Reachability Analysis of Nonlinear Systems Using Conservative Approximation

Eugene Asarin[1], Thao Dang[1], and Antoine Girard[2]

[1] VERIMAG
2 avenue de Vignate, 38610 Gières, France
[2] LMC-IMAG
51 rue des Mathématiques, 38041 Grenoble, France
{Eugene.Asarin,Thao.Dang,Antoine.Girard}@imag.fr

Abstract. In this paper we present an approach to approximate reachability computation for nonlinear continuous systems. Rather than studying a complex nonlinear system $\dot{x} = g(x)$, we study an approximating system $\dot{x} = f(x)$ which is easier to handle. In particular, the class of approximating systems we consider in this paper is piecewise linear, obtained by interpolating g over a mesh. In order to be conservative, we add a bounded input in the approximating system to account for the interpolation error. We thus develop a reachability method for systems with input, based on the relation between such systems and the corresponding autonomous systems in terms of reachable sets. This method is then extended to the approximate piecewise linear systems arising in our construction. The final result is a reachability algorithm for nonlinear continuous systems which allows to compute conservative approximations with as great degree of accuracy as desired, and more importantly, it has good convergence rate. If g is a C^2 function, our method is of order 2. Furthermore, the method can be straightforwardly extended to hybrid systems.

1 Introduction

Reachability computation is required by a variety of safety verification, analysis, and design problems for hybrid systems. The importance of the problem has motivated much research on reachability analysis of such systems (see [4, 12]). For a class of hybrid systems with piecewise constant derivatives, methods and tools for (exact) computation of reachable sets are well-developed [27, 20, 16]. For systems involving non-trivial continuous dynamics (described by differential equations), exact reachability computation is difficult, and even for linear differential equations it is feasible only for certain classes of matrices, depending on eigenstructure [19, 2]. Alternatively, several approximate methods have been developed, and some of them can be used for nonlinear continuous systems, such as [13, 7, 6, 22]. Basically, these methods numerically approximate reachable sets using a variety of set representations (such as polyhedra, level sets). A common point of these methods is that they work directly with the nonlinear differential equations, more precisely, they track the evolution of the reachable sets according to the flows of the nonlinear equations. In this work, we take an approach which differs from these methods in this aspect. The main idea of the approach is as follows.

Rather than studying a complex nonlinear system $\dot{x} = g(x)$, we study an *approximating system* $\dot{x} = f(x)$ which is easier to handle. The class of approximating systems we consider in this paper is *piecewise linear*, obtained by interpolating the function g over a mesh built on the state space of the system. Moreover, in order to be conservative, we add an input u to account for the error inherent in approximating g with f, and the result is a system with (bounded) input $\dot{x} = f(x) + u$. This construction gives rise to the question of how to deal with the input in the approximating system efficiently. We thus consider the relation between a system with input and the corresponding autonomous system in terms of reachable sets, and this study leads us to an abstract reachability algorithm for systems with input, which can then be extended to deal with the approximate interpolating systems. The final result is a reachability method for nonlinear systems which allows to compute conservative approximations with as great degree of accuracy as desired, and more importantly, it has good convergence rate. As we shall see later, if g is a C^2 function our method is of order 2. Furthermore, the method can be straightforwardly extended to hybrid systems and readily integrated in a verification tool.

The 'hybridization' idea has previously explored in [23, 17, 25, 26] where the approximating systems are systems with piecewise constant slopes or rectangular inclusions. On the other hand, similar ideas have been used for numerical integration of nonlinear differential equations [9, 10]. The reachability method proposed in [14] uses linear approximation in each integration step to obtain better approximations of the reachable sets in 2 dimensions. Recently, in [15], a control problem for a class of piecewise linear systems, similar to our approximating systems, is solved in terms of reachability conditions.

The paper is organized as follows. Section 2 is devoted to definitions and notations. In Section 3 we consider the reachability problem for (general) continuous systems with input. In Section 4 we present a method to approximate reachable sets of nonlinear systems by means of piecewise linear approximation. The theoretical result of Section 3 is the basis for the proof of the convergence of the method. Section 5 contains some examples illustrating our approach.

2 Basic definitions

We consider a nonlinear system

$$\dot{x}(t) = g(x(t)), \ x \in \mathcal{X} \subseteq \mathbb{R}^n. \tag{1}$$

As mentioned in the introduction, we approximate the system (1) with another system (which is easier to solve):

$$\dot{x}(t) = f(x(t)),\ x \in \mathcal{X} \subseteq \mathbb{R}^n. \tag{2}$$

Let μ be the bound of $||f - g||$, i.e. $||f(x) - g(x)|| \le \mu$ for all $x \in \mathcal{X}$ where $||\cdot||$ is some norm on $\mathbb{R}^n$. We assume that the function f is L-Lipschitz. In order to be able to capture all the behaviors of the original system (1), we introduce in the system (2) an input to account for the approximation error.

$$\begin{cases} \dot{x}(t) = s(x(t), u(t)) = f(x(t)) + u(t), \\ u(\cdot) \in \mathcal{U}_\mu \end{cases} \tag{3}$$

where $\mathcal{U}_\mu$ is the set of admissible inputs which consists of piecewise continuous functions u of the form $u : \mathbb{R}^+ \to \mathbb{R}^n$ such that $||u(\cdot)|| \le \mu$. It is not hard to see that the system (3) is an overapproximation of the original system (1) in the sense that all trajectories of (1) are contained in the set of trajectories of (3). $\diamond$

Given an initial point $x \in \mathcal{X}$, let $\Phi_f(t, x)$ be the trajectory starting from x of the system (2) and let $\Phi_s(t, x, u(\cdot))$ be the trajectory starting from x of the system (3) under input $u(\cdot) \in \mathcal{U}_\mu$. For a set of initial points $X_0 \subseteq \mathcal{X}$ and $T > 0$, the reachable sets of the autonomous system (2) during the interval $[0, T]$ is defined as:

$$K_f(T, X_0) = \{\, y = \Phi_f(t, x) \mid t \in [0, T], x \in X_0 \,\}.$$

Similarly, the reachable set of the system (3) from X_0 during the interval $[0, T]$ is defined as:

$$K_s(T, X_0) = \{\, y = \Phi_s(t, x, u(\cdot)) \mid t \in [0, T], x \in X_0, u(\cdot) \in \mathcal{U}_\mu \,\}.$$

3 Reachability analysis for systems with input

As mentioned earlier, with a view to deal with the input arisen in the construction of the approximating systems, we first consider the problem of deriving an approximation of the reachable set of a system with input from the reachable set of the corresponding autonomous system. More concretely, our goal is to compute the reacheable set $K_s(T, X_0)$ of the system with input (3), assuming that we are able to compute the image of a set $X \subseteq \mathcal{X}$ by the flow Φ_f of the autonomous system (2) for a given time $t \ge 0$, denoted by $\Phi_f(t, X)$.

We first describe an abstract algorithm to do so and then discuss the properties of the algorithm concerning conservativeness and convergence of the approximation. It is important to emphasize that these theoretical results are key to the validation of the reachability method for nonlinear systems, developed in the next section.

The idea to solve this problem relies on the following result, which is a consequence the Fundamental Inequality theorem [24] from the theory of dynamical systems (see Appendix A).

Lemma 1. *For all $t \geq 0$ and for all $u(\cdot) \in \mathcal{U}_\mu$,*

$$||\Phi_f(t,x) - \Phi_s(t,x,u(\cdot))|| \leq \frac{\mu}{2}(e^{Lt} - 1). \tag{4}$$

Hence, to approximate the reachable set of the system with input, we can appropriately expand the reachable set of the autonomous system by the amount given on the right hand side of the inequality (4). $\diamond$

We thus define an 'expanding' operation as follows: for a set $S \subset \mathbb{R}^n$ and a real number $\epsilon \geq 0$, the expanded set is $\mathcal{N}(S,\epsilon) = S \oplus \epsilon B$ where B is the unit ball at the origin, and $\oplus$ is the Minskowski sum. Our reachability computation procedure is summarized in Algorithm 1.

```
Input: Initial set X_0, Result: Approximation of K_s(T, X_0)
N = T/r              /* r is the time step */
ε = (μ/L)(e^{Lr} − 1)

/* –Initialization */
P_1 = K_f(r, X_0)
Q_1 = N(P_1, ε)
R_0 = Q_1

/* –Main loop */
for i ← 1 to N − 1 do
    P_{i+1} = Φ_f(r, Q_i)
    Q_{i+1} = N(P_{i+1}, ε)
    R_{i+1} = R_i ∪ Q_{i+1}
end
return R_N
```

Algorithm 1: Approximating the reachable set of the system with input

In Algorithm 1, r is the time step and the set Q_i represents an overapproximation of the reachable set during time interval $[(i-1)r,\, ir]$ of the system with input (3). The algorithm consists of the following two phases. The goal of the first phase is to initiliaze Q_1. This is done by computing P_1, which is indeed the reachable set of the autonomous system (2) for the first time interval $[0,r]$, and then expanding P_1 by the amount ϵ (which is the bound from (4)). In the main loop, each iteration i takes Q_i as input and computes Q_{i+1} as follows. The set P_{i+1} is first computed as the image of Q_i by the flow Φ_f of the autonomous system, and Q_{i+1} is then obtained by expanding P_{i+1} by ϵ. The result R_N is simply the union of all Q_i.

Properties of the approximation

We now present two properties concerning the conservativeness and convergence of the approximation produced by Algorithm 1.

Theorem 1. *Let R_N be the set computed by Algorithm 1. Then,*

- P1. (Conservative approximation) $K_s(T, X_0) \subseteq R_N$.
- P2. (Convergence of the approximation)

$$d_H(K_s(T, X_0), R_N) \leq 2\, \mu\, r\, e^{LT}.$$

where d_H is the Hausdorff distance.

As we can see from the theorem, the approximate set produced by Algorithm 1 is guaranteed to be an overapproximation of the exact reachable set. Moreover, it converges to the exact set with regard to the Hausdorff distance. The proof of the theorem can be found in Appendix B.1.

4 Reachability computation for nonlinear systems using piecewise linear approximation

In this section, we focus on the main problem of the paper, which is the computation of reachable sets for nonlinear continuous systems. Following the 'hybridization' idea, our method consists of two steps. We first approximate the (complex) nonlinear system by a piecewise (simple) linear system. A bound on the approximation error is estimated and then added to the piecewise linear system as uncertain input, which guarantees that the resulting system is indeed an overapproximation of the original system. The second step involves extending Algorithm 1, which is designed for continuous systems with input, to piecewise linear systems.

We first describe the two steps of the method and then discuss the convergence results.

4.1 Construction of approximating systems

We consider the nonlinear system

$$\dot{x}(t) = g(x(t)),\ x \in \mathcal{X} \subseteq \mathbb{R}^n \tag{5}$$

We assume that g is Lipschitz and the state space $\mathcal{X}$ is a bounded convex polyhedron in $\mathbb{R}^n$. To define an approximating system, we decompose the state space into polyhedral regions and then associate with each region a linear system using interpolation. The procedure of decomposition of the state space is called mesh generation. In the sequel, we will define these concepts formally.

Piecewise linear approximations using interpolation

Definition 1 (Mesh). *A mesh $\mathcal{M}$ of the set $\mathcal{X}$ is a finite set of convex polyhedra in $\mathbb{R}^n$, called* cells*, satisfying the following conditions: (1) The union of all cells $\bigcup_k C_k = \mathcal{X}$, and (2) If C_j and C_k are cells with non-empty intersection, then their intersection is a common face of both; we say that C_j and C_k are adjacent and denote their common face by $\partial(C_j, C_k)$.* ⋄

For a cell $C_k \in \mathcal{M}$, we denote by $V(C_k)$ the set of its vertices. The size of C_k is $h(C_k) = \max\{\, ||x - y|| \mid x, y \in C_k \,\}$. Then, the size (or granularity) of $\mathcal{M}$ is defined as $h(\mathcal{M}) = \max\{\, h(C_k) \mid C_k \in \mathcal{M} \,\}$. Two types of meshes are of practical interest: rectangular and simplicial. A mesh is called *rectangular* if its cells are all boxes in $\mathbb{R}^n$. If all the cells are simplices, then we say that $\mathcal{M}$ is a *simplicial mesh* or a *triangulation* of $\mathcal{X}$. We recall that a simplex in $\mathbb{R}^n$ is the convex hull of $(n+1)$ affinely independent points in $\mathbb{R}^n$, and $h(C_k)$ is simply the maximum edge length.

Given a mesh $\mathcal{M}$ of $\mathcal{X}$, we can derive a piecewise linear approximation of the function g, using interpolation over the mesh. We restrict our attention to simplicial meshes and the motivation will become clear in subsequent development. A discussion on mesh construction is deferred to the end of this section

Definition 2 (Piecewise linear approximation). *For each cell $C_k \in \mathcal{M}$, let $l_k : \mathbb{R}^n \to \mathbb{R}^n$ be an affine map of the form $l_k(x) = A_k x + b_k$ which interpolates g on the vertices of C_k, that is, $g(v) = l_k(v)$ for all $v \in V(C_k)$. Then, the piecewise linear approximation of g is defined as: $l(x) = l_k(x)$ if $x \in C_k$.* ⋄

The advantage of using simplicial meshes lies in the fact that the linear interpolant l_k can be defined uniquely since each cell C_k has $(n+1)$ vertice and, moreover, for any two adjacent cells C_k and C_j, we have $\forall x \in \partial(C_k, C_j)\ \ l_k(x) = l_j(x)$. This important property allows us to obtain the following approximating system:

$$\dot{x}(t) = l(x(t))$$

which is continuous and Lipschitz. This not only guarantees the existence and uniqueness of solutions, but also allows to derive a priori bound on the error of approximation, as we will show in the following. This bound will then be used to define a conservative approximating system.

Estimating interpolation error The error in the approximation of g by the abovedescribed linear interpolation is defined by the bound η of $||g(x) - l(x)||$ for $x \in \mathcal{X}$. We will estimate this bound for two cases: g is Lipschitz and g is a C^2 function. For brevity, we denote by h the size of the underlying mesh $\mathcal{M}$.

Lemma 2. *If g is Lipschitz, then*

$$\eta \leq h \frac{2n\, L}{n+1}\,,$$

where L is the Lipschitz constant of the function g.

An important remark is that the second partial derivatives of the linear approximation vanish; therefore, if g is a C^2 function, we can obtain a better error bound.

Lemma 3. *If g is a C^2 function with a second derivative bound K, then*

$$\eta \leq h^2 \frac{n^2 K}{2(n+1)^2}.$$

As we can see from the above lemmas, the bound η is of order $O(h)$ if g is Lipschitz, and it is of order $O(h^2)$ if g is a C^2 function with bounded second derivative. The proofs of these results are presented in Appendix B.2 and B.3.

Defining conservative approximating systems We can now use the bound η from the above lemmas to define an overapproximation of the nonlinear system (5), which has the form of the system with bounded input studied in Section 3:

$$\begin{cases} \dot{x}(t) = s(x(t), u(t)) = l(x(t)) + u(t), \\ u(\cdot) \in \mathcal{U}_\eta \end{cases} \tag{6}$$

Before continuing, we mention that it is straightforward to extend this method to a nonlinear systems with input of the form $\dot{x}(t) = g(x(t)) + u(t)$ where $u(\cdot) \in \mathcal{U}_\mu$ by defining an overapproximation of this system as: $\dot{x}(t) = l(x(t)) + u_1(t)$ where $u_1(\cdot) \in \mathcal{U}_\nu$ with $\nu = \eta + \mu$. $\diamond$

Using Lemma 1, we can show that the solution the approximate system (6) converges to the solution of the original system (5) and, moreover, the convergence is of the same order as the convergence of the interpolating function l to g. Indeed, for all $t \geq 0$ and for all $u(\cdot) \in \mathcal{U}_\eta$, we have

$$||\Phi_g(t,x) - \Phi_s(t,x,u(\cdot))|| \leq \frac{\eta}{2}(e^{Lt} - 1) \tag{7}$$

where $\Phi_g(t,x)$ and $\Phi_s(t,x,u(\cdot))$ are respectively the flows of the system (5) and of the system (6) under input $u(\cdot) \in \mathcal{U}_\eta$.

4.2 Reachability algorithm for piecewise linear systems

This section is concerned with the problem of computing reachable sets of the piecewise linear systems resulting from the above approximation. Naturally, such systems can be thought of as a special class of hybrid automata [1], for which existing reachability tools (such as [6, 5, 3]) can be used. In this work, we exploit the particular structure of these approximating systems in order to achieve better efficiency. Our reachability algorithm is an extension of Algorithm 1 for continuous systems with bounded input, and the convergence result is preserved.

When the system stays inside a cell, to compute the reachable sets we can combine Algorithm 1 with one of the available methods (e.g. [6, 5, 19, 3]) for the linear autonomous system. The remaining problem is to handle the changes in the dynamics that happen when the system moves from one cell to another.

Without loss of generality, we assume that initial set X_0 is a convex polyhedron inside the cell C, and let ∂C be the boundary of C. We thus focus on the problem of computing the *set of exit points*, that is, the set of points on ∂C which the system, starting from X_0, can reach to enter an adjacent cell. At these points the system changes the dynamics, and therefore it is important to detect this boundary crossing event.

Given a point $x \in C$, let $t^*(x)$ be the smallest time at which the system, starting at x, reaches the boundary ∂C. More precisely,

$$t^*(x) = \min\{\ t \geq 0 \mid \exists u(\cdot) \in \mathcal{U}_\eta : (\Phi_s(x, t, u(\cdot)) \in \partial C)\ \wedge \\ (\forall \tau \in [0, t)\ \Phi_s(x, \tau, u(\cdot)) \in int(C))\ \}.$$

A method to underapproximate $t^*(x)$ is proposed in [11]. This method also allows to determine whether the system, if starting at points in x, always remains in C, denoted by $t^*(x) = \infty$. A heuristic to envision the event of boundary crossing is to estimate t^* for the vertices of X_0. Moreover, by appropriately approximating the trajectories from the vertices of X_0 we can determine if no trajectories from X_0 will leave C, and we denote this by $t^*(X_0) = \infty$.

Note that in two dimensions, the set of exit points on the boundary between two adjacent cells is an interval; hence, to determine this set, it suffices to consider the trajectories from the vertices of X_0. In higher dimensions, we need to combine the estimation of t^* with reachability computation, as shown in Algorithm 2.

Input: Initial set X_0 inside cell C
Result: E = Set of exit points on ∂C, R = Reachable set in cell C
$t_{min} = t^*(X_0)$
if $t_{min} = \infty$ **then**
    $E = \emptyset$; $R = K_s(T, X_0)$
    return E, R
end
$R = K_s(t_{min}, X_0)$; $I = \emptyset$; $i = 0$
repeat
    $R_{i+1} = K_s(r, R_i) \cap C$
    $E = E \cup (R_{i+1} \cap \partial C)$
    $i = i + 1$
until $R_i = R_{i-1}$;
return $E, R = R_i$

Algorithm 2: Reachability computation for a cell

The algorithm first checks whether the system will always remain inside C, indicated by $t_{min} = \infty$. If it is not the case, the switching from the dynamics of C to the dynamics of an adjacent cell can happen only at time $t \geq t_{min}$. Therefore, the reachable set on the time interval $[0, t_{min}]$ is computed as for a system without switching. After time t_{min}, in each step we compute the intersection of the reachable set with the boundary of the current cell until no new reachable states inside C is found.

Once the computation for the cell C terminates, to propagate the reachable set inside a new cell C', we use Algorithm 2 starting from the intersection of the exit points E with C'.

Convergence result In order to show that our method is convergent, again we consider the approximation error in terms of the Hausdorff distance.

Let $K_g(T, X_0)$ be the reachable set of the nonlinear system (5) (which we want to compute), and let $\hat{K}_s(T, X_0)$ be the set computed by using Algorithm 2 for the piecewise linear system (6), as shown above.

Theorem 2.

$$d_H(\, K_g(T, X_0), \hat{K}_s(T, X_0)\,) \leq \eta(\frac{e^{LT} - 1}{L} + 2\, r\, e^{LT}).$$

The sketch of proof is as follows. The distance between the (exact) reachable sets of (5) and (6) is the bound given in (7). In addition, for the piecewise linear system we can prove that the distance between the approximate set $\hat{K}_s(T, X_0)$ and the exact set $K_s(T, X_0)$ is indeed the error of Algorithm 1, given by Theorem 1. Then, using the triangle inequality we obtain the inequality of Theorem 2. $\diamond$

As we have seen earlier, if g is a C^2 function with a second derivative bound, η is of order $0(h^2)$ where h is the size of the underlying mesh $\mathcal{M}$. Therefore by choosing appropriate time step r (depending on h), we can guarantee a quadratic error bound.

It is worth to mention that the continuity of the approximating systems is key to the convergence results. There are different choices for approximating functions allowing to achieve better convergence. Bilinear interpolation over quadrilaterals may offer a higher order approximation on a well-designed mesh. Another possibility is to use higher degree approximants (such as piecewise quadratic). This, however, requires the ability to deal with more complex autonomous systems.

Simplicial mesh construction We finish this section by a brief discussion on implementation issues. We have shown earlier a bound on the interpolation error which depends on the mesh size. However, it should be noted that the orientation and shape of the mesh may yield an order of magnitude significant improvement in approximation accuracy. The problem of finding an optimal mesh can be formulated as to minimize the interpolation error. However, the optimal meshes may have complex geometric structures which are expensive in storage and computation costs. In this work, we use a simple triangulation which offers important advantages regarding the operations required by our reachability algorithm.

We construct a simplicial mesh by triangulating a underlying rectangular grid. Indeed, a n-dimensional rectangle can be dissected into $n!$ simplices as follows. It can be assumed that the rectangle is a cube $[0,1]^n$. We consider a permutation $\pi = (i1, i2, \ldots, in)$ of $(1, 2, \ldots, n)$ and let S_π be the simplex defined by $0 \leq x_{i1} \leq x_{i2} \ldots \leq x_{in} \leq 1$. It is not hard to see that such $n!$ simplices S_π form a triangulation of the cube. More elaborated schemes allow to obtain a smaller number of simplices [21]. However, the advantage of this method is that it allows a compact representation of the resulting mesh and thus efficient manipulation. Indeed, we need just to store the coordinates of the grid, and all adjacency information (necessary to propagate the reachable set from one cell to another) can be encoded based on the permutations. In addition, an adaptive mesh can be generated on-the-fly during the progress of the reachability computation by considering the derivative variation locally.

5 Experimentation

Our reachability method was implemented and is being integrated in the tool d/dt in order to analyze hybrid systems. We have experimented the method on various examples. We now present some examples for illustrative purposes.

5.1 The Vanderpol equation

The first example is the Vanderpol equation, given below. Here, we are interested in detecting limit cycles of the system.

$$\begin{cases} \dot{x}(t) = y(t) \\ \dot{y}(t) = y(t)(1 - x^2(t)) - x(t). \end{cases}$$

We approximate the system by a piecewise linear interpolating system using a uniform triangular mesh of size $h = 0.05$. We add an input to the latter whichs account for the interpolation error. For a time step $r = 0.05$ and the initial set $X_0 = \{(x,y) | (x-2)^2 + (y-2)^2 \leq 0.25\}$, the reachable set is shown in Figure 1. We can see that the final reachable set (plotted on the right) contains the limit cycle.

5.2 Zermelo's problem

To illustrate the behavior of our algorithm on a nonlinear system with bounded input, we consider a classical problem of optimal control (Zermelo's problem).

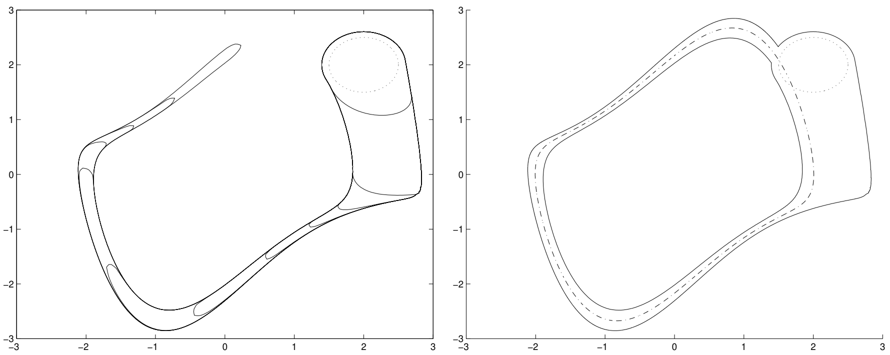

Fig. 1. Left: successive computations of the reachable set. Right: the final reachable set containing the limit cycle.

The dynamics of the system is as follows:

$$\begin{cases} \dot{x}(t) = y(t) - y^2(t) + \quad u_x(t) \\ \dot{y}(t) = \qquad\qquad\qquad u_y(t) \\ \sqrt{u_x(t)^2 + u_y(t)^2} \leq 0.1 \end{cases} \tag{8}$$

We perform the reachability computation with a time step $r = 0.01$ and the result can be seen in Figure 2.

6 Concluding remarks

In this paper, we proposed a framework for approximate reachability analysis of continuous nonlinear systems by means of piecewise linear approximation and developed a reachability algorithm with good convergence rate. This approach can be seen as an application of hybrid systems to deal with complex systems. It also shows a nice interplay between numerical and symbolic computation for safety verification.

The results presented in the paper open various interesting directions for future research. The convergence can be improved by using higher degree approximants, such as piecewise quadratic, and a reachability method for such approximating systems would be of great interest. Additionally, using rectangular meshes reduces significantly the complexity of reachability computation and thus the use of a mixed rectangular-simplicial mesh could allow to achieve a good trade-off between accuracy and computation cost. On the other hand, an important question to address is to find conditions or classes of hybrid system for which the convergence result of our method is preserved.

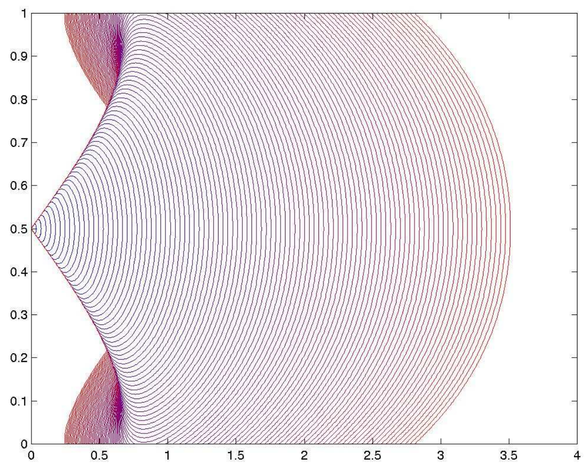

Fig. 2. Successive computations of the reachable set

Acknowledgements We would like to thank the colleagues in LMC and VERIMAG for useful discussions. This work is supported by the European project CC.

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*, 138:3–34, 1995.
2. H. Anai and V. Weispfenning. Reach set computations using real quantifier elimination. In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, LNCS 2034, pages 63–75. Springer-Verlag, 2001.
3. E. Asarin, T. Dang, and O. Maler. d/dt: A tool for verification of hybrid systems. In *Computer Aided Verification*, LNCS. Springer-Verlag, 2002.
4. M.D. Di Benedetto and A. Sangiovanni-Vincentelli. *Hybrid Systems: Computation and Control*. LNCS 2034. Springer-Verlag, 2001.
5. O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, LNCS 1790, pages 73–88. Springer-Verlag, 2000.
6. A. Chutinan and B.H. Krogh. Verification of polyhedral invariant hybrid automata using polygonal flow pipe approximations. In F. Vaandrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control*, LNCS 1569, pages 76–90. Springer-Verlag, 1999.

7. T. Dang and O. Maler. Reachability analysis via face lifting. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, LNCS 1386, pages 96–109. Springer-Verlag, 1998.
8. J. Dieudonné. *Calcul Infinitésimal*. Collection Méthodes. Hermann Paris, 1980.
9. J. Della Dora, A. Maignan, M. Mirica-Ruse, and S. Yovine. Hybrid computation. In *ISSAC'01*, 2001.
10. A. Girard. Approximate solutions of ODEs using piecewise linear vector fields. In *Proc. CASC'02*, 2002.
11. A. Girard. Detection of event occurence in piecewise linear hybrid systems. In *Proc. RASC 02, December 2002, Nottingham, UK*, 2002.
12. M. Greenstreet and C. Tomlin. *Hybrid Systems: Computation and Control*. LNCS. Springer-Verlag, 2002.
13. M.R. Greenstreet and I. Mitchell. Integrating projections. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control*, LNCS 1386, pages 159–1740. Springer-Verlag, 1998.
14. M.R. Greenstreet and I. Mitchell. Reachability analysis using polygonal projections. In F. Vaandrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control*, LNCS 1569, pages 76–90. Springer-Verlag, 1999.
15. L.C.G.J.M. Habets and J.H. van Schuppen. Control of piecewise-linear hybrid systems on simplices and rectangles. In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Control and Computation*, LNCS 2034, pages 261–273. Springer-Verlag, 2001.
16. T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.
17. T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. Analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43:540–554, 1998.
18. J. Hubbard and B. West. *Differential Equations: A Dynamical Systems Approach, Higher-Dimensional Systems*, volume 18 of *Texts in Applied Mathematics*. Springer Verlag, 1995.
19. G. Lafferriere, G. Pappas, and S. Yovine. Reachability computation for linear systems. In *Proc. of the 14th IFAC World Congress*, volume E, pages 7–12, 1999.
20. K. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *Software Tools for Technology Transfert*, 1(1), 1997.
21. T.H. Marshall. Volume formulae for regular hyperbolic cubes. In *Conform. Geom. Dyn.*, pages 25–28, 1998.
22. I. Mitchell and C. Tomlin. Level set method for computation in hybrid systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, LNCS 1790, pages 311–323. Springer-Verlag, 2000.
23. A. Puri and P. Varaiya. Verification of hybrid systems using abstraction. In A. Nerode P. Antsaklis, W. Kohn and S. Sastry, editors, *Hybrid Systems II*, LNCS 999. Springer-Verlag, 1995.
24. A.M. Stuart and A.R. Humphries. *Dynamical Systems and Numerical Analysis*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1996.
25. O. Stursberg and S. Kowalewski. Approximating switched continuous systems by rectangular automata. In *Proc. ECC'99*, 1999.
26. O. Stursberg, S. Kowalewski, and S. Engell. On the generation of timed approximations for continuous systems. *Mathematical and Computer Modelling of Dynamical Systems*, 6(1):51–70, 2000.
27. S. Yovine. Kronos: A verification tool for real-time systems. *Software Tools for Technology Transfer*, 1(1):123–133, 1997.

A Fundamental Inequality Theorem

Theorem 3 (Fundamental Inequality). *Let f be a function with values in $\mathbb{R}^n$, continuous and L-Lipschitz on a set $D \subset R^n$. Let $x_1(t)$ and $x_2(t)$ be functions with values in $\mathbb{R}^n$ continuous, piecewise differentiable on an interval $I \subset \mathbb{R}$ containing* 0 *such that* $\forall t \in I$, $x_1(t) \in D$, $x_2(t) \in D$, *and* $||\dot{x}_1(t) - f(x_1(t))|| \leq \epsilon_1$, $||\dot{x}_2(t) - f(x_2(t))|| \leq \epsilon_2$. *Then,*

$$\forall t \in I,\ ||x_1(t) - x_2(t)|| \leq ||x_1(0) - x_2(0)||e^{L|t|} + \frac{\epsilon_1 + \epsilon_2}{L}(e^{L|t|} - 1).$$

The proof of this theorem can be found in [18] or [8]. This theorem has many applications in the field of numerical analysis of dynamical systems (see [24]).

B Proofs

B.1 Proof of Theorem 1

We start by proving the first property. To show that $K_s(T, X_0) \subseteq R_N$, we first observe that, by definition, $R_N = \bigcup_{1 \leq i \leq N} Q_i$. Hence, it suffices to show that for all $i \in \{0, \ldots, N-1\}$

$$\forall x \in X_0 \;\; u(\cdot) \in \mathcal{U}_\mu \; t \in [ir, (i+1)r] \; \Phi_s(t, x, u(\cdot)) \in Q_{i+1}. \tag{9}$$

We will prove (9) by induction. We begin by the base case ($i = 0$). Let x be an element of X_0, $u(\cdot)$ an admissible control, and $t \in [0, r]$. We denote $y_c = \Phi_s(t, x, u(\cdot))$ and $y_a = \Phi_f(t, x)$. Using the Fundamental Inequality, we have

$$||y_a - y_c|| \leq \frac{\mu}{L}(e^{Lt} - 1) \leq \frac{\mu}{L}(e^{Lr} - 1).$$

It is easy to see that y_a is an element of P_1; therefore y_c is an element of Q_1, which implies that (9) holds for $i = 0$.

We now assume that the formula (9) holds for some $i \geq 0$. Given $x \in X_0$, $u(\cdot) \in \mathcal{U}_\mu$ and $t \in [(i+1)r, (i+2)r]$ we denote $y_c = \Phi_s(t, x, u(\cdot))$ and $z_c = \Phi_s(t - r, x, u(\cdot))$. Since (9) holds for i, we have $z_c \in Q_{i+1}$. Let $y_a = \Phi_f(r, z_c)$. Again, by the Fundamental Inequality,

$$||y_a - y_c|| \leq \frac{\mu}{L}(e^{Lr} - 1).$$

In addition, $y_a \in P_{i+2}$. It then follows that $y_c \in Q_{i+2}$, which shows that (9) holds for $i + 1$. ⋄

Proceeding with the convergence property, we need to show that

$$d_H(K_s(T, X_0), R_N) \leq 2\,\mu\, r\, e^{LT}.$$

We denote by δ_i the Hausdorff semi-distance from the set R_i, computed in iteration i of Algorithm 1, to the corresponding exact set $K_s(ir, X_0)$:

$$\delta_i = \sup_{x \in R_i} \inf_{y \in K_s(ir, X_0)} ||x - y||.$$

To estimate the error bound, we determine the relation between δ_i and δ_{i+1}.

Let x_a^* be an element of R_{i+1} with $i \geq 1$. There are two cases: (1) $x_a^* \in R_i$, and (2) $x_a^* \notin R_i$.

For the first case where $x_a^* \in R_i$, it is not hard to see that there exists x_c^* in $K_s(ir, X_0) \subseteq K_s((i+1)r, X_0)$ such that $||x_a^* - x_c^*|| \leq \delta_i$.

We now consider the second case where $x_a^* \notin R_i$. Then, there exists $y_a^* \in P_{i+1}$ such that

$$||y_a^* - x_a^*|| \leq \frac{\mu}{L}(e^{Lr} - 1). \tag{10}$$

Let z_a^* be the 'predecessor' of y_a^* by the autonomous system such that $y_a^* = \Phi_f(r, z_a^*)$. Since P_{i+1} is obtained by applying Φ_f to the set Q_i, therefore $z_a^* \in Q_i$, which means that $z_a^* \in R_i$. Then, there exists z_c^* in the exact reachable set $K_s(ir, X_0)$ of the system with input such that $||z_a^* - z_c^*|| \leq \delta_i$.

We consider a point x_c^* defined as: $x_c^* = \Phi_s(r, z_c^*, u^*(\cdot))$ where $u^*(\cdot)$ is defined as follows:

$$u^*(t) = \frac{L(x_a^* - y_a^*)}{(e^{Lr} - 1)}$$

It is easy to see that $u^*(\cdot)$ which is an admissible input of the system (3). In other words, x_c^* is the successor of z_c^* by the system with input, and hence x_c^* is an element of $K_s((i+1)r, X_0)$. In the following we will determine δ_{i+1} by estimating an upper bound of $||x_a^* - x_c^*||$.

Let $z_a(t)$ and $z_c(t)$ be the solutions of the following equations:

$$\begin{cases} \dot{z}_a(t) = f(z_a(t)), & z_a(0) = z_a^*, \\ \dot{z}_c(t) = f(z_c(t)) + u^*(t), & z_c(0) = z_c^*. \end{cases}$$

We also define two functions $x_a(t)$ and $x_c(t)$ as follows:

$$\begin{cases} x_a(t) = z_a(t) + \frac{t}{r}(x_a^* - y_a^*), \\ x_c(t) = z_c(t). \end{cases}$$

Using the Fundamental Inequality and the bound μ on the input, we have

$$\begin{aligned} ||z_a(t) - z_c(t))|| &\leq ||z_a^* - z_c^*||e^{Lt} + \frac{\mu}{L}(e^{Lt} - 1) \\ &\leq \delta_i \, e^{Lt} + \frac{\mu}{L}(e^{Lt} - 1). \end{aligned} \tag{13}$$

On the other hand, since $z_a(r) = y_a^*$ and $z_c(r) = x_c^*$, we can write:

$$x_a^* - x_c^* = x_a(r) - x_c(r) = (x_a(0) - x_c(0)) + \int_0^r (\dot{x}_a(s) - \dot{x}_c(s))ds.$$

In addition, $||x_a(0) - x_c(0)|| = ||z_a^* - z_c^*|| \leq \delta_i$. Therefore,

$$||x_a^* - x_c^*|| \leq \delta_i + ||\int_0^r (\dot{x}_a(s) - \dot{x}_c(s))ds||. \tag{14}$$

We now focus on the term inside the integral of (14).

$$||\dot{x}_a(s) - \dot{x}_c(s)|| = ||f(z_a(s)) + \frac{(x_a^* - y_a^*)}{r} - f(z_c(s))||$$

Since f is L-Lipschitz, $||f(z_a(s)) - f(z_c(s))|| \leq L\,||z_a(s) - z_c(s)||$. Using (10) and (13) yields

$$\begin{aligned}||\dot{x}_a(s) - \dot{x}_c(s)|| &\leq \frac{||x_a^* - y_a^*||}{r} + L\,||z_a(s) - z_c(s)|| \\ &\leq \frac{\mu}{Lr}(e^{Lr} - 1) + L\,(\delta_i\,e^{Ls} + \frac{\mu}{L}(e^{Ls} - 1)).\end{aligned}$$

Combining the above inequality with (14) and developing the integral gives

$$\delta_{i+1} \leq \delta_i\,e^{Lr} + \frac{2\,\mu}{L}\,(e^{Lr} - 1 - L\,r).$$

Observe that $e^{Lr} - 1 - Lr \leq \frac{1}{2}\,e^{Lr}\,(Lr)^2$, then

$$\delta_{i+1} \leq \delta_i\,e^{Lr} + \mu\,L\,e^{Lr}\,r^2,$$

which leads to

$$\begin{aligned}\delta_N &\leq \delta_1\,e^{L(N-1)r} + \mu\,L\,e^{Lr}\,r^2 \sum_{i=0}^{i=N-2} e^{iLr} \\ &\leq \delta_1\,e^{L(N-1)r} + \mu\,L\,e^{Lr}\,r^2\,\frac{e^{L\,(N-1)\,r} - 1}{e^{Lr} - 1}.\end{aligned}$$

Since $e^{Lr} - 1 \geq Lr$ and, in addition, we can prove that $\delta_1 \leq \mu\,r\,e^{Lr}$. The above thus leads to $\delta_N \leq 2\,\mu\,r\,e^{LT}$. This completes the proof of the theorem. ■

B.2 Proof of Lemma 2

We first estimate an upper bound of $||g(x) - l_k(x)||$ for all points x inside a cell $C_k \in \mathcal{M}$. Let v be a vertex of C_k. By triangle inequality, we have

$$||g(x) - l_k(x)|| \leq ||g(x) - g(v)|| + ||g(v) - l_k(x)||.$$

By definition, $g(v) = l_k(v)$. In addition, g is L-Lipschitz, it then follows that

$$||g(x) - l_k(x)|| \leq ||g(x) - g(v)|| + ||l_k(v) - l_k(x)|| \leq 2\,L||x - v||. \tag{15}$$

Note that the above inequality holds for any vertex $v \in V(C_k)$. In order to get a tight upper bound on $||g(x) - l_k(x)||$, we can estimate a bound on $||x - v||$ for each vertex v and then choose the smallest bound. Let $V(C_k) = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of C_k. A point $x \in C_k$ can be written as:

$$\begin{cases} x = \sum_{i=1}^{n} \alpha_i \, v_i \\ \sum_{i=1}^{n} \alpha_i = 1 \text{ and } \ \forall i \in \{1, \ldots, n\} \ \alpha_i \geq 0. \end{cases} \tag{16}$$

We observe, from the conditions (16), that there exists $j \in \{1, \ldots, n\}$ such that $\alpha_j \geq \frac{1}{n+1}$. Since $\sum_{i=1}^{n} \alpha_i v_j = v_j$, we can write $x - v_j = \sum_{i=1, i \neq j}^{n} \alpha_i (v_i - v_j)$. Additionally, $||v_i - v_j|| \leq h(C_k)$; therefore,

$$||x - v_j|| \leq h(C_k) \sum_{i=1, i \neq j}^{n} \alpha_i = h(C_k)\,(1 - \alpha_j) \leq h(C_k) \frac{n}{n+1}. \tag{17}$$

Using this bound in (15) we get

$$\forall x \in C_k \ \ ||g(x) - l_k(x)|| \leq 2\, h(C_k) \frac{n\,L}{n+1}.$$

Note that for all cells $C_k \in \mathcal{M}$, $h(C_k) \leq h(\mathcal{M}) = h$. Hence, we obtain

$$\forall x \in \mathcal{X} \ \ ||g(x) - l(x)|| \leq 2\, h \frac{n\,L}{n+1},$$

which completes the proof. ■

B.3 Proof of Lemma 3

For a given cell $C_k \in \mathcal{M}$, we define the function $e(x) = g(x) - l_k(x)$ and let $x^* = \arg\max_{x \in C_k} ||e(x)||$ (note that the simplex C_k is compact). Let v be a vertex of C_k, and all points in the line segment connecting x^* and v can be written as:

$$x(\gamma) = x^* + \gamma\,(v - x^*), \ \ \gamma \in [0, 1]. \tag{18}$$

To determine a bound on $e(x^*)$, we define a function $z(\gamma) = e(x(\gamma))$ for $\gamma \in [0, 1]$. Expanding z with respect to γ gives

$$z(1) = z(0) + \frac{dz}{d\gamma}(0) + \int_0^1 \frac{d^2 z}{d\gamma^2}(s)\,(1 - s) ds \tag{19}$$

The i^{th} coordinate of a point $y \in \mathbb{R}^n$ is denoted by y_i. We can see from (18) that $dx_i/d\gamma = (v_i - x_i^*)$. Then,

$$\frac{dz}{d\gamma}(\gamma) = \sum_{i=1}^{n} \frac{\partial e}{\partial x_i}(x(\gamma))\,(v_i - x_i^*)$$
$$\frac{d^2 z}{d\gamma^2}(\gamma) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 e}{\partial x_i \partial x_j}(x(\gamma))\,(v_i - x_i^*)\,(v_j - x_j^*)$$

The second partial derivatives $\partial^2 l_k / \partial x_i \partial x_j$ vanish for all $i, j \in \{1, 2, \ldots, n\}$, hence

$$\frac{d^2 z}{d\gamma^2}(\gamma) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 g}{\partial x_i \partial x_j}(x(\gamma))\,(v_i - x_i^*)\,(v_j - x_j^*)$$

Similar to the inequality (17) established in the proof of Lemma 2, we can show that there exists $v \in V(C_k)$ such that $\forall i \in \{1, \ldots, n\}$ $||v_i - x_i^*|| \leq h(C_k)n/(n+1)$. Then, using the bound K on the second derivatives of the function g, we obtain

$$||\frac{d^2 z}{d\gamma^2}(\gamma)|| \leq (h(C_k))^2 \frac{n^2\,K}{(n+1)^2}$$

In addition, $||e(x^*)||$ is maximum, which implies that $\frac{dz}{d\gamma}(0) = 0$. By definition of the interpolating function, $g(v) = l_k(v)$, then $z(1) = 0$. Therefore, (19) becomes

$$g(x^*) - l_k(x^*) + \int_0^1 \frac{d^2 z}{d\gamma^2}(s)\,(1-s)ds = 0.$$

Using (20), it then follows that

$$||g(x^*) - l_k(x^*)|| \leq (h(C_k))^2 \frac{n^2\,K}{(n+1)^2} \int_0^1 (1-s)ds = (h(C_k))^2 \frac{n^2\,K}{2\,(n+1)^2}.$$

Finally, considering all cells in the mesh $\mathcal{M}$, we can easily deduce

$$\forall x \in \mathcal{X}\ ||g(x) - l(x)|| \leq h^2 \frac{n^2\,K}{2\,(n+1)^2},$$

which yields the result of the lemma. ∎