maler, and A. Pnueli, "Reachability analysis of dynamical systems having piecewise constant derivatives," *Theoretical Computer Science*, vol. 138, no. 1, pp. 35–65, 1995.
- [7] R. Bagnara, E. Ricci, E. Zaffanella, and P. Hill, "Possibly not closed convex polyhedra and the Parma Polyhedra Library," in *Int. Symp. Static Analysis*, vol. 2477 in LNCS. Springer, 2002, pp. 213–229.
- [8] O. Botchkarev and S. Tripakis, "Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations." in *HSCC'00*, vol. 1790 in LNCS. Springer, 2000, pp. 73–88.
- [9] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid systems verification." *IEEE Trans. Aut. Cont.*, vol. 48, pp. 64–75, 2003.
- [10] T. Dang, "Approximate reachability computation for polynomial systems." in *HSCC'06*, vol. 3927 in LNCS. Springer, 2006, pp. 138–152.
- [11] T. Dang and O. Maler, "Reachability analysis via face lifting." in *HSCC'98*, vol. 1386 in LNCS. Springer, 1998, pp. 96–109.
- [12] L. Doyen, T. A. Henzinger, and J.-F. Raskin, "Automatic rectangular refinement of affine hybrid systems," in *FORMATS'05*, vol. 3829 in LNCS. Springer, 2005, pp. 144–161.
- [13] G. Frehse, "PHAVer: Algorithmic verification of hybrid systems past HyTech." in *HSCC'05*, vol. 3414 in LNCS. Springer, 2005, pp. 258–273.
- [14] G. Frehse, B. H. Krogh, and R. A. Rutenbar, "Verification of hybrid systems using iterative refinement," in *Proc. SRC TECHCON 2005, Portland, USA, Oct. 24-26, 2005*, 2005.
- [15] —, "Verifying analog oscillator circuits using forward/backward refinement," in *Proc. DATE'06*, 2006.
- [16] A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs." in *HSCC'06*, vol. 3927 in LNCS. Springer, 2006, pp. 257–271.
- [17] A. Girard and G. J. Pappas, "Verification using simulation." in *HSCC'06*, vol. 3927 in LNCS. Springer, 2006, pp. 272–286.
- [18] A. Girard, "Reachability of uncertain linear systems using zonotopes." in *HSCC'05*, vol. 3414 in LNCS. Springer, 2005, pp. 291–305.
- [19] S. G. Govindaraju and D. L. Dill, "Verification by approximate forward and backward reachability," in *ICCAD*, 1998, pp. 366–370.
- [20] Z. Han and B. Krogh, "Reachability analysis of large-scale affine systems using low dimensional polytopes." in *HSCC'06*, vol. 3927 in LNCS. Springer, 2006, pp. 287–301.
- [21] T. A. Henzinger, "The theory of hybrid automata," in *Proc. IEEE Symp. Logic in Computer Science*. IEEE Computer Society Press, 1996, pp. 278–292.
- [22] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HYTECH: A model checker for hybrid systems." *STTT*, vol. 1, no. 1-2, pp. 110–122, 1997.
- [23] —, "Algorithmic analysis of nonlinear hybrid systems." *IEEE Trans. Automatic Control*, vol. 43, pp. 540–554, 1998.
- [24] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect." *Computing*, vol. 61, pp. 47–68, 1998.
- [25] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis." in *HSCC'00*, vol. 1790 in LNCS. Springer, 2000, pp. 202–214.
- [26] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic reachability computation for families of linear vector fields." *J. Symb. Comput.*, vol. 32, no. 3, pp. 231–253, 2001.
- [27] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates." in *HSCC'04*, vol. 2993, 2004, pp. 477–492.
- [28] S. Ratschan and Z. She, "Safety verification of hybrid systems by constraint propagation based abstraction refinement," in *HSCC'05*, vol. 3414 in LNCS. Springer, 2005, pp. 573–589.
- [29] O. Stursberg and B. H. Krogh, "Efficient representation and computation of reachable sets for hybrid systems." in *HSCC'03*, vol. 2623 in LNCS. Springer, 2003, pp. 482–497.
- [30] A. Tiwari, "Approximate reachability for linear systems." in *HSCC'03*, vol. 2623 in LNCS. Springer, 2003, pp. 514–525.
- [31] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification and control of hybrid systems." *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.
- [32] P. Varaiya, "Reach set computation using optimal control." in *KIT Workshop*, 1998, pp. 377–383.