REACHABILITY COMPUTATION FOR UNCERTAIN PLANAR AFFINE SYSTEMS USING LINEAR ABSTRACTIONS

Othman Nasri
Marie-Anne Lefebvre
Hervé Guéguen

Supélec - IETR
BP 81127 Cesson-Sévigné Cedex, France

Abstract: Reachability computation is the central problem arising in the verification of hybrid or continuous systems. One approach, among others, to compute an over approximation of the reachable space is to split the continuous state space and to abstract the continuous dynamics in each resulting cell by a linear differential inclusion for which the reachable space may be computed with polyhedra. A previous work proposed to use characteristics of the affine continuous dynamics to guide the polyhedral partition. This paper presents an extension of this approach to uncertain planar systems where one parameter of the model may take its value in a polytope. It is shown that the result for all values of the parameter may be deduced from the computation for a finite number of values. An algorithm that performs the reachability computation and determines the minimum number of values of the parameter required at each step is proposed and exemplified.

Keywords: Hybrid systems, reachability, abstractions

1. INTRODUCTION

Reachability computation is the central problem in the verification of hybrid or continuous systems (Guéguen and Zaytoon, 2004) and has become a major research issue in hybrid systems. Most approaches to solve this problem are based on a combination of numerical integration and geometrical algorithms (Girard*, et al.*, 2006; Henzinger*, et al.*, 2000). However it is also possible to use hybridization methods to perform this computation. The basic idea, introduced by (Henzinger*, et al.*, 1998), consists in splitting the continuous state space into cells and abstracting the continuous dynamics in each cell, by a linear differential inclusion for which the reachable space may be computed with polyhedra (Frehse, 2005). One key point is then to find a trade off between the number of cells that are introduced and the accuracy of the over-approximation. The choice of the hyperplanes that define the cells is also important and it is possible to use structural properties of the continuous dynamics to guide this choice (Lefebvre and Guéguen, 2006).

For affine systems, defined by equation (1), it is then possible (Lefebvre and Guéguen, 2006) to use left eigenvectors of matrix $\mathbf{A}$ to define the hyperplanes that split the continuous regions. This approach leads to interesting results but is limited to systems where the model is exactly known.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \qquad (1)$$

This paper is a first step towards the extension of the proposed approach to models where the parameter **b** of equation (1) is unknown and takes its values in a polytope. As a first attempt, it only considers planar systems where matrix **A** is non singular. In the next section, the approach for certain planar systems is briefly presented in order to illustrate the basic ideas. The principles and important properties for uncertain systems are presented in section 3. The proposed algorithm to compute an over-approximation of the reachable space is then presented in section 4. Finally the application of the algorithm to an example is shown in section 5.

2. ABSTRACTION BASED REACHABILITY FOR PLANAR REGULAR CERTAIN SYSTEMS

The dynamics of the systems considered in this section is specified by equation (1) where the dimension of the state space is two, matrix $\mathbf{A}$ is not singular and vector $\mathbf{b}$ is perfectly known. It is then possible to define the equilibrium point of the system by equation (2).

$$\mathbf{x}_e = \mathbf{A}^{-1}\mathbf{b} \tag{2}$$

This dynamics is associated with a polytopic region of the state space, denoted *Inv*, and the aim of the reachability calculation is to compute the set of points of this region that may be reached by the dynamics, from a given region denoted *Init*, namely the set:

$$\left\{\mathbf{x} / \exists \mathbf{x}_0 \in Init, \exists t \geq 0 \; s.a. \; \mathbf{x} = \Phi(t) \; with \; \Phi(0) = \mathbf{x}_0 \right. \\ \left. and \; \forall \tau \leq t \; \dot{\Phi}(\tau) = \mathbf{A}\Phi(\tau) + \mathbf{b} \; and \; \Phi(\tau) \in Inv \right\} \tag{3}$$

The initial region *Init* will be considered as a polytope and it is then possible to compute the reachable space as the convex hull of the reachable space from each of its vertices. As it is difficult to have an explicit representation of the set defined by (3), hybridization methods aim at abstracting the continuous dynamics by polytopic differential inclusions on cells defined by splitting the region *Inv* with hyperplanes.

When applied to the category of systems considered in this paper, the method described in (Lefebvre and Guéguen, 2006) leads to consider families of hyperplanes, orthogonal to some vectors $\{\mathbf{q}_l\}$ generated by linear combinations of two left eigenvectors of matrix $\mathbf{A}$ if they are real, or two given vectors otherwise. Vectors $\{\mathbf{q}_l\}$ are chosen and ordered in such a way that each cell is defined by the intersection of the region *Inv* with the sector defined by (4).

$$\mathbf{q}_i{}^T(\mathbf{x}-\mathbf{x}_e) \geq 0 \wedge \mathbf{q}_{i+1}{}^T(\mathbf{x}-\mathbf{x}_e) \leq 0 \tag{4}$$

It is then possible to characterize the vector field in each point of this cell by (5), which can be used to abstract the dynamics. In (5), vectors $\boldsymbol{\gamma}$ are defined with respect to the vector field on the boundaries of the cell and are characterized by (6).

$$\boldsymbol{\gamma}_i{}^T\dot{\mathbf{x}} \geq 0 \wedge \boldsymbol{\gamma}_{i+1}^T\dot{\mathbf{x}} \leq 0 \tag{5}$$

$$\boldsymbol{\gamma}_i = \left(\mathbf{A}^T\right)^{-1}\mathbf{q}_i \tag{6}$$

The computation of the reachable space from a point $\mathbf{x}_0$ within the cell is then straightforward and its result is given by the conjunction of (4) and (7).

$$\boldsymbol{\gamma}_i{}^T(\mathbf{x}-\mathbf{x}_0) \geq 0 \wedge \boldsymbol{\gamma}_{i+1}^T(\mathbf{x}-\mathbf{x}_0) \leq 0 \tag{7}$$

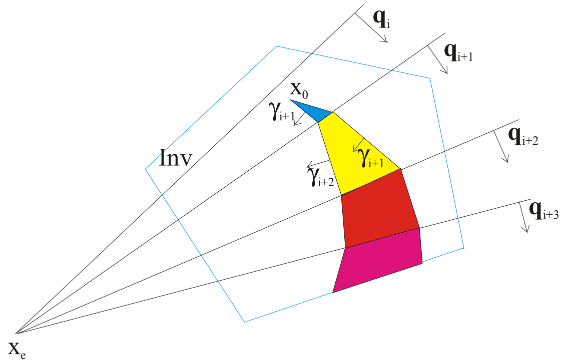


Fig 1. First steps of the reachable space computation with abstraction

Another simplification is induced by the choice of the separating hyperplanes: if the vector $\mathbf{q}_i$ is not a left eigenvector, the boundary it defines is crossed in only one direction, and if $\mathbf{q}_i$ is a real left eigenvector, then the corresponding hyperplane cannot be crossed. It is then possible, to choose the vectors $\{\mathbf{q}_l\}$ so that $\mathbf{q}_i$ characterizes the boundary through which the state trajectories enter the cell and $\mathbf{q}_{i+1}$ the one through which they leave it. So, state trajectories evolve from the cell specified by $(\mathbf{q}_i, \mathbf{q}_{i+1})$ to the one specified by $(\mathbf{q}_{i+1}, \mathbf{q}_{i+2})$. The first steps of the reachability computation are illustrated in figure 1. The intersection of the polyhedron defined by (7) with the outgoing boundary (specified by $\mathbf{q}_{i+1}$) has two vertices that are used as starting points for the computation within the next cell.

3. BASIC PRINCIPLES FOR UNCERTAIN SYSTEMS

It is now assumed that the dynamics of the system is still characterized by equation (1) but that vector $\mathbf{b}$ is unknown and may be characterized by equation (8), where $\mathbf{b}_0$ and $\mathbf{b}_1$ are 2 known vectors and α is the unknown parameter. It is also supposed that the value of this parameter is fixed. The problem is then to find a method to compute an over-approximation of the reachable space from an initial point whatever the value of α is.

$$\mathbf{b}_\alpha = (1-\alpha)\mathbf{b}_0 + \alpha\mathbf{b}_1 \; with \; \alpha \in [0,1] \tag{8}$$

When the system is characterized by equations (1) and (8), it is possible to associate an equilibrium point to each value of α. This point may be computed from the equilibrium points associated to $\mathbf{b}_0$ and $\mathbf{b}_1$ according to equation (9).

$$\mathbf{x}_{e_\alpha} = (1-\alpha)\mathbf{x}_{e_0} + \alpha\mathbf{x}_{e_1} \tag{9}$$

It is then possible to associate to each value of α the set of cells defined by the set of vectors $\{\mathbf{q}_l\}$, that are the intersection of the region *Inv* with the region characterized by (10). From now on, each of these cells that depends on the value of α and on the pair $(\mathbf{q}_i, \mathbf{q}_{i+1})$ will be denoted by $S_{i,\alpha}$ and its boundaries by $I_{i,\alpha}$ and $I_{i+1,\alpha}$ (11).

$$\mathbf{q}_i^T(\mathbf{x}-\mathbf{x}_{e_\alpha}) \geq 0 \wedge \mathbf{q}_{i+1}^T(\mathbf{x}-\mathbf{x}_{e_\alpha}) \leq 0 \qquad (10)$$

$$I_{i,\alpha} = \{\mathbf{x} / \mathbf{q}_i^T(\mathbf{x}-\mathbf{x}_{e_\alpha}) = 0\} \qquad (11)$$

For each point of the cell $S_{i,\alpha}$, the vector field may be abstracted by a differential inclusion in the region, characterized by equation (5), that does not depend on the value of α but only on the pair $(\mathbf{q}_i, \mathbf{q}_{i+1})$. The basic principle of the algorithm proposed below is based on the property that the differential inclusion associated to a cell $S_{i,\alpha}$ does not depend on the value of α but only on the index i.

It is then possible to express three properties that are useful to design an algorithm to compute an over-approximation of the reachable space.

Property 1. If α_0 and α_1 are such that, for all values of α between α_0 and α_1, an incoming point $\mathbf{x}_\alpha$ within the region $S_{i,\alpha}$ is aligned with the incoming points $\mathbf{x}_0$ and $\mathbf{x}_1$ within the regions S_{i,α_0} and S_{i,α_1}, the relative outgoing points of the reachable space within $S_{i,\alpha}$ are also aligned with the relative outgoing points of S_{i,α_0} and S_{i,α_1}.

This property can be illustrated by figure 2 and proved by considering that it is possible to find $\beta \in [0,1]$ such that $\alpha = \beta\alpha_0 + (1-\beta)\alpha_1$. As $\mathbf{x}_0 \in I_{i,\alpha_0}$, $\mathbf{x}_\alpha \in I_{i,\alpha}$ and $\mathbf{x}_1 \in I_{i,\alpha_1}$, it is possible to deduce from (11) and (9) that if $\mathbf{x}_0$, $\mathbf{x}_\alpha$ and $\mathbf{x}_1$ are aligned then

$$\mathbf{x}_\alpha = \beta\mathbf{x}_0 + (1-\beta)\mathbf{x}_1$$

Then it is possible to compute the outgoing point by $\mathbf{q}_{i+1}^T(\mathbf{y}_\alpha - \mathbf{x}_{e_\alpha}) = 0 \wedge \gamma_i^T(\mathbf{y}_\alpha - \mathbf{x}_\alpha) = 0$ and check that the solution is given by $\mathbf{y}_\alpha = \beta\mathbf{y}_0 + (1-\beta)\mathbf{y}_1$ where $\mathbf{y}_0$ and $\mathbf{y}_1$ are the relative outgoing points of sectors S_{i,α_0} and S_{i,α_1}.

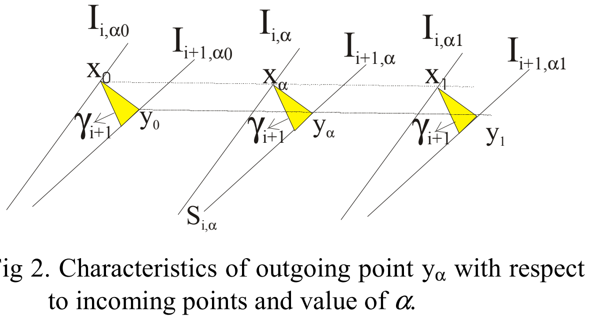


Fig 2. Characteristics of outgoing point y_α with respect to incoming points and value of α.

Property 2. If α_0 and α_1 and $(\mathbf{q}_i, \mathbf{q}_{i+1})$ are such that the assumption of property 1 is verified, then the over approximation of the reachable space from the set of entering points $\mathbf{x}_\alpha$ is

$$\bigcup_{\alpha\in[\alpha_0,\alpha_1]} A_{i,\alpha}(\mathbf{x}_\alpha) = convexhull(A_{i,\alpha_0}(\mathbf{x}_0), A_{i,\alpha_1}(\mathbf{x}_1)) \qquad (12)$$

where $A_{i,\alpha}(\mathbf{x}_\alpha)$ is characterized by the conjunction of (10) and (13).

$$\gamma_i^T(\mathbf{x}-\mathbf{x}_\alpha) \geq 0 \wedge \gamma_{i+1}^T(\mathbf{x}-\mathbf{x}_\alpha) \leq 0 \qquad (13)$$

This property is directly deduced from property 1. It is very important because it allows to compute the union of reachable region for a continuous variation of the parameter α from the computation for two values.

Property 3. If α_0 and α_1 are such that $\mathbf{x}_{init} \in S_{i,\alpha_0} \cap S_{i,\alpha_1}$ then $\forall\alpha \in [\alpha_0, \alpha_1]$ $\mathbf{x}_{init} \in S_{i,\alpha}$ and the vertices of the intersection of $A_{i,\alpha}(\mathbf{x}_{init})$ with $I_{i+1,\alpha}$ are aligned for all values of α between α_0 and α_1.

The first part of this property is proved by considering that if $\alpha = \beta\alpha_0 + (1-\beta)\alpha_1$ then $\mathbf{q}_i^T(\mathbf{x}_{init} - \mathbf{x}_{e\alpha}) = \beta\mathbf{q}_i^T(\mathbf{x}_{init} - \mathbf{x}_{e\alpha_0}) + (1-\beta)\mathbf{q}_i^T(\mathbf{x}_{init} - \mathbf{x}_{e\alpha1})$ is positive and that $\mathbf{q}_{i+1}^T(\mathbf{x}_{init} - \mathbf{x}_{e\alpha})$ is negative with equivalent considerations.

The second part of the property is illustrated in figure 3 and is proved by considering that these vertices are the intersection of $\gamma_i^T(\mathbf{x}-\mathbf{x}_{init}) = 0$ (or $\gamma_{i+1}^T(\mathbf{x}-\mathbf{x}_{init}) = 0$), that does not depend on α, with $I_{i+1,\alpha}$.

So if α_0 and α_1 are such that the assumption of property 3 is verified, then the incoming points in the next region $S_{i+1,\alpha}$ are all aligned and property 2 can be used to compute the reachable region within this set of cells. This can be iterated as long as the considered regions do not intersect the boundary of *Inv* and leads to the algorithm presented in the next section.

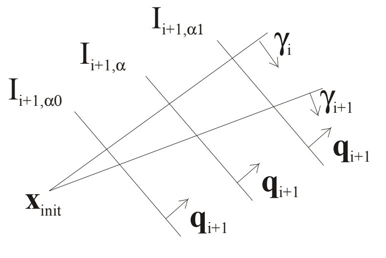


Fig 3. Initialisation

4. REACHABILITY ALGORITHM

This algorithm aims at computing an over approximation of the reachable space from an initial point that is valid for all values of the unknown parameter. It is based on properties 2 and 3 of the previous section that allow to compute the global set with a continuous variation of the parameter from the computation for a finite number of values of this parameter. It is summarized in algorithm 1.

Algorithm 1: Global reachability calculus

Input: (**A**, $\mathbf{b}_0$, $\mathbf{b}_1$) continuous dynamics with the domain of the unknown parameter $\mathbf{b}$, *Inv* the considered region of the state space, $\mathbf{x}_{init}$ the initial point,

Result: $R(\mathbf{x}_{init})$ the over approximation of the reachable space from $\mathbf{x}_{init}$, within the region *Inv,* with respect to the dynamics constraint and valid for all the values of the parameter.

Step 1: decomposition of the region of interest.

Step 2: computation of the initial value of the index *i* of the cells $S_{i,\alpha}$ and of the initial set of values for the parameter α and initialisation of incoming points.

Repeat

Step 3: computation of the reachable region within the set of cells $S_{i,\alpha}$ for the index *i*: $\bigcup_k R_{i,k}$.

Step 4: computation for the next iteration of the index of cells $S_{i,\alpha}$, of the set of pertinent values of the parameter α, and of the incoming points.

Until *end*

4.1 Initialization phase

The initialization of the calculus consists in the two first steps of algorithm 1. The first step may be summarized by the interface of algorithm 2. The details for this step may be found in (Lefebvre and Guéguen, 2006). The important point is that the vectors are chosen and the set $\{\mathbf{q}_l\}$ is ordered so that cells are characterized by (10) and $\mathbf{q}_i$ characterizes the border of cell $S_{i,\alpha}$ through which trajectories enter the cell.

Algorithm 2: decomposition

Input: (**A**, $\mathbf{b}_0$, $\mathbf{b}_1$), *Inv,*

Result: $\{\mathbf{q}_l\}$ the set of vectors defining the cells, $\{\gamma_l\}$ the set of vectors defining the differential inclusions and $(\mathbf{x}_{e0}, \mathbf{x}_{e1})$ the equilibrium points associated to $\mathbf{b}_0$ and $\mathbf{b}_1$.

The step 2 of the algorithm is based on property 3 of the previous section. It is summarized by algorithm 3.

Algorithm 3: initialisation of the loop

Input: $\{\mathbf{q}_l\}$, $(\mathbf{x}_{e0}, \mathbf{x}_{e1})$, $\mathbf{x}_{init}$,

Result: V_alp_init the ordered set of values of the parameter α that will have to be considered, the initial index *i, V_alp* the initial set of values of the parameter to be considered in the next step, for each value of the parameter the associated incoming points.

Step 1: computation of *V_alp_init*.

Step 2: for each value of *V_alp_init*, computation of the index *j* such that $\mathbf{x}_{init} \in S_{j\alpha}$ and initialisation of the index *i* to the minimum of these indexes.

Step 3: choice of values of *V_alp_init* such that $\mathbf{x}_{init} \in S_{i\alpha}$ to initialise *V_alp* and for each value of *V_alp* initialisation of the incoming points $\mathbf{p}_{1,\alpha}$ and $\mathbf{p}_{2,\alpha}$ to $\mathbf{x}_{init}$.

Step one consists in computing for each vector $\mathbf{q}_l$ the value of $\alpha \in [0,1]$, if it exists, such that $x_{init} \in I_{l\alpha}$ and to add this value to *V_alp_init* that is initialized to $\{0,1\}$. At the end of this step, two consecutive values in *V_alp_init* verify assumption of property 3.

4.2 Iterative phase

The first step of the iterative phase (step 3) is the computation of the reachable region within the cells. Firstly, for each value of *V_alp*, the region $A_{i,\alpha}$ characterized by (14), is computed. Then, property 2 is used to compute for each pair (α_k, α_{k+1}) of consecutive values in *V_alp,* $R_{i,k}$ (15) the reachable region within $S_{i,\alpha}$ for $\alpha \in [\alpha_k, \alpha_{k+1}]$. Finally the value of $R(\mathbf{x}_{init})$ after this iteration is computed (16).

$$A_{i,\alpha} = convexhull(A_{i,\alpha}(\mathbf{p}_{1,\alpha}), A_{i\alpha}(\mathbf{p}_{2,\alpha})) \quad (14)$$

$$R_{i,k} = convex\_hull(A_{i,\alpha_k}, A_{i,\alpha_{k+1}}) \cap Inv \quad (15)$$

$$R(\mathbf{x}_{init}) = R(\mathbf{x}_{init}) \cup \left(\bigcup_k R_{i,k} \right) \quad (16)$$

Step 4 is the preparation for the next iteration and may be summarized by algorithm 4.

Algorithm 4: preparation for the next iteration

Input: *V_alp,* $\{A_{i,\alpha}\}$ the set of reachable regions for each value in *V_alp* , $\mathbf{q}_{i+1}$, *Inv*, *V_alp_init.*

Result: *i* the index of the cells, *V_alp* the set of values of the parameter, and for each value the associated incoming points, $\mathbf{p}_{1,\alpha}$ and $\mathbf{p}_{2,\alpha}$.

Step 1: for each element of *V_alp,* computation of $O_\alpha = A_\alpha \cap I_{i+1,\alpha}$ the set of possible outgoing points.

Step 2: for all pairs (α_k, α_{k+1}) of consecutive elements of *V_alp* computation of Out_k the convex hull of O_{α_k} and $O_{\alpha_{k+1}}$.

Step 3: for all regions Out_k computation of Int_k the intersection with the region Inv.

If Int_k=Out_k nothing

If Int_k=∅ the relative values of α are deleted from V_alp

else computation of the values of α such that the vertices of Int_k belongs to $I_{i+1,\alpha}$ and insertion of these values in V_alp.

Step 4: if $\mathbf{q}_{i+1}$ is a left eigenvector, setting of V_alp to empty. Increment of index i.

Step 5: computation for all elements of V_alp of the incoming points $(p_{1,\alpha}, p_{2,\alpha})$ in the cell $S_{i,\alpha}$.

Step 6: if $\mathbf{x}_{init} \in S_{i\alpha}$ for some value of V_alp_init, insertion of the relevant value to V_alp and setting of the associated incoming points to $\mathbf{x}_{init}$.

Step 7: for each three-tuple $(\alpha_j, \alpha_{j+1}, \alpha_{j+2})$ of consecutive values in V_alp, if the 3 points $(p_{1,\alpha_j}, p_{1,\alpha_{j+1}}, p_{1,\alpha_{j+2}})$ are aligned, and so are the points $(p_{2,\alpha_j}, p_{2,\alpha_{j+1}}, p_{2,\alpha_{j+2}})$, then deletion of the value α_{j+1} from V_alp.

Step 3 of this algorithm 4 is central in the approach because it ensures that at each step all the pairs of consecutive elements of the set V_alp verify the assumptions of property 2. The calculus for continuous variation of α from the computation for a finite number of values of this parameter, performed at step 4 of the global algorithm, is then valid. The case when the intersection Int_k is neither empty nor equal to Out_k is illustrated in figure 4. As the outgoing domain Out_k intersects the boundary of the region Inv, in the next step the incoming points for all values of α between α_1 and α_2 will not be aligned. A new value α* is then introduced, such that for all α between α_1 and α* on the one hand, and between α* and α_2 on the other hand, the incoming points are aligned and property 2 can be used in the next step. Step 7 of algorithm 4 ensures that the set of values V_alp is minimal as it deletes intermediate values that are not mandatory to guaranty this line condition.

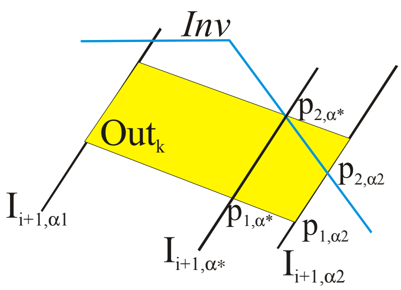


Fig 4. Intersection of the outgoing domain with the region Inv

The loop is stopped when the set V_alp is empty or when the global estimation of the reachable space does not evolve any more but of course there is no guaranty that it will really stop.

5. EXAMPLE

In order to illustrate this algorithm, its application to the computation of the reachable space from the initial point $x_{init}^T = [4 \quad 5]$ for the system specified by the following values is considered. For this system, matrix $\mathbf{A}$ has two real left eigenvectors that are used to generate the decomposition and 15 vectors $\mathbf{q}_i$ are considered.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix},\ \mathbf{b}_0 = \begin{bmatrix} 0 \\ 4 \end{bmatrix},\ \mathbf{b}_1 = \begin{bmatrix} -1 \\ 19 \end{bmatrix}$$

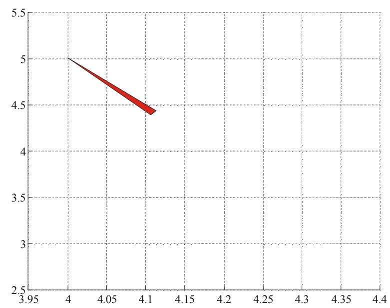
$$Inv: \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 3 & -2 \end{bmatrix} \mathbf{x} \le \begin{bmatrix} 10 \\ -10 \\ 10 \\ -1 \\ 15 \end{bmatrix}$$

Step 2 of algorithm 1 then computes that the algorithm is initialized with:

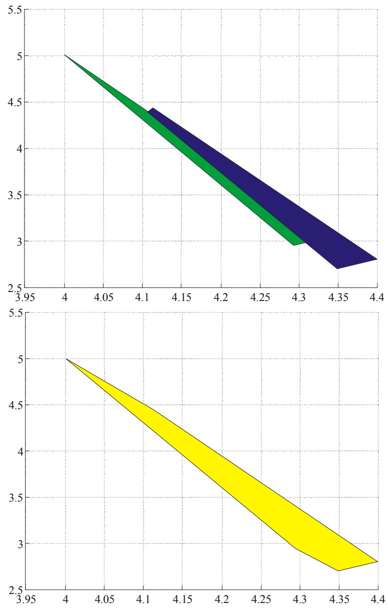
$$V\_alp\_init = \{0, 0.25, 0.9077, 1\}$$
$$i = 4$$
$$V\_alp = \{0.9077, 1\}$$

The first two iterations of the reachability calculus are shown in figure 5. The results for the first iteration is shown in figure 5.a. For this step, the considered cells are $S_{4,\alpha}$ for the 2 values of the parameter α. As for $\alpha = 0.9077$, $\mathbf{x}_{init}$ belongs to the outgoing border of the cell, the reachable space within this cell is only this point and the global reachable space is computed for $\alpha = 1$. For the second iteration, the considered cells are $S_{5,\alpha}$ and there are 3 values for α. For $\alpha = 0.25$ the reachable space is one point, for $\alpha = 0.9077$ it is computed from $\mathbf{x}_{init}$, and for $\alpha = 1$ it is computed from the vertices stemming from the previous iteration. The polytopes A_α for this iteration are represented in figure 5.b and the reachable region for all values of α between 0.25 and 1 (R_1) is drawn in figure 5.c.

The result of the global computation is displayed in figure 6. In this case, the global computation stops and the reachable region is bounded by a line defined by the equilibrium points of the system for all possible values of α.



a)



b)

c)

Fig. 5. First iterations of the reachability Computation

The sets of values of α that are considered at each iteration of the calculus are summarized in table 1. Three phases may be seen. In a first phase (iterations 1, 2, 3) at each iteration a new value from the initial set computed at step 2 of the global algorithm is added. During the second phase (iterations 4, 5, 6), the set does not change. Then, at the end of iteration 6, the boundary of the region *Inv* is crossed for some values of α as shown in figure 7. The values of α associated to the vertices of Int_1 and Int_2 are then introduced by step 3 of algorithm 4. From this iteration to the end of the computation, the algorithm adapts the values of α according to this step 3 and to step 7 of algorithm 4.

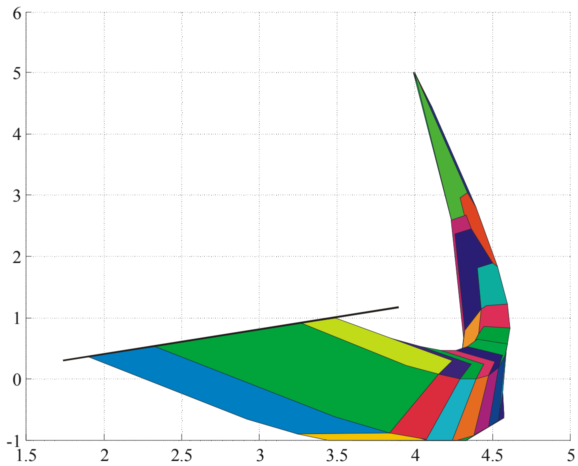


Fig. 6. Global reachable region

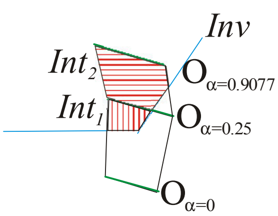


Fig. 7 Details of the first intersection with the boundary of *Inv*.

Table 1 Values of parameters α with respect to the calculus index

1	0.9077	1				
2	0.25	0.9077	1			
3	0	0.25	0.9077	1		
4	0	0.25	0.9077	1		
5	0	0.25	0.9077	1		
6	0	0.25	0.9077	1		
7	0.1275	0.1667	0.25	0.3653	0.9077	1
8	0.2971	0.3725	04987	0.9077	1	
9	0.3611	0.4848	0.5231	0.9077	1	
10	0.3611	0.5329	0.9077	1		
11	0.3611	0.5329	09077	1		
12	0.3611	0.5329	0.9077	1		
13	0.3611	0.5329	1			

CONCLUSION

In this paper, a method for reachability analysis of uncertain affine systems has been presented. It extends a previous work which allows the reachable space to be easily computed when the dynamics of the system under study is completely known. This approach also proved to be suited to uncertain systems as it allows to compute the over-approximation of the reachable space for uncertain systems where a parameter takes its values in a continuous space using a finite number of particular values. The set of these values is adapted at each iteration in order to limit the over-approximations of the reachable space to those linked, on the one hand, to uncertainty and, on the other hand, to the abstraction technique.

In this work, it has been assumed that the unknown parameter was fixed. Future work will consider reachability computation when this parameter may evolve with time, for example to take into account systems with bounded inputs (Girard*, et al.*, 2006).

REFERENCES

Frehse, G. (2005). PHAver: Algorithmic verification of hybrid systems past HyTech. in *HSCC2005: Hybrid Systems Computation and Control* (M. Morari and L. Thiele (Eds.))
Girard, A., C. Le Guernic and O. Maler (2006). Efficient computation of reachable sets of linear time-invariant systems with imputs. in *HSCC 2006:Hybrid Systems Computation and Control*
Guéguen, H. and J. Zaytoon (2004). On the formal verification of hybrid systems. *Control Engineering Practice,* **12,** 1253-1267
Henzinger, T., P. Ho and H. Wong-Toi (1998). Algorithmic analysis of non-linear hybrid systems. *IEEE Trans on Automatic Control,* **43,** 540-554
Henzinger, T., B. Horowitz, M. R. and W.-T. H. (2000). Beyond Hytech: Hybrid systems analysis using interval numerical methods. in *HSCC2000: Hybrid Systems Computation and Control* (N. Lynch and B. Krogh (Eds.))
Lefebvre, M.-A. and H. Guéguen (2006). Hybrid Abstraction of Affine Systems. *Nonlinear Analysis: Hybrid Systems and Applications, Elsevier,* doi:10.1016/j.na.2005.12.016**,**