Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Safety verification and reachability analysis for hybrid systems

H. Guéguen * **M.A. Lefebvre** * **O. Nasri** * **J. Zaytoon** **

* *SUPELEC / IETR, Rennes, France (herve.gueguen@supelec.fr).*
** *CRESTIC, Reims, France (janan.zaytoon@univ-reims.fr)*

Abstract: Safety verification and reachability analysis for hybrid systems is a very active research domain. Many approaches that seem quite different, have been proposed to solve this complex problem. This paper presents an overview of various approaches for autonomous, continuous time hybrid systems and present them with respect to basic problems related to verification.

Keywords: Hybrid systems; safety; reachability; verification

1. INTRODUCTION

Formal verification of properties is a very important area of analysis of hybrid dynamical systems. It is, indeed, essential to use methods and tools to guaranty that the global behaviour of the system is correct and consistent with the specifications. This is especially true for safety properties that insure that the system is not dangerous for itself or its environment as long as the assumptions on which the model is built are fulfilled.

It has been early proved that the problems of safety verification and reachability analysis of hybrid systems are, in general, not decidable (Alur et al., 1995; Lafferriere et al., 1999), however algorithms and technics have improved and been applied to more and more complex systems in various fields (Penna et al., 2003; Mitra et al., 2003; Belta et al., 2004).

As it has been shown, for example, by Guéguen and Zaytoon (2004), safety properties are very important for hybrid systems and their verification may be performed with a reachability computation in the hybrid state space. Basic ideas have not really evolved since the first works, however new techniques have been proposed and algorithms have improved. The aim of this paper is then to present the problems of safety properties verification and reachability computation for continuous time autonomous hybrid systems with deterministic continuous dynamics in each mode and to propose a classification of recent improvements. As it can be seen below, to overcome the difficulties in verification and reachability analysis it is necessary to make choices regarding mathematical representations of regions of the continuous state space, general principles and algorithms. These choices depend on each other and must be consistent, however all approaches are based on common considerations that have been used to structure this paper.

In section 2 the problems of verification and the principles of two main categories of approaches are presented. As it will be seen, continuous reachability computation is a central point in these two categories. A first level of answer is to decide whether a continuous region is reachable from another one without explicitly computing the reachable space. Approaches that offer this type of answer will be presented in section 3. Approaches presented in section 4 give a second level of answer as they compute an over-approximation of the reachable space that generally allows to check safety properties (Guéguen and Zaytoon, 2004).

2. VERIFICATION AND REACHABILITY

Most of the work about verification stems from research on discrete event systems. For these systems efficient approaches have been proposed (Schnoebelen et al., 2004) and to extend them to hybrid systems it is necessary to take into account infinite sets of states that continuously evolve. In order to achieve this, two main approaches may be used. The first one is based on an abstraction of the continuous behavior by a discrete event system that can be checked using efficient tools (Penna et al., 2003). Section 2.2, shows that the difficulty is then to build this discrete event system that has the feature that, if a property is proved for the discrete event system, it is also true for the original system. To build the discrete event system, it is necessary to construct the discrete transition and this is based on determining whether a continuous region is reachable from another one by continuous dynamics. The second approach consists in adapting and extending discrete event methods to continuous dynamics. As it will be seen in section 2.3, verification is then also based on the study of reachability of regions of hybrid state space.

Reachability and related conditions computation is also the central point of the design of controller for safety of systems with discrete inputs (Cassez et al., 2002) or continuous inputs (Tomlin et al., 2003). Results within this framework may be useful to research about verification, however controller design, especially with continuous inputs is based on complex methods, for example to solve Hamilton-Jacobi equation, that will not be considered in this paper.

2.1 Hybrid automata

In order to explain the various methods we will use the following hybrid automata formalism (Alur et al., 1995) to model the systems.

A hybrid automaton, without continuous inputs and synchronising events is specified by the tuple $< L, X, Inv, F, A >$ where:

- L is the set of locations or discrete states,
- X is the continuous state space,
- Inv is a function that maps each location to a region of X: $Inv(l_i)$ or, what will be considered as equivalent, is a set of predicates on state variables that characterizes this region,
- F associates to each location l_i in L, the continuous dynamics,
- A is a set of tuple $< l, guard, Jump, l' >$ where
  - l and l' are locations
  - $guard$ is a region in X or a set of predicates on state variables
  - $Jump$ is a map from the state space to itself.

The behavior of such an automaton is specified by the set $\Theta((l_i, \mathbf{x}_i))$ of admissible trajectories τ from an initial state $(l_i, \mathbf{x}_i)$. Each of these trajectories is a finite or infinite ordered set:

$$\tau = \{(l_{\tau_0}, 0, \Phi_0), (l_{\tau_1}, t_1, \Phi_1), \cdots, (l_{\tau_k}, t_k, \Phi_k), \cdots\}$$

such as

- $l_{\tau_0} = l_i$ and $\Phi_0(0) = \mathbf{x}_i$,
- for all indices k, $\forall t \in [t_k, t_{k+1}[$
  - the continuous state is $\mathbf{x}(t) = \Phi_k(t)$
  - the continuous dynamics is specified by the activity of the location: $\dot{\Phi}_k(t) \in F(l_{\tau_k}, \mathbf{x}(t))$
  - continuous state remains within the invariant of the location : $\Phi_k(t) \in Inv(l_{\tau_k})$
- for all indices k, there exists a transition $< l, guard, Jump, l' >$ in A such as
  - $l = l_{\tau_k}$, $l' = l_{\tau_{k+1}}$
  - the state before the firing is within its guard: $\Phi_k(t_{k+1}) \in guard$
  - the initial state in the next location is the result of applying the jump function to the state before the firing: $\Phi_{k+1}(t_{k+1}) = Jump(\Phi_k(t_{k+1}))$.

For a trajectory each term $(l_{\tau_k}, t_k, \Phi_k)$ is then associated to a time interval $[t_k, t_{k+1}]$ when the location does not change and the continuous state evolves according to the continuous dynamics associated to this active location l_{τ_k}, thus defining a continuous transition. Each time t_k is a discrete transition firing time when the location changes. A trajectory is then a sequence of discrete and continuous transitions.

For a trajectory τ of a hybrid automaton, the state at time t will be denoted $\tau(t)$ and is defined by $\tau(t) = (l_{\tau_k}, \Phi_k(t))$ where k is defined by $t \in [t_k, t_{k+1}[$.

These definitions of a hybrid automaton and of the set of its trajectories from a state, leads to the following definitions of the successor and predecessor sets of a point of the hybrid state $(l_i, \mathbf{x}_i)$.

The discrete successor set of the point $(l_i, \mathbf{x}_i)$ is the set of points reachable by the firing of a discrete transition:

$$Succ_D((l_i, \mathbf{x}_i)) = \{(l_k, \mathbf{x}) | \exists (l_i, guard, Jump, l_k) \in A \wedge (\mathbf{x}_i \in guard) \wedge (\mathbf{x} = Jump(\mathbf{x}_i)\}$$

Symmetrically the discrete predecessor set of the point $(l_i, \mathbf{x}_i)$ is the set of points from which it is possible to reach this point by a discrete transition:

$$Pred_D((l_i, \mathbf{x}_i)) = \{(l_k, \mathbf{x}) | \exists (l_k, guard, Jump, l_i) \in A \wedge (\mathbf{x} \in guard) \wedge (\mathbf{x}_i = Jump(\mathbf{x})\}$$

The continuous successor and predecessor sets of a point are defined in relation with the the continuous transitions:

$$\begin{aligned} Succ_C((l_i, \mathbf{x}_i)) &= \{(l_i, \mathbf{x}) | \exists \tau \in \Theta((l_i, \mathbf{x}_i)),\ \exists t \in [0, t_1], \\ &\qquad \mathbf{x} = \Phi_0(t)\} \\ Pred_C((l_i, \mathbf{x}_i)) &= \{(l_i, \mathbf{x}) | (l_i, \mathbf{x}_i) \in Succ_C((l_i, \mathbf{x}))\} \end{aligned} \tag{1}$$

Finally, hybrid successor and predecessor sets of a point are related to hybrid trajectories:

$$\begin{aligned} Succ_H((l_i, \mathbf{x}_i)) &= \{(l_k, \mathbf{x}) | \exists \tau \in \Theta((l_i, \mathbf{x}_i)), \\ &\qquad \exists t,\ \tau(t) = (l_k, \mathbf{x})\} \\ Pred_H((l_i, \mathbf{x}_i)) &= \{(l_k, \mathbf{x}) | (l_i, \mathbf{x}_i) \in Succ_H((l_k, \mathbf{x}))\} \end{aligned}$$

These definitions are easily extended to regions of the hybrid state space considering that the image of the region is the union of the images of its points. For example, the continuous and discrete successor set of a region are given by:

$$Succ_C(R) = \bigcup_{(l_i, \mathbf{x}) \in R} Succ_C((l_i, \mathbf{x})) \tag{2}$$

$$Succ_D(R) = \bigcup_{(l_i, \mathbf{x}) \in R} Succ_D((l_i, \mathbf{x})) \tag{3}$$

2.2 Discrete event abstraction based verification

The general principle of these approaches is to build a discrete event model equivalent to the hybrid system such that verification of the property for the discrete model guaranties the property for the hybrid system. Equivalence is considered here as bisimulation, that is the possibility to define a map from the hybrid state space to the discrete state space such that the map associates to each trajectory of one model a trajectory of the other one (Chutinan and Krogh, 2001).

The first step then consists in defining the regions of the hybrid state space that are worth being considered to build the discrete model. It is generally not necessary to consider all the state space but some specific areas such as the guards of the transitions, the invariants of the locations or regions linked to the property, or sometimes the borders of these regions. Each of these first regions is associated with a discrete state. Then the discrete transitions are built and the regions are split in an iterative way according to reachability considerations (Tabuada et al., 2002).

At the beginning of each iteration, a set of hybrid areas $\{(l_i, P_i)\}$, (where l_i is a location and P_i a region of the continuous state space) is given and each hybrid area is associated with a discrete state q_i. In the discrete model there exists a path of transitions from q_i to q_j if there exists a hybrid trajectory from one point of (l_i, P_i) to one point of (l_j, P_j). The first step then consists in computing for each pair (q_i, q_j), such that there exists a transition from q_i to q_j, the intersection of P_i with the restriction to l_i of the set hybrid predecessors of (l_j, P_j) [1].

Two possibilities can then arise:

- $Pred_H(l_j, P_j)_{|l_i} \cap P_i \neq P_i$: then from some points of (l_i, P_i) it is possible to reach (l_j, P_j) and for others this is not possible. The region (l_i, P_i) is then deleted from the set of areas and two news areas are added [2] $(l_i, P_i \cap Pred_H(l_j, P_j)_{|l_i})$ and $(l_i, P_i - Pred_H(l_j, P_j)_{|l_i})$. Then, from all points of the first area it is possible to reach (l_j, P_j) and a discrete transition is created between the associated discrete states. On the contrary it is not possible to reach (l_j, P_j) from any point of the second area and there is no transition between the associated discrete states.
- $Pred_H(l_j, P_j)_{|l_i} \cap P_i = P_i$: then, from all points of (l_i, P_i) it is possible to reach (l_j, P_j) and nothing is changed.

The iterative building of the discrete model stops when there is no more change in the set of hybrid areas in two consecutive iterations. The discrete model is then a bisimulation of the hybrid system that can be used for verification.

In order to refine the model, it is actually not useful to consider the hybrid predecessor set $Pred_H(l_j, P_j)$ that is complex to compute and introduces a lot of redundancy in the transition structure of the discrete model. It is then more relevant to consider more local and simple predecessor set if they are consistent with the choice of regions. For example, if the guard of the transitions are used to define the regions at the first iteration, it is possible to use the predecessor set with one continuous and one discrete transition (i.e. $\widehat{Pred}_H(l_j, P_j) = Pred_D(Pred_C(l_j, P_j))$), if the borders of the guard are used, it is necessary to consider $\widetilde{Pred}_H(l_j, P_j) = Pred_C(Pred_D(Pred_C(l_j, P_j)))$.

One difficulty of this approach is of course to compute the hybrid predecessor set of an area but the main difficulty is linked to the iterative algorithm because it is impossible to guaranty its convergence. To avoid this difficulty, the iterative decomposition of the region is generally stopped at some step. The resulting discrete model is then an abstraction of the hybrid system, i.e. each trajectory of the hybrid systems is mapped to a trajectory of the discrete model but some trajectories of the discrete model are not related to any trajectory of the hybrid system. It is not possible to check all properties with this abstract model but it may be used for safety properties. If it is possible to show that for all trajectories of the discrete model the property holds, then the property is checked for the hybrid system. If not, in order to conclude, it is necessary to determine whether the trajectories that violate the property are related to any trajectory of the hybrid system or not.

In order to answer this question it is necessary to refine the abstraction by splitting some regions. Some heuristics may be used to guide the choice of the regions that are considered when refining the model. It is then possible to begin with the closest regions of the forbidden area (Alur et al., 2003), but it is also possible to use the result of the verification and especially the counter-example trajectory given by the checking tool, to guide the refinement. It is then performed in the vicinity of this trajectory (Fehnker et al., 2005). It may be noticed that it is sometime possible to refute a trajectory with other considerations such as tranversality of flows and guards (Stursberg et al., 2004) that are simpler to compute.

This approach that builds a discrete event model of a continuous or hybrid system may be found in various propositions (Tiwari and Khanna, 2004; Alur et al., 2002, 2003; Ratschan and She, 2005; Blouin et al., 2003; Kloetzer and Belta, 2006) according to the assumptions that are made about the system (guards, invariants, continuous dynamics, ...). However the basic problem of these approaches is to compute the predecessor sets and mainly the continuous ones [3].

2.3 Hybrid reachability based verification

A second family of approaches allows to check reachability properties of hybrid systems. This restriction to reachability properties may seem to be an important one but, as the state space of hybrid systems implicitly includes time, a lot of useful properties, especially safety properties, may be expressed as reachability properties (Guéguen and Zaytoon, 2004). The question is then to determine whether it is possible to reach a given area of the hybrid space $R_c = \bigcup_{k_c}(l_{k_c}, P_{k_c})$ from an initial region $R_0 = \bigcup_{i0}(l_{i0}, P_{i0})$ or not. The answer is given by considering the sets $Succ_H(R_0)$ and R_c, or the sets R_0 and $Pred_H(R_c)$.

According to the definition of trajectories of hybrid automata in section 2.1 the hybrid successor set of the area R_0 is given by the limit of the series of areas, starting at R_0, defined by:

$$R_1 = Succ_C(R_0)$$
$$R_i = R_{i-1} \cup Succ_C(Succ_D(R_{i-1}))$$

There is no guaranty that this limit exists especially when some invariants are not bounded. In all cases, even when one is sure that the limit exists, it may be difficult to compute it because, it is necessary first to find an algorithm that converges on the limit and secondly to compute the successor sets and especially the continuous ones. As this approach is used to check safety properties, the computation of the reachable set $Succ_H(R_0)$ is used to establish whether this set has an empty intersection with

[1] The restriction $B_{|l_i}$ of the set B of hybrid areas to the location l_i is the union of the continuous regions associated to l_i in B: $B_{|l_i} = \cup_{(l_i, P_k) \in B} P_k$

[2] It is some time usefull to add more than two areas in order to get convex regions.

[3] it may be noticed that it is also possible to consider the successor sets.

R_c or sometimes whether it is included in R_c. It is then possible to use over-approximations of the reachable set that are generally easier to compute and may lead to a positive conclusion.

This reachability verification is used in tools such as HyTech (Henzinger et al., 1997), PhaVer (Frehse, 2005) or d/dt (Dang, 2000).

2.4 Conclusion

The computation of the hybrid successor set of an area, or of its predecessor set, is the central point of verification of hybrid systems with a discrete event approach, where local hybrid reachability considerations are used to build the discrete model, or with a hybrid approach where global reachability is directly considered. Of course, discrete successors computation leads to classical problems of combinatorial explosion but considering actual works the main limitations are related to continuous reachability.

Characterizing continuous successors may be considered according to two aspects. The first one consists in concluding whether a region is reachable from an other one, for example to build the discrete event abstraction, and does not necessarily require to explicitly compute the reachable set. The second one aims at computing this reachable set, or an over-approximation, in order to use it for hybrid reachability computation or discrete event abstraction refinement.

3. CHARACTERIZING REACHABLE SPACE

The approaches that are presented in this section aim at establishing whether it is possible to reach a given area from an other one without explicitly computing the reachable space but by characterizing it by specific properties. They are mainly used in discrete event abstraction techniques. Three families may be specified. The first one consists in defining borders that are not crossed by trajectories and that separate the initial and goal regions. The second one aims at searching partial characteristics of the intersection of the reachable and the goal regions, that are easier to compute and then to prove that they are inconsistent. Conversely the third family aims at proving that there exists trajectories from one region to the other one.

3.1 Displaying uncrossable borders

To find a border that trajectories do not cross and that is between the initial region and the goal one is obviously a mean to prove that the second region is not reachable. A first approach consists in characterizing invariant domains, i.e. regions of the continuous space that continuous trajectories never leave, and that include the initial region. If such a domain has an empty intersection with the goal region, this one is obviously not reachable. The second approach aims at explicitly computing a border that is not crossed between the regions.

Characterizing invariant space The basic idea of this approach is to use structural properties of the continuous dynamics in order to define some borders of such invariant domain. It is therefore specific of some classes of systems. So for linear systems $\dot{\mathbf{x}} = A\mathbf{x}$ it is possible, with specific assumptions, to find invariant regions with borders specified by linear constraints (Tiwari, 2003) whereas for affine systems $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}$, it is possible, with different hypotheses, to find polynomial constraints (Rodriguez-Carbonell and Tiwari, 2005).

For example, for a system defined by $\dot{\mathbf{x}} = A\mathbf{x}$, from the linear term $p = \mathbf{c}^T\mathbf{x}$ where $\mathbf{c}$ is a real eigenvector of A^T, it is possible to compute $\dot{p} = \lambda p$ where λ is the related eigenvalue and then $p = e^{\lambda t}\mathbf{c}^T\mathbf{x}_0$. The sign of the linear term p is then constant and its norm ($|p|$) increases or decreases according to the sign of λ. If $\lambda > 0$, for example, this value increases and the inequality $\mathbf{c}^T\mathbf{x} > \alpha$ with $\alpha \geq 0$ defines an invariant domain. Moreover the reachable space from the initial domain P_0 is constrained by the inequality $|\mathbf{c}^T\mathbf{x}| \geq \min_{P_0}(|\mathbf{c}^T\mathbf{x}|)$. If $\mathbf{c}^T\mathbf{x}$ is positive on P_0 for example, the constraint $\mathbf{c}^T\mathbf{x} \geq \min_{P_0}(\mathbf{c}^T\mathbf{x})$ holds on the reachable space from P_0 as shown on figure 1. Symmetrically, it is possible to show that if $\lambda < 0$, then the reachable space is specified by the constraint $|\mathbf{c}^T\mathbf{x}| \leq \max_{X_0}(|\mathbf{c}^T\mathbf{x}|)$ that can be refined according to the sign of $\mathbf{c}^T\mathbf{x}$ on P_0. When the eigenvalues are complex and conjugate ($\lambda = \alpha \pm j\beta$), with a negative real part, it is also possible to find bounds for the linear term $p = \mathbf{c}^T\mathbf{x}$ where $\mathbf{c}$ is a real linear combination of eigenvectors: $|p| \leq (d_1^2 + d_2^2)^{\frac{1}{2}}$ where d_1 and d_2 are deduced from the maximum values of the linear term on the initial domain.

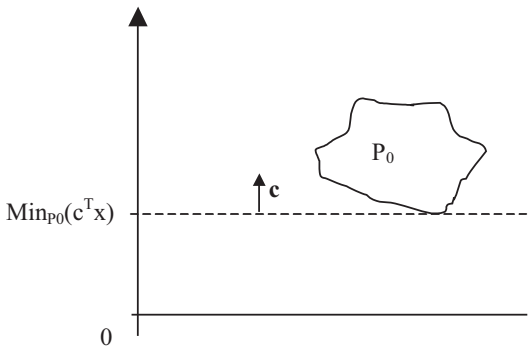

Fig. 1. Qualitative invariant

For affine systems $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}$ with rational eigenvalues, it is possible to compute algebraic invariant regions (i.e. specified with polynomial equations) that include the reachable space when the initial region is algebraic (Rodriguez-Carbonell and Tiwari, 2005): the temporal solution of the state equation can be expressed as a combination of real exponential, sine, and cosine functions depending on eigenvalues. For example, for a complex eigenvalue $\lambda = \alpha \pm j\beta$, terms $t^k e^{\alpha t}\cos\beta t$ and $t^k e^{\alpha t}\sin\beta t$ appear in the temporal response. As eigenvalues are rational it is possible to find two rational numbers p and q such as each eigenvalue may be expressed by $\lambda = a_\lambda p + j b_\lambda q$, where a_λ and b_λ are integers. Then the terms of the temporal solution may be expressed as polynomials on variables t, e^{pt}, e^{-pt}, $cos(qt)$ and $sin(qt)$. It is then possible to eliminate time from the expression of the reachable space to characterize it, making it possible to check whether these constraints that have been computed are satisfied by the goal region or not.

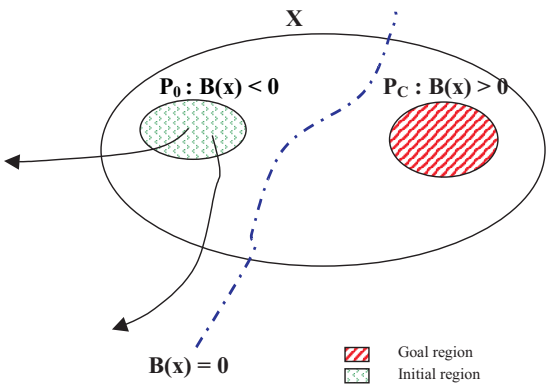

Fig. 2. Barrier certificates

Barrier certificates A second approach (Prajna and Jadbabaie, 2004; Glavaski et al., 2005) aims at explicitly searching for an uncrossable border between the initial and the goal regions. This is formalized by the computation of a map ('barrier certificate') from the state space to the set of reals whose sign is different on the set of initial and goal states under increasing or decreasing constraints. So, if it possible to find a map $B(\mathbf{x})$ that respects constraints of (4), where $f(\mathbf{x})$ is the system dynamics, it is possible to conclude that the goal region P_C is unreachable from the initial region P_0, as shown on figure 2, because the last equation imposes that it is impossible to cross the border defined by $B(\mathbf{x}) = 0$ from the region where $B(\mathbf{x}) < 0$.

$$\begin{aligned} \forall \mathbf{x} \in P_C \qquad & B(\mathbf{x}) > 0 \\ \forall \mathbf{x} \in P_0 \qquad & B(\mathbf{x}) < 0 \\ \forall \mathbf{x} \in X \qquad & B(\mathbf{x}) = 0 \Rightarrow \frac{\partial B(\mathbf{x})}{\partial x} f(\mathbf{x}) \leq 0 \end{aligned} \tag{4}$$

When considering hybrid reachability of area R_C from area R_0, it is necessary to search for a certificate $B_l(\mathbf{x})$ for each location l. The previous conditions are modified to take into account the initial, invariant or dangerous region of each location as well as the jumps and become:

$$\begin{aligned} \forall \mathbf{x} \in R_{C|l} \qquad & B_l(\mathbf{x}) > 0 \\ \forall \mathbf{x} \in R_{0|l} \qquad & B_l(\mathbf{x}) < 0 \\ \forall \mathbf{x} \in Inv(l) \qquad & B_l(\mathbf{x}) = 0 \Rightarrow \frac{\partial B_l(\mathbf{x})}{\partial \mathbf{x}} F(l, \mathbf{x}) \leq 0 \\ \forall < l, guard, \sigma, Jump, m > \in A & \\ (\mathbf{x} \in guard) \wedge (B_l(\mathbf{x}) < 0) & \Rightarrow B_m(Jump(\mathbf{x})) < 0 \end{aligned} \tag{5}$$

Finding the functions $B_l(\mathbf{x})$ is obviously difficult. When the hypothesis that the dynamics is polynomial and the initial, final, ..., sets are semi-algebraic (i.e. specified by polynomial inequalities) it is possible to search for candidate functions of the type:

$$B_l(\mathbf{x}) = b_{l,0}(\mathbf{x}) + \sum_{i=1}^{m} c_{l,i} b_{l,i}(\mathbf{x})$$

where $b_{l,i}(\mathbf{x})$ are single terms.

The set of constraints (5) can then be expressed by non-negative terms and the problem can be expressed as a sum of squares (SOS) that can be solved by semi-definite programming in the convex case. However constraints (5) do not define a convex case and it is necessary to first find a solution of the convex case defined by this set of constraints without $B_l(\mathbf{x}) = 0$ before finding less conservative solutions of the initial problem.

3.2 Constraints inconsistency

The difficulty of characterizing $Succ_C(R_0)$ the continuous reachable set from the area R_0 can sometimes be overcome by working in specific sub-spaces to find local constraints and then proving that the global set of constraints is inconsistent. Constraints can be searched on temporal or spatial aspects of the explicit solution of the state equation and this approach will be illustrated in case of linear systems.

Temporal constraints on reachability in eigensubspaces When considering a linear system specified by $\dot{\mathbf{x}} = A\mathbf{x}$, it is interesting to consider its eigenvalues and its eigen subspaces. This is particularly true when the matrix A is diagonalizable, as the solution of the differential equation is then the sum of the solutions in these subspaces that have lower dimension. The basic idea of the approach (Yazarel and Pappas, 2004) is to take advantage of this simplicity to compute temporal constraints and study their consistency.

Let us consider the problem of reachability of region P_C from region P_0 and an eigenvalue λ of A with an associated eigen subspace with dimension 1 (this case is illustrated for 2 dimensions on figure 3), it is easy to compute, for example with linear programming, the upper and lower bounds of the projections of these regions (P_0 and P_C) on the eigen subspace associated with λ, (z_0^l, z_0^u) and (z_C^l, z_C^u). In other respects, the projection of the trajectory from the point $\mathbf{x}_0$ is specified by $z(t) = z_0 e^{-\lambda t}$ where z_0 is the projection of $\mathbf{x}_0$. It is then possible to compute the minimum and maximum time necessary to go from the projection of region P_0 to the one of P_C. For example the maximum time is given by:

$$t_{max,\lambda} = \max\left(\frac{1}{\lambda}\log\left(\frac{z_C^l}{z_0^u}\right), \frac{1}{\lambda}\log\left(\frac{z_C^u}{z_0^l}\right)\right)$$

If the maximum time is negative according to this equation, it means that P_C is unreachable from P_0 as its projection is unreachable in one eigen subspace.

If this computation is performed with an other eigen subspace, it is possible to consider the two time intervals. It can then be deduced from the emptiness of their intersection that P_C is not reachable from P_0.

Equivalent considerations can be used with complex eigenvalues $\lambda = \alpha \pm i\beta$. Using polar coordinates (ρ, θ) to express the projection of the state on the eigen subspace leads to temporal solutions of the state equation $\rho(t) = e^{\alpha t}\rho_0$ and $\theta(t) = \beta t + \theta_0$. However the optimisation problem that gives the temporal bounds is, generally, neither convex nor linear. A possible solution is then to solve the problem first according to ρ and then, if needed, to refine this solution according to θ.

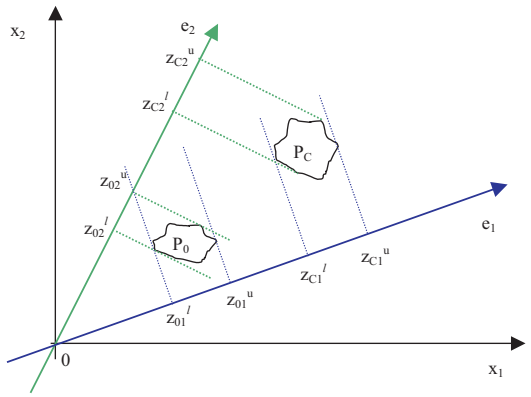

Fig. 3. Determining projection bounds

Spatial constraints on reachability in eigensubspaces Solving the state equation in eigen subspace of matrix A can also be used to compute constraints on coordinates of reachable points and goal region (Yazarel et al., 2004). The proof that there does not exist any point that fulfils all constraints can be performed by optimisation procedures that bring limitations on the type of initial and goal regions.

If the system is specified by $\dot{\mathbf{x}} = A\mathbf{x}$, where the matrix A is diagonalizable with rational eigenvalues, it is possible to work in the basis defined by the eigenvectors and to find a rational q such as each eigenvalue may be expressed by $\lambda_i = k_i q$ where k_i is an integer. Then the reachable space is characterized for each eigen subspace by the set of points such that it exists t and $z_{i,0}$ and $z_i = e^{\lambda_i t} z_{i,0}$ that can be rewritten $z_i = (e^{qt})^{k_i} z_{i,0}$.

Considering pairs of eigen subspaces, it is possible to eliminate $e^{q.t}$ and then to deduce that reachable points are characterized by $z_i^{k_j}.z_{j,0}^{k_i} - z_j^{k_i}.z_{i,0}^{k_j} = 0$.

In other respects, the polynomial $P_t(\mathbf{z}) = \sum_i \lambda_i.z_i^2$ can be checked to be increasing with time and can then be used to write the constraint $P_t(\mathbf{z}) - P_t(\mathbf{z}_0) \geq 0$ that imposes that the point $\mathbf{z}$ is reached at some positive time.

If the initial and goal regions are specified by polynomial constraints, the set of all the above constraints defines a semi-algebraic set that can be checked to be empty using a sum of square decomposition (Yazarel et al., 2004).

Equivalent consideration on reachability in eigen subspaces also results in polynomial constraints when A is nilpotent or with pure imaginary eigenvalues.

3.3 Existence of trajectories

The approaches presented above aim at proving that it is not possible to reach the goal region (P_C) from the initial one (P_0) while remaining in the region Inv. However they are very conservative and it is not possible to conclude on the property when they fail to provide a result. It is then interesting to check whether a trajectory exists from one region to the other one. The notion of reachability certificate, that is close to barrier certificate notion (Prajna and Rantzer, 2005), aims at proving that such a trajectory exists.

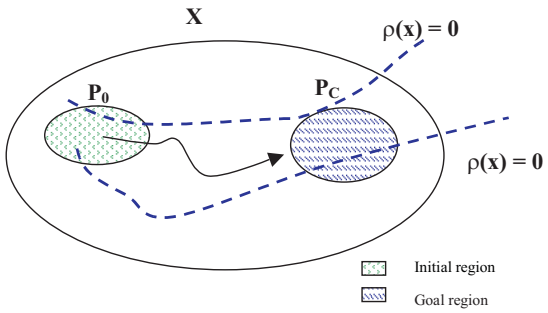

Fig. 4. Reachability Certificate

For the dynamical system specified by $\dot{\mathbf{x}} = f(\mathbf{x})$, if there exists a function ρ continuously differentiable that meets the conditions (6) then there exists a continuous trajectory from P_0 to P_C. Intuitively speaking, these conditions express that some trajectories starting in P_0 (where ρ is globally positive) and such as ρ remains positive, goes out of the complementary part of P_C in the invariant Inv (denoted $Inv - P_C$) within a finite time interval. Moreover they can go out only by crossing the border of P_C as for all other borders of Inv, ρ is negative. These trajectories are then constrained by the 0 level of function ρ, as shown on figure 4.

$$\begin{aligned} \int_{P_0} \rho(\mathbf{x})dx > 0 \\ \rho(\mathbf{x}) < 0 \qquad \forall \mathbf{x} \in cl(\partial Inv - \partial P_C) \\ div(\rho f)(\mathbf{x}) > 0 \qquad \forall \mathbf{x} \in cl(Inv - P_C) \end{aligned} \tag{6}$$

where ∂X and $cl(X)$ stands for the bound and the closure of X and $div(\rho f)$ is the divergence of the product.

When dynamics are polynomial and the regions are defined by polynomial constraints it is possible to search for a polynomial function ρ with sum of squares.

4. REACHABLE SPACE COMPUTATION

The previous section was devoted to approaches that aim at concluding whether it is possible to reach a specific region from an other one or not. This information is not always sufficient to solve the verification problem and it is then mandatory to explicitly compute the reachable space. This section presents how this reachable space can be computed or at least one over-approximation that is sufficient for safety properties (Guéguen and Zaytoon, 2004). After reminding general principles of reachability computation, specific difficult problems and propositions to solve them will be presented.

4.1 General presentation

The computation of the continuous reachable space in order to use it in an event abstraction method or a hybrid reachability computation (see section 2) consists in computing the set specified by equation (2) for a set $R = (l_i, P_0)$.

To obtain an explicit expression of this set, it is mandatory to eliminate time and the dependency to the specific point

$\mathbf{x}_0$ (see (Guéguen and Zaytoon, 2004) for more details). For some specific systems, this can be done using quantifier elimination tools (Lafferriere et al., 1999). Another approach could be based on classical integration of the differential equation. However, as a set of initial conditions is considered and the vector flow may be specified with uncertainties, the set of trajectories to simulate is infinite. Moreover, to prove safety properties it is necessary to guaranty the result of the simulation. Finally, it is sometimes possible to easily eliminate time as it can be seen for the 2 dimensions example of equation (7) illustrated on figure 5, but this approach is limited to continuous dynamics specified by linear differential inclusions. For other systems it is necessary to use more complex methods. However the exact computation of $Succ_C(l_i, P_0)$ is not mandatory for safety analysis so the methods presented below mainly aim at computing over-approximations of this set.

$$\begin{aligned} F_i &= \{\dot{\mathbf{x}} \mid (2,\ -1)\dot{\mathbf{x}} \geq 0 \ \wedge\ (1,\ -3)\dot{\mathbf{x}} \leq 0\} \\ P_0 &= \{\mathbf{x} \mid 0 \leq (1,\ 0)\mathbf{x} \leq 1 \ \wedge\ (0,\ 1)\mathbf{x} = 0\} \\ Succ(l_i, P_0) &= \{\mathbf{x} \mid (0,\ 1)\mathbf{x} \geq 0 \ \wedge (2,\ -1)\mathbf{x} \geq 0 \\ &\quad \wedge\ (1,\ -3)\mathbf{x} - 1 \leq 0\} \end{aligned} \tag{7}$$

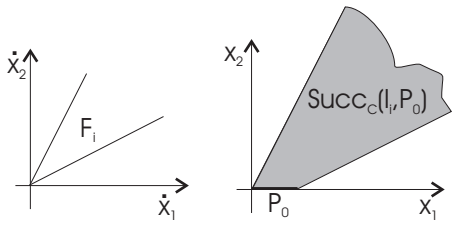

Fig. 5. Simple time elimination

A first approach to compute the reachable space is to change the continuous dynamics into a set of linear differential inclusions for which simple time elimination may be used (Henzinger et al., 1998; Frehse, 2005; Lefebvre and Guéguen, 2006). To achieve this, a partition of the invariant is specified and a linear inclusion, valid for this set of points, is associated to each element of this partition. This defines an abstraction of the continuous dynamics and the over-approximation of the reachable space is computed by a spatial iteration based on the partition (see (Guéguen and Zaytoon, 2004) for more details). Of course, the accuracy of the result depends on the choice of the regions and a sensible choice (Lefebvre and Guéguen, 2006) improves the balance of complexity and accuracy. Finally, time elimination from linear differential inclusion leads to linear borders and then favours the choice of polyhedral regions.

Most approaches are based of the consideration that it is not necessary to consider the successor set as defined by equation (2), especially if the invariant is bounded, but finite time reachability is sufficient. It is then possible to use a time sampled computation to get useful information. Beside approaches based on interval integration of ordinary diffrential equations (R.J.Lohner, 1987) most approaches are based on the same basic principle.

This one consists in choosing a time step δ and computing the series of the over-approximations P_k of the reachable space between times $k\delta$ and $(k+1)\delta$, (Chutinan and Krogh, 2003; Girard, 2005; Asarin et al., 2006; Hickey and Wittenberg, 2004). The first steps of the method are illustrated on figure 6 when polyhedral regions are chosen. The first step (figure 6.a) consists in computing X_1 image of the initial region X_0 after time δ. The second step (figure 6.b) then consists in searching for a polyhedron P_0 that includes the whole trajectory between the two times. These steps are then iterated to compute reachable space at the next times (figure 6.c). In order to reduce the approximations it is generally pertinent to use X_i and not P_{i-1} as the basis of the next step. However the second step that computes P_{i-1} from X_{i-1} and X_i is complex and time consuming, it is therefore sometimes useful to use P_{i-1} especially when this does not induce further approximations. This is the case for linear dynamics ($\dot{\mathbf{x}} = A\mathbf{x}$) as the evolution from one time step to the other one is known, constant and given by $e^{A\delta}$. From the computation of P_0 it is easy to iteratively compute the series $P_i = e^{A\delta}P_{i-1}$.

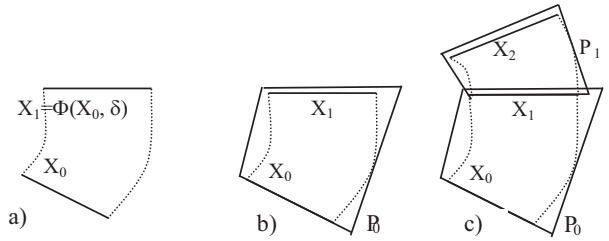

Fig. 6. Sampled approximation

This approach can also be used for systems specified by $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}$ where $\mathbf{u}$ stands for a bounded uncertainty. At each time step the approximation of the reachable space is then given (Girard, 2005) by $P_i = e^{A\delta}P_{i-1} \oplus V$ where V is a region that depends on the uncertainty and the dynamics and where $\oplus$ is the Minkowski sum [4] of the sets.

As it can be seen, computing continuous reachable space, by simple time elimination or sampled time computation, is based on rather simple considerations but it is difficult to implement. The first point is that it is necessary to make computations on regions of the state space, such as intersection, union, dynamics evolution or Minkowski sum. It is therefore mandatory to consider regions that are simple and allows efficient calculus. The complexity of the computation depends on the dimension of the state space and on the characteristics of the dynamics. Some of the approaches that are presented below then deal with the problem of sets and computation in order to cope with more complex systems and other approaches consider the problem of model transformation in order to change complex systems into systems that can be solved with existing algorithms.

4.2 Space regions

In order to choose a type of sets for continuous space regions it is necessary to consider their compacity [5] and the complexity of the computation on this type of sets. Another important point is the closure of the type of sets

[4] The Minkowski sum of 2 sets A and B, is defined by $A \oplus B = \{\ a + b \mid a \in A \ \wedge\ b \in B\ \}$

[5] The information necessary to describe a region

with respect to the operations needed for the reachability computation, because, when the result of one operation is not a set of the given type, it is necessary to compute a new set, that has the good type and that includes it. This may induce complexity and approximations. For example, if sets are ellipsoids, as the Minkowski sum of two ellipsoids is not an ellipsoid it is necessary to compute a new ellipsoid that includes the result of the sum.

If some works consider polynomial regions, (see for example Dang (2006)), classical sets that are considered in reachability computation are ellipsoids (Kurzhanski and Variya, 2000) and various types of polyhedrons. Ellipsoids are compact and closed for the transformation induced by linear dynamics, however they are not closed for other operations and this may induce important approximations. From now on we will focus on polyhedral sets.

Hyperrectangles The most simple type of polyhedrons is hyper-rectangles that are polyhedrons where all borders are normal to one of the basis vectors, that is, where the border is specified by a constraint on only one component. One main difficulty is that this type of sets is not closed for continuous dynamics changes, as it can be seen on figure 7.a where the rectangle A_1 is the image of A_0 after some time step. As A_1 is not a hyper-rectangle it must be approximated by the hyper-rectangle in dashed line. This wrapping effect is well-known in the interval computation field (see e.g. (Nedialkov et al., 1999)). One solution to overcome this effect is to express each intermediate result in an intermediate basis (figure 7.b). New results in the field of solving differential equations by guarantied interval computation bring new interest to hyper-rectangles especially when modelling uncertainties are considered.

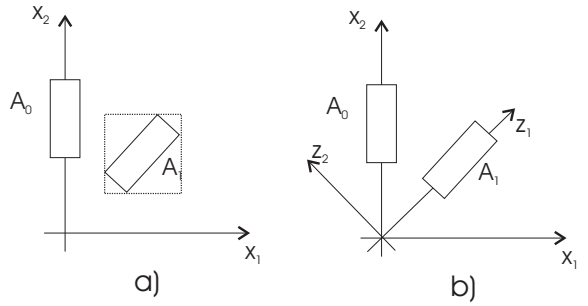

Fig. 7. Wrapping effect

Polyhedrons Using general convex polyhedrons as continuous space regions for reachability computation is not new as they naturally appear when dealing with linear differential inclusion systems. One problem that one has to face is that the iterative computation quickly leads to polyhedrons specified by linear constraints with more and more complex coefficients. For example, when rational coefficients are used in order to guaranty the result, this leads, after a few steps, to need integers with a number of bits that are higher than common languages capacity. If polyhedral libraries such as 'Parma Polyhedra Library' (Bagnara et al., 2002) allows such very long coding of integers it is useful to be able to simplify the coefficients of the constraints while guarantying that this simplification specifies an over-set.

Such an approach of constraints simplification is proposed by Frehse (2005) and illustrated on figure 8. From a linear constraint ($c^T\mathbf{x} < b$) expressed with a given number of bits (for example 7 bits on figure 8.a), it is possible to compute integer coefficients ($\tilde{c}$ and $\tilde{b}$) coded with a lower number of bits (for example 3 for $\tilde{c}$ on figure 8.b) such that the new constraint $\tilde{c}^T\mathbf{x} < \tilde{b}$ (computation of the new constraint figure 8.c and limiting to 3 the number of bits 8.d) implies the old one $c^T\mathbf{x} < b$.

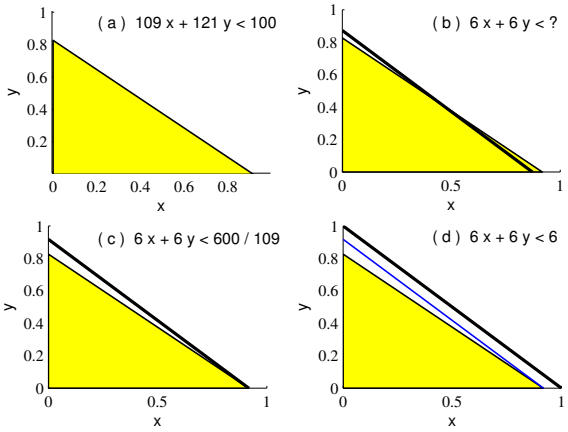

Fig. 8. Limiting the number of bits of a constraint(Asarin et al., 2006)

Zonotopes The last type of polyhedrons useful to express continuous space regions is zonotopes (Kühn, 1998; Girard, 2005) that are compact and closed for most operations involved in reachability computation.

A zonotope is defined by its centre c and its generators $g_1, ..., g_m$ by:

$$\mathbf{Z} = (c, \langle g_1, ..., g_m \rangle) = \{c + \sum_{j=0}^{m} \alpha_j g_j \mid \forall j,\ \alpha_j \in [-1; 1]\}$$

that specifies it in a very compact way (figure 9). Moreover the set of zonotopes is the smaller set of connex regions, such that the regions are not reduced to a point, and the set is closed under linear transformation and Minkowski sum (Combastel, 2003). This type of polyhedrons is then really interesting for continuous reachability analysis of linear systems with bounded inputs.

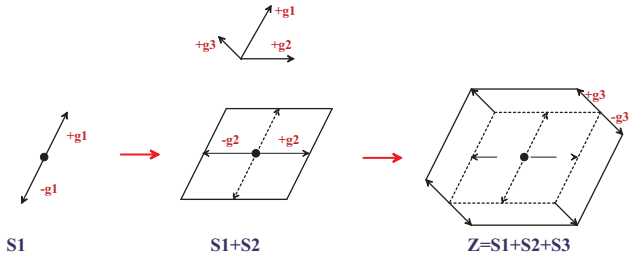

Fig. 9. Planar zonotope Z with 3 generators

As for general polyhedrons, the iteration of reachability computation leads to increase the number of generators and the complexity of zonotopes. It may therefore be

mandatory, in order to make the computation simpler, to substitute some zonotopes by others that include them but have less generators.

Using zonotopes for linear systems allows to compute, with a sampled approach, an over-approximation of the reachable space for high dimension systems. One main limit when using zonotopes for hybrid systems is linked to the computation of intersection of the reachable space with invariant and guard conditions.

4.3 Complexity reduction

As stated above the complexity of the reachability computation is linked to dimension of continuous state space and to the continuous dynamics characteristics. Methods to reduce complexity of systems to change them in systems tractable by existing algorithms are based on these two points.

Dimension reduction The first idea in order to reduce the dimension of the system is to display some specific sub-spaces of the state space such that the projection of the state in one sub-space is of low influence on the behaviour of the projection in the other one (Asarin and Dang, 2004; Han and Krogh, 2005, 2006). If such sub-spaces exist, the computation may be performed in each sub-space considering the influence of the other one as a disturbance.

An other approach for dimension reduction, based on trajectories similarities is proposed by Girard et al. (2006). This approach makes it possible to build a reduce order system and to guaranty that its trajectories are in the neighbourhood of the projections of the initial system with a given accuracy. From the result of the reachability computation for the reduced system and the guarantied accuracy, it is possible to obtain an approximation of the reachable space that leads to the conclusion whether a given region is reachable or not.

One main difficulty with these projections and reduced order approaches is that, when a hybrid system is considered, the reduced order state space may not be the same for all locations. It is then necessary at transition firing time to establish the new starting region in one space from the intersection of the reachable region with the transition guard in the other space. This may introduce approximations that have to be limitted.

Hybridisation A second approach to reduce the complexity of computation consists in approximating continuous dynamics by a simpler one (Asarin et al., 2003, 2002). An example of this approach is linearisation of non-linear dynamics with the computation of a bounded error. This leads to approximate the equation $\dot{\mathbf{x}} = f(\mathbf{x})$ by the equation $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}$ where $\mathbf{b}$ is a bounded uncertainty such as for all $\mathbf{x}$, $f(\mathbf{x})$ is in $A\mathbf{x} + \mathbf{b}$. Of course, the smaller the region on which the approximation is computed, the better it may be. This approach is then based on a partition of the invariant of each location and the computation of a piecewise affine uncertain system whose reachable space is an over-approximation of the real one. One difficulty associated with this approach in order to combine simplicity of computation with accuracy of the result stands in the choice of the partition elements and linearization method.

5. CONCLUSION

Verification of continuous time hybrid systems may be investigated along two main directions that are quite similar to the ones used for discrete time systems with different computation technics. The first one consists in building an equivalent discrete event system that is used to check the property. This leads to local reachability considerations. The second one considers the global reachability problem to check the property on the hybrid system. As in both cases the reachability computation is complex, over-approximations of the reachable space are often considered as they allow to check whether safety properties are met but less easily when they are not.

The general principles of verification such as abstraction, hybrid reachability computation, counter-example refinement, forward and backward research iteration are well established and the main works are now connected to computational aspects that allows their implementation. The aim is to find the best compromise between compacity of sets coding, the relevance of such regions, the complexity and the accuracy of computations. Until now, the choices have been mainly driven by continuous reachability and differential equation solving. For hybrid systems, transitions firing is however an important point to take into account, as a good choice from the point of view of continuous reachability may lead to large approximations when computing intersections with guards of transitions and images by jump functions. The global result of the hybrid reachability computation could then be rather rough.

Finally, it is important to be able to integrate these approaches with other control design tools. This will allow control engineers to use various checking approaches and strategies in a global control systems engineering process.

REFERENCES

R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoritical Computer Science*, 138:3–34, 1995.

R. Alur, T. Dang, and F. Ivancic. Reachability analysis of hybrid systems via predicate abstraction. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control: 5th International Workshop, HSCC 2002, Stanford, CA, USA, LNCS 2289*, pages 35–48. Springer, march 2002.

R. Alur, F Ivancic, and T. Dang. Progress on reachability analysis of hybrid systems using predicate abstraction. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, LNCS 2623*, pages 4–19. Springer, april 2003.

E. Asarin and T Dang. Abstraction by projection and application to multi-affine systems. In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC2004, Philadelphia, PA, USA, LNCS 2993*, pages 32–47. Springer, 2004.

E. Asarin, G. Schneider, and S. Yovine. Towards computing phase portraits of polygonal differential inclusions. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control: 5th International Workshop, HSCC 2002, Stanford, CA, USA, LNCS 2289*, pages 49–61. Springer, march 2002.

E. Asarin, T. Dang, and A. Girard. Reachability analysis of non-linear systems using conservative approximation. In O. Maler and A. Pnueli, editors, *HSCC2003*, pages 20–35. Springer, april 2003.

E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O.Maler. Recent progress in continuous and hybrid reachability analysis. In *CACSD06, Munich, Germany,*, october 2006.

R. Bagnara, E. Ricci, E. Zaffanella, and P.M. Hill. Possibly not closed convex polyhedra and the parma polyhedra library. In M.V. Hermenegildo and G. Puebla, editors, *Proc. of Int Symp on Static Analysis LNCS 2477*, pages 213–229. Springer, 2002.

C. Belta, P Finin, L. Habets, A. Halasz, M. Imielinski, R. Vijay Kumar, and H. Rubin. Understanding the bacterial stringent response using reachability analysis of hybrid systems. In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC2004, Philadelphia, PA, USA, LNCS 2993*, pages 111–125. Springer, 2004.

S. Blouin, M. Guay, and K Rudie. Discrete abstractions for two dimensional nearly integrable continuous systems. In S. Engel, H. Guéguen, and J. Zaytoon, editors, *ADHS03 : IFAC conference on Analysis and Design of Hybrid Systems, Saint-Malo, France*, pages 343–348. IFAC, Elsevier, juin 2003.

F. Cassez, T Henzinger, and J.F. Raskin. A comparison of control problems for times and hybrid systems. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control: 5th International Workshop, HSCC 2002, Stanford, CA, USA, LNCS 2289*, pages 134–148. Springer, march 2002.

A. Chutinan and B. Krogh. Verification of infinite-state dynamic systems using approximate quotient transition systems. *IEEE Trans. on Automatic Control*, 46:1401–1410, 2001.

A. Chutinan and B. Krogh. Computation techniques for hybrid system verification. *IEEE Trans. on Automatic Control*, 48:64–75, 2003.

C. Combastel. A state bounding observer based on zonotopes. In *ECC: The 7th Workshop on Elliptic Curve Cryptography*, 2003.

T. Dang. *Vérification et synthèse des systèmes hybrides.* PhD thesis, INPG, octobre 2000.

T. Dang. Approximate reachability computation for polynomial systems. In J Hespanha and A Tiwari, editors, *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC2006, Santa Barbara, CA, USA, LNCS 3927*, pages 138–152. Springler, March 2006.

A. Fehnker, E. Clarke, S. Jha, and B. Krogh. Refining abstractions of hybrid systems using counterexample fragments. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 242–257. Springer, march 2005.

G. Frehse. Phaver: algorithmic verification of hybrid systems past hytech. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 258–273. Springer, march 2005.

A. Girard. Reachability of uncertain linear systems using zonotopes. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 291–305. Springer, march 2005.

A. Girard, A.A. Julius, and G. Pappas. Approximate simulation relations for hybrid systems. In C.G. Cassandras, A. Giua, C. Seatzu, and J. Zaytoon, editors, *2nd IFAC Conference on Anlysis and Design of Hybrid Systems, ADHS06,*, pages 106–111, june 2006.

S. Glavaski, A. Papachristodoulou, and K. Ariyur. Safety verification of controlled advanced life support system using barrier certifcates. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 306–321. Springer, march 2005.

H. Guéguen and J. Zaytoon. On the formal verification of hybrid systems. *Control Engineering Practice*, 12(10): 1253–1268, 2004.

Z. Han and B. Krogh. Reachability analysis for affine systems using ϵ-decomposition. In *ECC CDC 2005.* IEEE - EUCA, december 2005.

Z. Han and B. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In J Hespanha and A Tiwari, editors, *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC2006, Santa Barbara, CA, USA, LNCS 3927*, pages 287–301. Springer, march 2006.

T. Henzinger, P.H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1:110–122, 1997.

T. A. Henzinger, P.H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):540–554, april 1998.

T. Hickey and D. Wittenberg. Rigourous modelling of hybrid systems using interval arithmetic constraints. In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC2004, Philadelphia, PA, USA, LNCS 2993*, pages 402–416. Springer, 2004.

M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In J Hespanha and A Tiwari, editors, *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC2006, Santa Barbara, CA, USA, LNCS 3927*, pages 348–362. Springer, march 2006.

W. Kühn. Zonotope dynamics in numerical quality control. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 125–134. Springer, 1998.

A.B. Kurzhanski and P. Variya. Ellipsoïdal techniques for reachability analysis. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control: third International Workshop, HSCC 2000, Pittsburgh, PA, USA, LNCS 1790*, pages 202–214. Springer, march 2000.

G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In F. Vaandrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control: Second International Workshop, HSCC'99, LNCS 1569*, pages 137–151. Springer, 1999.

M.A. Lefebvre and H. Guéguen. Hybrid abstractions of affine systems. *NonLinear Analysis*, 65:1150–1167, September 2006.

S. Mitra, Y. Wang, N. Lynch, and E. Feron. Safety verification of model helicopter controller using hybrid input/output automata. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, LNCS 2623*, pages 343–358. Springer, april 2003.

N.S. Nedialkov, K.R. Jackson, and G.F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105:21–68, 1999.

G. Della Penna, B. Intrigila, I. Melatti, A. Parisse, M. Minichino, E. Ciancamerla, E. Tronci, and M. Venturini Zilli. Automatic verification of a turbogas control system with the murϕ verifier. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, LNCS 2623*, pages 141–155. Springer, april 2003.

S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC2004, Philadelphia, PA, USA, LNCS 2993*, pages 477–492. Springer, 2004.

S. Prajna and A. Rantzer. Primal-dual tests for safety and reachability. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 542–556. Springer, march 2005.

S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 573–589. Springer, march 2005.

R.J.Lohner. Enclosing the solutions of ordinary iniial and boundary value problems. In *Computer Arithmethic: Scientific Computation and Programming Languages, Wiley-Teubner Series in Computer Science*, pages 255–286, 1987.

E. Rodriguez-Carbonell and A. Tiwari. Generating polynomial invariance for hybrid systems. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC2005, Zurich, Switzerland, LNCS 3414*, pages 590–605. Springer, march 2005.

P. Schnoebelen, B. Bérard, F. Laroussinie, M. Bidoit, and A. Petit. *Vérification de logiciels : techniques et outils du model-checking*. Vuibert, 2004.

O. Stursberg, A. Fehnker, Z. Han, and B. Krogh. Verification of a cruise control system using counterexample-guided search. *Control Engineering Practice*, 12(10): 1269–1278, 2004.

P. Tabuada, G. Pappas, and P. Lima. Composing abstractions of hybrid systems. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control: 5th International Workshop, HSCC 2002, Stanford, CA, USA, LNCS 2289*, pages 436–450. Springer, march 2002.

A. Tiwari. Approximate reachability for linear systems. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, LNCS 2623*, pages 514–525. Springer, april 2003.

A. Tiwari and G. Khanna. Nonlinear systems: approximating reach sets. In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC2004, Philadelphia, PA, USA, LNCS 2993*, pages 600–614. Springer, 2004.

C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceeding of the IEEE*, 91:986–1001, July 2003.

H. Yazarel and G. Pappas. Geometric programming relaxations for linear systems reachability. In *American Control Conference*, 2004.

H. Yazarel, S. Prajna, and G.J. Pappas. Sos for safety. In *43rd IEEE CDC*, 2004.