# d/dt: A TOOL FOR REACHABILITY ANALYSIS OF CONTINUOUS AND HYBRID SYSTEMS [1]

**Eugene Asarin, Thao Dang, and Oded Maler**

*Verimag, 2, av. de Vignate, Centre Equation*
*38610 Gières, France.*

Abstract: This paper describes the tool **d/dt** which allows to solve various reachability-based problems for linear and non-linear, continuous and hybrid dynamical systems. The basic functionality of **d/dt** is to take a dynamical system defined by a differential equation $\dot{\mathbf{x}} = f(\mathbf{x})$ and a set $F$ of initial states and give an over-approximation of the "tube" of all trajectories starting from any point $\mathbf{x}_0 \in F$. To this end, two reachability techniques are proposed: one is applicable for non-linear systems, and the other is specialized for linear systems possibly with uncertain bounded input. Both methods use polyhedra to approximate reachable sets. Extensions of these techniques include verification of hybrid systems and switching controller synthesis.

Keywords: Reachability analysis, hybrid systems, verification, switching controller synthesis

## 1. INTRODUCTION

Recent developments in embedded control systems have motivated interest in the reachability problem for hybrid dynamical systems, that is systems that contain both discrete and continuous dynamics; see for example (Chutinan and Krogh, 1999; Pappas *et al.*, 1999; Botchkarev and Tripakis, 2000; Kurzhanski and Varaiya, 2000) and references therein. Effective reachability computation for hybrid systems is a key to verifying their correctness. Reachabilily computation can also be used to synthesize controllers that keep the system within a pre-specified region of the state space (Tomlin *et al.*, 2000; Asarin *et al.*, 2000). The main ingredient of such reachability algorithms is a method for representing and computing reachable sets of continuous systems. In this paper we develop a framework for approximate solutions to the reachability problem for linear and non-linear, continuous and hybrid systems and present new techniques for approximating reachable sets of such systems. These reachability techniques were implemented in a tool, called **d/dt** which, given a description of the system and a set of initial states, can produce a polyhedral approximation of the reachable set. The tool also allows to solve various reachability-based problems such as safety verification and controller synthesis. The paper is organized as follows. Section 2 presents our approach to the reachability problem for continuous systems and an abstract algorithm for its solution. Based on this algorithm, we develop in Section 3 a technique for approximating reachable sets of general non-linear continuous systems. Section 4 presents a more efficient technique specialized for linear contin-

[1] Partially supported by the European project Esprit-LTR 26270 VHS (Verification of Hybrid Systems).

uous systems possibly with uncertain, bounded input. Section 5 discusses the applications of these techniques for solving the reachability, verification and controller synthesis problems for hybrid systems. Finally, in Section 6 we describe briefly the tool **d/dt** and illustrate our approach with an example.

## 2. REACHABILITY PROBLEM

Consider a continuous system $\mathcal{C}$ defined by the ordinary differential equation

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ and $f$ is globally Lipschitz over all $\mathbf{x} \in \mathcal{X}$. Before stating the reachability problem, we introduce some definitions and notations. Let $\mathcal{T} = \mathbb{R}^+$ be the *time domain.* Let $\xi_{\mathbf{x}} : \mathcal{T} \to \mathcal{X}$ be a solution of (1) with initial condition $\mathbf{x}(0) = \mathbf{x}$.

*Definition 1.* (Reachable states). The reachability operator $\delta_I : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ is defined for a subset $F$ of $\mathcal{X}$ and a time interval $I \subseteq \mathcal{T}$ as $\delta_I(F) = \{\mathbf{x}' \mid \exists \mathbf{x} \in F\ \exists t \in I\ \mathbf{x}' = \xi_{\mathbf{x}}(t)\}$.

For simplicity, we use the notation $\delta_r$ for $\delta_{[r,r]}$ (states reachable after exactly $r$ time) and $\delta_I(\mathbf{x})$ for $\delta_I(\{\mathbf{x}\})$. The reachable set from $F$ is therefore $\delta_{[0,\infty)}(F)$, denoted by $\delta(F)$.

Our reachability problem is stated as follows. Given a set $F \subseteq \mathcal{X}$, we want to compute the set $\delta(F)$ of states reachable from $F$ by the system $\mathcal{C}$. We emphasize that by "computing" we mean providing a representation of the reachable set that can be tested for intersection with another set. The problem of exact computation of reachable sets is difficult, and even for linear systems it can be done only for restricted classes of matrices (Pappas *et al.*, 1999; Anai and Weispfenning, 2001). In this paper we focus on techniques of approximation of such sets.
Numerical integration is a common method to approximate solutions of differential equations. The basic idea of this approach is the following. Given a time step $r > 0$, starting from a single point $\mathbf{x}$, one can numerically integrate the differential equation to obtain an approximation $\hat{\xi}_{\mathbf{x}}$ of the solution $\xi_{\mathbf{x}}$ at time points $r, 2r, \ldots$. This approach, although not adequate for solving the reachability problem since we need to compute solutions for all points in the initial set and for all time points $t > 0$, suggests the following iterative algorithm for approximating reachable sets.

*Algorithm 1.* (Computation of $\delta(F)$).
$P^0 := F$;
**repeat** $k = 0, 1, 2, \ldots$
$P^{k+1} := P^k \cup \delta_{[0,r]}(P^k)$;
**until** $P^{k+1} = P^k$

In order to efficiently implement Algorithm 1, we use non-convex orthogonal polyhedra (which can be described as unions of closed full-dimensional hyper-rectangles) to store reachable states. Such polyhedra admit several canonical representations (Bournez *et al.*, 1999) allowing efficient manipulation including Boolean operations, equivalence checking *in any dimension.* Since orthogonal polyhedra are closed under the union operation, the set of reachable states accumulated over the execution of the algorithm is represented as a unique polyhedron; this facilitates the test for termination of the algorithm. In the following two sections we present two techniques for over-approximating reachable sets of continuous systems based on Algorithm 1. The first technique can be used for non-linear systems, and the second is specialized for linear systems and extended to systems with uncertain, bounded input. Note that for safety verification purposes over-approximations of reachable sets are needed. However, other problems, such as controller synthesis, require under-approximations of these sets.

## 3. REACHABILITY TECHNIQUE FOR NON-LINEAR CONTINUOUS SYSTEMS

This reachability technique, which we call "face lifting" is inspired by the technique suggested in (Greenstreet, 1996) for two-dimensional systems. The advantage of our technique is its easy application for higher dimensions.

We assume now that $F$ is a *convex polyhedron.* The solution can be generalized to problems where the initial sets are non-convex polyhedra since these can be decomposed into finitely many convex polyhedra. Due to continuity of trajectories, to compute the reachable set $\delta(F)$ it is sufficient to look at the boundary of $F$.

Let $E$ be the set of faces of $F$. Consider a face $e \in E$ whose supporting plane is $P(e) = \{\mathbf{x} \mid \langle \mathbf{n}(e), \mathbf{x} \rangle = c_e\}$ where $\mathbf{n}(e)$ is the *unit outward normal* to $e$. Thus, the polyhedron $F$ can be written as intersection of half-spaces: $F = \bigcap_{e \in E} H(e)$ where $H(e) = \{\mathbf{x} \mid \langle \mathbf{n}(e), \mathbf{x} \rangle \leq c_e\}$. We start by constructing the neighborhoods of the faces of $F$ as follows. The neighborhood of $F$ is defined as $N_{\boldsymbol{v}}(F) = \bigcap_{e \in E} \{\mathbf{x} \mid \langle \mathbf{n}(e), \mathbf{x} \rangle \leq c_e + v_e\}$, that is the polyhedron obtained by pushing outward each face $e$ of $F$ by an amount $v_e > 0$. Then, the neighborhood of face $e$ is defined as $N(e) = H_o(e) \cap N_{\boldsymbol{v}}(F)$ where $H_o(e) = \{\mathbf{x} \mid \langle \mathbf{n}(e), \mathbf{x} \rangle \geq c_e\}$ (see Figure 1).

The technique is based on the fact that we can find the time step $r$ and the neighborhood $N_{\boldsymbol{v}}(F)$ such that *all trajectories starting from the boundary of $F$ stay in $N_{\boldsymbol{v}}(F)$ for at least $r$ time*. This plays a crucial role in the subsequent discussion. Let $f_e(\mathbf{x})$ denote the *outward component* of $f(\mathbf{x})$ relative to $e$, i.e. $f_e(\mathbf{x}) = \langle \mathbf{n}(e), f(\mathbf{x}) \rangle$. Let $\hat{f}_e$ be the maximum of $f_e$ over $N(e)$: $\hat{f}_e = \max\{f_e(\mathbf{x}) \mid \mathbf{x} \in N(e)\}$. A trajectory starting from the boundary of $F$

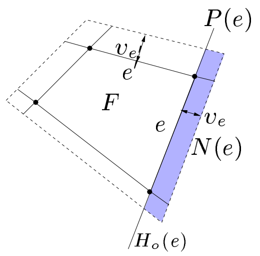



Fig. 1. Neighborhood construction.

which goes out of $F$ at some time $t \in (0, r]$ is called *outward trajectory*. Consider an outward trajectory $\xi_{\mathbf{x}}$ starting from point $\mathbf{x} \in e$. By solution of the differential equation (1) we have $\langle \mathbf{n}(e), \xi_{\mathbf{x}}(t) \rangle = c_e + \int_0^t f_e(\mathbf{x}(s)) ds$. Thus, for every time point $t \in [0, r]$ we have $\langle \mathbf{n}(e), \xi_{\mathbf{x}}(t) \rangle \leq c_e + t\hat{f}_e$. Let $\bar{f}_e = \hat{f}_e$ if $\hat{f}_e > 0$ and $\bar{f}_e = 0$ otherwise. It is not hard to see that for all $t \in [0, r]$ $\langle \mathbf{n}(e), \xi_{\mathbf{x}}(t) \rangle \leq c_e + r\bar{f}_e$. This means that all the points reachable from $e$ in the time interval $[0, r]$ remain inside the half-space $H_l(e) = \{\mathbf{x} \mid \langle \mathbf{n}(e), \mathbf{x} \rangle \leq c_e + r\bar{f}_e\}$. Geometrically speaking, $H_l(e)$ is the half-space resulting from lifting $H(e)$ in the direction $\mathbf{n}(e)$ by the amount $r\bar{f}_e$. It then follows that $\delta_{[0,r]}(e) \subseteq H_l(e)$. Now, consider a face $e'$ which is *adjacent to* $e$. Similarly, we have $\delta_{[0,r]}(e') \subseteq H_l(e')$. Furthermore, we can prove that any outward trajectory from $e$ cannot leave the half-space $H_l(e')$ at time $t \in [0, r]$ (see (Dang, 2000)). Therefore, the polyhedron $F_l = \bigcap_{e \in E} H_l(e)$ is an over-approximation of the reachable set $\delta_{[0,r]}(F)$. As an example, consider the initial polygon $F$ shown in Figure 2 (left); only edges $e$

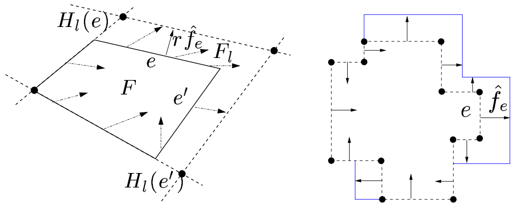



Fig. 2. Illustration of the "face lifing" technique.

and $e'$ have a positive outward component of $f$ and their half-spaces are thus pushed outward.

It remains to use this procedure for over-approximating the operator $\delta_{[0,r]}$ in Algorithm 1 to compute an over-approximation of the whole reachable set. Since the calculation of the lifting amount for every face $e$ is based uniquely on the maximum of $f_e$ over $N(e)$, to avoid excessively conservative approximations some of the faces must be split a priori and, as a result, the resulting set need not be convex. In three or higher dimensions, manipulation of arbitrary non-convex polyhedra is complex; therefore, we restrict ourselves to orthogonal polyhedra since their faces which are $(n-1)$-dimensional hyper-rectangles can be systematically enumerated. Moreover, such polyhedra are closed under the lifting operation (see Figure 2 (right)).

Note that the correctness of our algorithm is guaranteed by the assumption that all trajectories departing from the boundary of $F$ remain in $N_{\boldsymbol{v}}(F)$ during the time interval $[0, r]$. In (Dang, 2000) we present a procedure for determining the time step $r$ and the neighborhood size $v_e$ so that this assumption is always fulfilled. We also prove that *locally the error can be made arbitrarily small* by fine-tuning these parameters. However, because of the accumulation of over-approximation error this technique can guarantee error bounds only for a *finite* time horizon. In the next section we present a technique specialized for linear systems which can avoid the over-approximation error accumulation.

## 4. REACHABILITY TECHNIQUE FOR LINEAR CONTINUOUS SYSTEMS

We describe first an algorithm for approximating reachable sets of linear continuous systems without input and then an extension to systems with uncertain, bounded input.

We consider a linear system $\dot{\mathbf{x}} = A\mathbf{x}$. As before, we assume an initial convex polyhedron $F$

which can be written as the convex hull of its vertices. Since $\delta_r(F) = e^{Ar}F$, the reachable set at time point $r$ is simply the convex hull of the points reachable at time $r$ from the vertices of $F$ (which can be computed by matrix exponentiation). To over-approximate $\delta_{[0,r]}(F)$, we observe that the convex hull $C = conv(F \cup \delta_r(F))$ is, in general, neither a subset nor a superset of $\delta_{[0,r]}(F)$. However, we can find an upper bound $\vartheta = \mathcal{O}(r^2)$ on the distance between $C$ and the exact reachable set and it thus suffices to push outward the half-spaces of $C$ by $\vartheta$ to obtain a new polyhedron $C_o$, and we write it as $C_o = bloat(C, \vartheta)$. Although $C_o$ is already an over-approximation of $\delta_{[0,r]}(F)$, as mentioned earlier, to facilitate termination checking we over-approximate $C_o$ by an orthogonal polyhedron. Figure 3 illustrates this reachability procedure

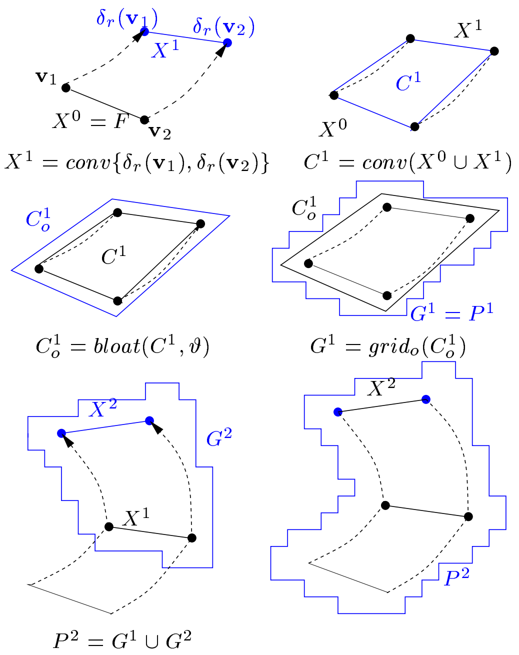



Fig. 3. Illustration of the reachability technique for linear systems.

with a simple two-dimensional example where $F$ is a line segment with two vertices $\mathbf{v}_1$ and $\mathbf{v}_2$. The dotted lines in the first figure are the real trajectories starting from the vertices of $F$, and the exact set $\delta_{[0,r]}(F)$ lies between these lines. After the first iteration we obtain the orthogonal polyhedron $G^1 \supseteq \delta_{[0,r]}(F)$. In the next iteration, we repeat these four steps *starting from* $X^1 = \delta_r(F)$ to obtain $G^2 \supseteq \delta_{[r,2r]}(F)$. The orthogonal polyhedron $P^2 = G^1 \cup G^2$ is thus an over-approximation of the reachable set after two iterations. An important advantage of this procedure is that it *avoids the propagation of over-approximation error* since $P^2$ is computed based on $X^1$ which does not contain over-approximation error (assuming that the matrix exponential can be exactly computed). Moreover, we can prove that the *global error in the approximation can be made arbitrarily small* by changing the time step $r$ and the size of hyper-rectangles in orthogonal polyhedra. This technique is more time-efficient than the technique "face lifting" and allows us to treat systems in up to 6 dimensions.

In the following we present an extension of this technique to *linear systems with uncertain, bounded input* of the form $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}$ where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u} \in U$, a convex and compact set. We assume a set of admissible inputs $\mathcal{U}$ consisting of measurable functions $\mu : \mathcal{T} \to U$. To compute reachable sets of such systems, we need to modify the algorithm for systems without input since it is impossible to simulate trajectories from the vertices of $F$ with all possible inputs. Inspired by the work (Varaiya, 1998), to compute the reachable set $\delta_t(F)$ at time $t$, we exploit the fact that for each half-space $H$ of the polyhedron $F$ there exists an input function $\mu^* \in \mathcal{U}$ such that calculating the reachable set from $H$ under $\mu^*$ is sufficient to derive a tight polyhedral approximation of $\delta_t(F)$. Intuitively, the evolution of the normal vector $\lambda$ of $H$, governed by the adjoint system $\dot{\lambda} = -A^T\lambda$, is independent of the input; therefore, $\mu^*(t)$ is the input that steers the half-space $H$ "farthest" in the direction $\lambda(t)$ and it can be determined using the Maximum Principle. In addition, the estimation of the bloating amount $\vartheta$ applied to the convex hull $C$ that guarantees over-approximations must take into account uncertainty in the input; see details in (Dang, 2000).

## 5. REACHABILITY ANALYSIS OF HYBRID SYSTEMS

This section shows how the abovedescribed reachability techniques for continuous systems are extended to hybrid systems. The formalism we use to model hybrid systems is hybrid automata (Alur *et al.*, 1995). Figure 4 is a graphical illustration of a hybrid automaton with two modes $q_1$ and $q_2$ whose continuous dynamics are defined by the differential equations $\dot{\mathbf{x}} = f_1(\mathbf{x})$ and $\dot{\mathbf{x}} = f_2(\mathbf{x})$, respectively. The staying conditions of a mode (e.g. $\mathbf{x} \in H_1$ for mode $q_1$) must be satisfied by the continuous variable $\mathbf{x}$ in order for the system to stay in that mode, and the conditions for switching between modes are defined by transition guards (e.g.

$\mathbf{x} \in G_{12}$ for the transition from $q_1$ to $q_2$). While being at mode $q_1$ where $\mathbf{x} \in H_1$, the system evolves according to $\dot{\mathbf{x}} = f_1(\mathbf{x})$. Whenever it reaches a point $\mathbf{x}' \in G_{12}$, the transition from $q_1$ to $q_2$ is enabled and the system can switch to mode $q_2$. At $q_2$ the continuous variable is reset to a new value according to the reset function $R_{12}$, which becomes then the initial state for the continuous evolution according to $\dot{\mathbf{x}} = f_2(\mathbf{x})$. In this work, all the staying sets and transition guards $H_q$ and $G_{qq'}$ are *convex polyhedra* in $\mathbb{R}^n$ and all the reset functions are of the form $R_{qq'}(\mathbf{x}) = D_{qq'}\mathbf{x} + J_{qq'}$ where $D_{qq'}$ is an $n \times n$ matrix and $J_{qq'}$ is a convex polyhedron in $\mathbb{R}^n$. Note that this captures both deterministic and non-deterministic resets.

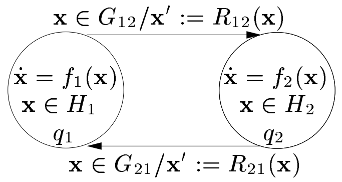



Fig. 4. A hybrid automaton.

Reachability computation for hybrid systems requires successive computations of reachable states by continuous dynamics and discrete transitions. Unlike continuous systems, the continuous evolution of the hybrid automaton at a mode $q$ is constrained by the staying conditions of the mode, which means that whenever the system leaves $H_q$ no further continuous evolution at the current mode $q$ is possible. Hence, to account for staying conditions we modify the reachability algorithms for continuous systems as follows. In each iteration we intersect the computed reachable set with $H_q$ and start the next iteration from the resulting set which contains only states satisfying the staying conditions. We turn now to the computation of reachable states by discrete transitions. The set of states reachable from a set $P$ by the discrete transition from $q$ to $q'$ can be written as $R_{qq'}(P \cap G_{qq'}) \cap H_{q'}$. This set consists of points in $H_{q'}$ resulting from applying the reset function $R_{qq'}$ to all points in $P$ satisfying the transition guard $G_{qq'}$. Therefore, in our context, the computation of reachable states by discrete transitions involves Boolean and geometric operations on polyhedra.

## 6. THE TOOL **d/dt**

We have implemented the above reachability algorithms in the tool **d/dt**. The tool is also extended to include *automatic safety verification* and *switching controller synthesis* for hybrid systems. To prove a safety property which states that the hybrid system never reaches a set of bad states, we iterately over-approximate the reachable set of the system and check for its intersection with the set of bad states. The problem of safety controller synthesis, on the other hand, is to find a controller that switches the system between modes in order to meet a safety specification. The synthesis of such a controller is based on the maximal invariant set, that is the set of states from which the controller by switching properly can avoid bad states, and the reachability technique for hybrid systems can be used to approximate this set (Asarin *et al.*, 2000).

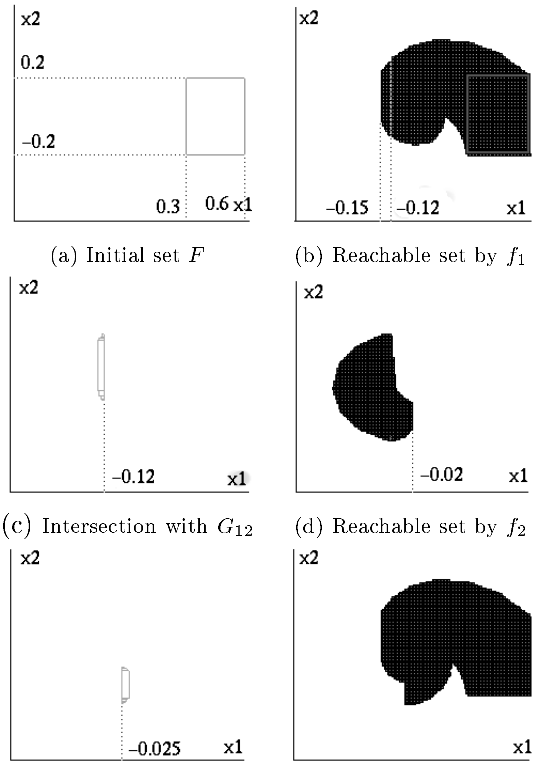
To illustrate our approach, we present now the results obtained using the tool **d/dt** for the hybrid automaton of Figure 4. The continuous dynamics of the two modes are $f_1 = A_1\mathbf{x} + \mathbf{u}$, $\mathbf{u} \in U = [-0.1, 0.2] \times [-0.2, 0.1]$ and $f_2 = A_2\mathbf{x}$ where

$$A_1 = \begin{pmatrix} -2.0 & -3.0 \\ 3.0 & -2.0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0.0 & -0.6 \\ 3.0 & 0.0 \end{pmatrix}.$$

The staying conditions $H_1$ and $H_2$ are $x_1 \geq -0.15$ and $x_1 \leq -0.02$, respectively; the transition guards $G_{12}$ and $G_{21}$ are $\mathbf{x}_1 \leq -0.12$ and $\mathbf{x}_1 \geq -0.025$; all the reset functions are the identity. The initial set is the rectangle $F = [0.3, 0.6] \times [-0.2, 0.2]$ shown in Figure 5-(a), and the system starts at $q_1$. The reachable states from $F$ by $f_1$ are computed until the trajectories all go out of $H_1$ (see Figure 5-(b)). From the intersection with the guard $G_{12}$ shown in Figure 5-(c) the dynamics $f_2$ is applied (see Figure 5-(d)). Again, the intersection of the reachable set with the guard $G_{21}$ is computed, and from there no new reachable states by $f_1$ satisfy the guard $G_{12}$ and the computation terminates (see Figures 5-(e,f)).

## 7. CONCLUSION

We have presented the reachability techniques for continuous systems and their applications for the analysis and control of hybrid systems. The algorithms were implemented in the tool **d/dt**. We have applied successfully the tool to some examples inspired by real-life applications (see (Dang, 2000; Asarin *et al.*, 2001; Belta *et al.*, n.d.)). The tool is currently under testing with examples taken from various applications,




(a) Initial set $F$ (b) Reachable set by $f_1$ (c) Intersection with $G_{12}$ (d) Reachable set by $f_2$ (e) Intersection with $G_{21}$ (f) Reachable set by $f_1$

Fig. 5. Computation of the reachable set.

and more improvements can be made based on the accumulated experience. We hope that our reachability techniques will be used by control engineers to complement existing analysis and control methods.

## 8. REFERENCES


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

Anai, H. and V. Weispfenning (2001). Reach set computations using real quantifier elimination. In: *Hybrid Systems: Computation and Control* (M.D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds.). LNCS 2034. Springer-Verlag. pp. 63–75.

Asarin, E., O. Bournez, T. Dang, O. Maler and A. Pnueli (2000). Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE* **88**, 1011–1025.

Asarin, E., S. Bansal, B. Espiau, T. Dang and O. Maler (2001). On hybrid control of under-actuated mechanical systems. In: *Hybrid Systems: Computation and Control* (M.D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds.). LNCS 2034. Springer-Verlag. pp. 77–88.

Belta, C., J. Schug, T. Dang, V. Kumar, G.J. Pappas, H. Rubin and P. Dunlap (n.d.). Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium *vibrio fisheri*. Submitted to CDC 2001, Florida.

Botchkarev, O. and S. Tripakis (2000). Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In: *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, Eds.). LNCS 1790. Springer-Verlag. pp. 73–88.

Bournez, O., O. Maler and A. Pnueli (1999). Orthogonal polyhedra: Representation and computation. In: *Hybrid Systems: Computation and Control* (F. Vaandrager and J. van Schuppen, Eds.). LNCS 1569. Springer-Verlag. pp. 46–60.

Chutinan, A. and B.H. Krogh (1999). Verification of polyhedral invariant hybrid automata using polygonal flow pipe approximations. In: *Hybrid Systems: Computation and Control* (F. Vaandrager and J. van Schuppen, Eds.). LNCS 1569. Springer-Verlag. pp. 76–90.

Dang, T. (2000). Verification and Synthesis of Hybrid Systems. PhD thesis. INPG, Verimag.

Greenstreet, M.R. (1996). Verifying safety properties of differential equations. In: *Computer Aided Verification, CAV'96* (R. Alur and T.A. Henzinger, Eds.). LNCS 1102. Springer-Verlag. pp. 277–287.

Kurzhanski, A. and P. Varaiya (2000). Ellipsoidal techniques for reachability analysis. In: *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, Eds.). LNCS 1790. Springer-Verlag. pp. 202–214.

Pappas, G., G. Lafferriere and S. Yovine (1999). A new class of decidable hybrid systems. In: *Hybrid Systems: Computation and Control* (F. Vaandrager and J. van Schuppen, Eds.). LNCS 1569. Springer-Verlag. pp. 29–31.

Tomlin, C., J. Lygeros and S. Sastry (2000). A game-theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE* **88**, 940–970.

Varaiya, P. (1998). Reach set computation using optimal control. In: *Proc. KIT Workshop, Verimag, Grenoble.* pp. 377–383.