Zonotope/Hyperplane Intersection for Hybrid Systems Reachability Analysis

Antoine Girard[1] and Colas Le Guernic[2]

[1] Laboratoire Jean Kuntzmann, Université Joseph Fourier
Antoine.Girard@imag.fr
[2] VERIMAG, Université Joseph Fourier
Colas.Le-Guernic@imag.fr

Abstract. In this paper, we are concerned with the problem of computing the reachable sets of hybrid systems with (possibly high dimensional) linear continuous dynamics and guards defined by switching hyperplanes. For the reachability analysis of the continuous dynamics, we use an efficient approximation algorithm based on zonotopes. In order to use this technique for the analysis of hybrid systems, we must also deal with the discrete transitions in a satisfactory (i.e. scalable and accurate) way. For that purpose, we need to approximate the intersection of the continuous reachable sets with the guards enabling the discrete transitions. The main contribution of this paper is a novel algorithm for computing efficiently a tight over-approximation of the intersection of (possibly high-order) zonotopes with a hyperplane. We show the accuracy and the scalability of our approach by considering two examples of reachability analysis of hybrid systems.

1 Introduction

Reachability analysis has been a major research issue in hybrid systems over the past decade [1,2,3,4,5,6,7,8,9]. This research has been motivated by the fact that a successful reachability analysis makes it possible to extend approaches, initially developed in the field of computer science for discrete systems, for analysis and control of hybrid systems [10,11,12,13]. This work resulted in several methods for computing approximations of the reachable sets using, for instance, polytopes [2,3], ellipsoids [4,9] or level sets [5]. The next step was to improve the scalability of these approaches in order to be able to handle larger hybrid systems. Various scalable approaches have been proposed for the reachability analysis of continuous (essentially linear) systems based on classes of polytopes such as hyper-rectangles [6] and zonotopes [7,8], or on ellipsoids [9]. However, in order to use these techniques for the analysis of hybrid systems, we must also deal with the discrete transitions in a satisfactory (i.e. scalable and accurate) way. For that purpose, we need to approximate the intersection of the continuous reachable sets with the guards enabling the discrete transitions.

In this paper, we present a new technique for reachability analysis of hybrid systems with (possibly high dimensional) linear continuous dynamics and guards

M. Egerstedt and B. Mishra (Eds.): HSCC 2008, LNCS 4981, pp. 215–228, 2008.

defined by switching hyperplanes. The reachable set is approximated using zonotopes. The reachability analysis of the continuous dynamics is processed using the algorithm presented in [8]. We handle discrete transitions of the hybrid systems by proposing two new algorithms for computing tight over-approximations of the intersection of a zonotope with a hyperplane. The paper is organized as follows. In section 2, we present briefly the algorithm for reachability analysis of linear systems proposed in [8] and discuss the needs for its extension to hybrid systems reachability. Section 3 is the main contribution of the paper, we first show that the problem of computing a tight over-approximation of the intersection of a zonotope with a hyperplane can be reduced to the problem of computing the intersection of a two dimensional zonotope with a line. Then, we present two efficient algorithms that solve this problem. In section 4, we show the accuracy and the scalability of our approach by considering two examples of reachability analysis of hybrid systems.

2 Reachability of Hybrid Systems

We define informally the class of hybrid systems we consider. The system has several discrete modes; in each mode q, the continuous dynamics of the system is given by a linear differential equation of the form:

$$\dot{x}(t) = A_q x(t) + B_q u(t),\ u(t) \in U_q,$$

where $x(t) \in \mathbb{R}^d$ is the continuous state and $u(t) \in \mathbb{R}^p$ is the continuous input of the system. The system switches from a mode q to mode q' when the continuous state reaches a guard $G_e \subseteq \mathbb{R}^d$ where $e = (q, q')$. We shall assume that the guards are given by switching planes:

$$G_e = \{x \in \mathbb{R}^d : x \cdot n_e = \gamma_e\} \text{ where } n_e \in \mathbb{R}^d \text{ and } \gamma_e \in \mathbb{R}.$$

For simplicity we assume that the reset maps are the identity map, and that there are no Zeno behaviours. In the following, we discuss the over-approximation of the reachable set of the hybrid system by the union of zonotopes.

2.1 Zonotopes

A zonotope is a polytope which can be defined as the Minkowski sum of a finite set of segments. Equivalently it can be seen as the image of a cube by an affine transformation. Formally, a zonotope is a subset of $\mathbb{R}^d$ represented by a center $c \in \mathbb{R}^d$ and a list of generators $g_1, \dots, g_r \in \mathbb{R}^d$:

$$Z = \langle c; g_1, \dots, g_r \rangle = \left\{ c + \sum_{i=1}^{r} \alpha_i g_i : \forall i,\ -1 \le \alpha_i \le 1 \right\}.$$

Each zonotope is a centrally-symmetric convex polytope. Hyper-rectangles and parallelotopes are zonotopes with d generators. The class of zonotopes is closed under arbitrary linear transformations and under the Minkowski sum. The image of a zonotope $Z = \langle c; g_1, \dots, g_r \rangle$ under a linear transformation Φ is the zonotope

$$\Phi Z = \langle \Phi c; \Phi g_1, \dots, \Phi g_r \rangle.$$

The Minkowski sum of two zonotopes $Z = \langle c; g_1, \dots, g_r \rangle$ and $Z' = \langle c'; g'_1, \dots, g'_{r'} \rangle$ is the zonotope

$$Z \oplus Z' = \langle c + c'; g_1, \dots, g_r, g'_1, \dots, g'_{r'} \rangle.$$

Further, it is to be noted that these two operations can be implemented efficiently even in high dimension. This makes the class of zonotopes suitable for reachability analysis.

2.2 Continuous Reachability

We first explain how we handle the continuous dynamics of the hybrid systems. In the following, the results on reachability analysis of linear systems are very briefly described. Details on our approach can be found in [7,8]. Let us consider a linear system of the form:

$$\dot{x}(t) = Ax(t) + Bu(t),\ x(0) \in I,\ u(t) \in U.$$

We want to over-approximate the set of states that are reachable by the linear system within a time interval $[0;T]$ for some initial state in I and admissible input function $u : [0;T] \to U$. We assume that the sets I and U are given by zonotopes. We choose an integration step $\tau = T/(N+1)$ and compute a sequence of zonotopes $\Omega_0, \dots, \Omega_N$ such that Ω_i contains all the states reachable within the time interval $[i\tau, (i+1)\tau]$. We do not detail how the first zonotope of the sequence, Ω_0, is computed (see [7]). Then, the other elements of the sequence can be computed from a recurrence relation of the form:

$$\Omega_{i+1} = \Phi \Omega_i \oplus V,\ i = 0, \dots, N-1 \tag{1}$$

where the matrix $\Phi = e^{\tau A}$ and V is a zonotope that depends on τ, A, B and U (see again [7]). Algorithm 1 is taken from [8] and implements efficiently the computation of the zonotopes $\Omega_1, \dots, \Omega_N$. The time and memory complexities of Algorithm 1 are $\mathcal{O}(Nd^3)$ and $\mathcal{O}(Nd^2)$ respectively.

2.3 Hybrid Reachability

We now discuss the use of Algorithm 1 for reachability analysis of a hybrid system. Again, we keep the discussion informal; our algorithm is similar to the algorithms for reachability analysis of hybrid systems using polytopes [11,2]. Let us assume that the initial discrete mode is q and that the set of initial continuous states is I_q. We start by computing an over-approximation of the reachable set by the continuous dynamics associated with mode q using Algorithm 1; we stop after a zonotope Ω_i has completely crossed a switching plane G_e with $e = (q, q')$

Algorithm 1. Reachability of linear time-invariant systems

Input: The matrix Φ, the sets Ω_0 and U, a positive integer N.
Output: The first N terms of the sequence defined in equation (1).

$X_0 \leftarrow \Omega_0$
$V_0 \leftarrow U$
$S_0 \leftarrow \{0\}$
for i from 0 to $N-1$ **do**
  $X_{i+1} \leftarrow \Phi X_i$ $\triangleright\ X_{i+1} = \Phi^{i+1}\Omega_0$
  $S_{i+1} \leftarrow S_i \oplus V_i$ $\triangleright\ S_{i+1} = \Phi^{i}U \oplus \cdots \oplus U$
  $V_{i+1} \leftarrow \Phi V_i$ $\triangleright\ V_{i+1} = \Phi^{i+1}U$
  $\Omega_{i+1} \leftarrow X_{i+1} \oplus S_{i+1}$ $\triangleright\ \Omega_{i+1} = \Phi^{i+1}\Omega_0 \oplus \Phi^{i}U \oplus \cdots \oplus U$
end for
return $\{\Omega_1, \ldots, \Omega_N\}$

or after a specified number of steps is reached. Then, for all transition e of the form $e = (q, q')$ we need to compute a zonotope $I_{q'}$ which over-approximates the intersection of the reachable set with the hyperplane G_e:

$$(\Omega_0 \cup \cdots \cup \Omega_N) \cap G_e \subseteq I_{q'}.$$

Then, we start over with the discrete mode q' and the set of initial continuous states $I_{q'}$. Hence, we can see that the computation of a good over-approximation of the intersection of a zonotope with a hyperplane is required in order to extend Algorithm 1 for reachability analysis of a hybrid system.

3 Intersection of a Zonotope and a Hyperplane

It is known that detecting the intersection between a zonotope and a hyperplane is an easy problem [7]. Given a zonotope $Z = \langle c; g_1, \ldots, g_r \rangle$ and a hyperplane $G = \{x \in \mathbb{R}^d : x \cdot n = \gamma\}$, we have

$$Z \cap G \neq \emptyset \iff c \cdot n - \sum_{i=1}^{r} |g_i \cdot n| \leq \gamma \leq c \cdot n + \sum_{i=1}^{r} |g_i \cdot n|.$$

Furthermore, in the context of reachability analysis, this can be done efficiently while computing the reachable sets [8]. However, computing this intersection (when it is not empty) is actually a much more complicated problem.

This intersection might not be a zonotope, thus a larger class of sets needs to be considered for this computation. Obviously, we can express the zonotope Z as a polytope, and then compute the intersection between the polytope and the hyperplane G. The good news is that computing a H-representation [14] of a zonotope can be done polynomially in the number of its facets [15], the bad news is that a zonotope with r generators in dimension d might have up to $2\binom{r}{d-1}$ facets [16]. Even for relatively small zonotopes, this can be prohibitively large. Further, the zonotope Ω_k computed by Algorithm 1 typically has about

kd generators. Thus, it is clear that this approach is untractable. Another approach is to over-approximate the zonotope before computing the intersection. However, even if the over-approximation of the zonotope is tight (i.e. the over-approximation touches the zonotope in several points), the over-approximation of the intersection is generally not. We propose a third approach which allows to compute a tight over-approximation of the intersection of a zonotope and a hyperplane. Most of the operations are done in two dimensional spaces, thus leading to efficient computations.

3.1 From Dimension d to Dimension 2

Finding a tight polyhedral over-approximation P of a set X can be done by bounding this set using several hyperplanes with normal vectors in a given finite set $\mathcal{D} = \{\ell_1, \ldots, \ell_p\}$. The computation involves determining, for each $\ell \in \mathcal{D}$, the infimum m_ℓ and supremum M_ℓ of the sets $\{x \cdot \ell : x \in X\}$. Then, the over-approximation P is given by

$$P = \{x \in \mathbb{R}^d : \forall \ell \in \mathcal{D},\, m_\ell \leq x \cdot \ell \leq M_\ell\}.$$

In our case[1], X is the intersection of the zonotope Z and the hyperplane G. For the reasons we already explained, we can not expect to solve this problem in the full dimensional state-space $\mathbb{R}^d$. The following proposition will allow us to reduce this problem to a two-dimensional problem.

Proposition 1. *Let G be a hyperplane, $G = \{x \in \mathbb{R}^d : x \cdot n = \gamma\}$, Z a set, and ℓ a vector. Let $\Pi_{n,\ell}$ be the following linear transformation:*

$$\begin{aligned} \Pi_{n,\ell} : \mathbb{R}^d &\to \mathbb{R}^2 \\ x &\mapsto (x \cdot n, x \cdot \ell) \end{aligned}$$

Then, we have the following equality

$$\{x \cdot \ell : x \in Z \cap G\} = \{y : (\gamma, y) \in \Pi_{n,\ell}(Z)\}$$

Proof. Let y belongs to $\{x \cdot \ell : x \in Z \cap G\}$, then there exists x in $Z \cap G$ such that $x \cdot \ell = y$. Since $x \in G$, we have $x \cdot n = \gamma$. Therefore $(\gamma, y) = \Pi_{n,\ell}(x) \in \Pi_{n,\ell}(Z)$ because $x \in Z$. Thus, $y \in \{y : (\gamma, y) \in \Pi_{n,\ell}(Z)\}$. Conversely, if $y \in \{y : (\gamma, y) \in \Pi_{n,\ell}(Z)\}$, then $(\gamma, y) \in \Pi_{n,\ell}(Z)$. It follows that there exists $x \in Z$ such that $x \cdot n = \gamma$ and $x \cdot \ell = y$. Since $x \cdot n = \gamma$, it follows that $x \in G$. Thus, $y = x \cdot \ell$ with $x \in Z \cap G$ and it follows that $y \in \{x \cdot \ell : x \in Z \cap G\}$. ■

This proposition states that we can reduce the problem of computing a tight polyhedral over-approximation of the intersection of a set Z and a hyperplane G to the problem of projecting Z on a plane and then computing the intersection of the 2-dimensional set $\Pi_{n,\ell}(Z)$ and the line $L_\gamma = \{(x, y) \in \mathbb{R}^2 : x = \gamma\}$. This must be done for each vector $\ell \in \mathcal{D}$. Algorithm 2 implements this idea.

[1] The results presented in section 3.1 hold for an arbitrary set Z (not only a zonotope).

Algorithm 2. Dimension reduction

Input: A set Z, a hyperplane $G = \{x \in \mathbb{R}^d : x \cdot n = \gamma\}$ and a finite set $\mathcal{D}$ of directions.
Output: A polytope approximating tightly $Z \cap G$ in directions given by $\mathcal{D}$.
1: **for** ℓ in $\mathcal{D}$ **do**
2: $\quad S^{\pi}_{n,\ell} \leftarrow \Pi_{n,\ell}(Z)$
3: $\quad [m_\ell; M_\ell] \leftarrow$ BOUND_INTERSECT_2D($S^{\pi}_{n,\ell}$, L_γ)
4: **end for**
5: **return** $\{x \in \mathbb{R}^d : \forall \ell \in \mathcal{D},\, m_\ell \leq x \cdot \ell \leq M_\ell\}$

In our case, the set Z is a zonotope, then the projection $\Pi_{n,\ell}(Z)$ is a two-dimensional zonotope which can be computed efficiently:

$$\Pi_{n,\ell}(\langle c; g_1, \ldots, g_r \rangle) = \langle \Pi_{n,\ell}(c); \Pi_{n,\ell}(g_1), \ldots, \Pi_{n,\ell}(g_r) \rangle.$$

Remark 1. For each generator g of the zonotope Z, one has to compute $\Pi_{n,\ell}(g)$ for all ℓ in $\mathcal{D}$, but instead of computing these projections independently, which would lead to $2|\mathcal{D}|$ scalar products, one can observe that all the $\Pi_{n,\ell}(g)$ involves computing the scalar product $n \cdot g$, thus only $|\mathcal{D}|+1$ scalar products are necessary for each generator of Z. The projections can thus be done by computing the product of a $(|\mathcal{D}| + 1) \times d$ matrix by a $d \times (r + 1)$ matrix.

The computation of the intersection of $\Pi_{n,\ell}(Z)$ and the line L_γ is investigated in the next subsection where two algorithms are proposed to solve this problem.

3.2 Intersection of a Zonogon and a Line

Algorithm 2 requires the computation of the intersection of a two dimensional zonotope, with a line. In a two dimensional space, a zonotope is called a zonogon and its number of vertices, as its number of edges, is two times its number of generators. Thus, it is possible to express a zonogon as a polygon (two dimensional polytope) which can easily be intersected with a line. For the simplicity of the notations, we now denote by $Z = \langle c; g_1, \ldots, g_r \rangle$ the zonogon that we want to intersect with $L_\gamma = \{(x, y) : x = \gamma\}$. An extremely naive way of determining the list of vertices of a zonogon is to generate the list of points $\{c + \sum_{i=1}^{r} \alpha_i g_i : \forall i, \alpha_i = -1 \text{ or } \alpha_i = 1\}$ and then to take the convex hull of this set. This is clearly not a good approach since we need to compute a list of 2^r points.

Scanning the vertices. It is known that the facets of a zonotope $\langle c; g_1, \ldots, g_r \rangle$ are zonotopes whose generators are taken from the list $\{g_1, \ldots, g_r\}$. Then, we can deduce that the edges of a zonogon are segments of the form $[P; P + 2g]$ where P is a vertex of the zonogon and g a generator. Therefore, it is sufficient to scan the generators in trigonometric (or anti-trigonometric) order to scan the vertices of the zonogon in a way that is similar to the gift wrapping algorithm [17]. This idea is implemented in Algorithm 3.

Algorithm 3. BOUND_INTERSECT_2D

Input: A zonogon $Z = \langle c; g_1, \dots, g_r \rangle$ and a line $L_\gamma = \{(x, y) : x = \gamma\}$
Output: A segment $[m; M]$ such that $\{\gamma\} \times [m; M] = Z \cap L_\gamma$.

$P \leftarrow c$ ▷ current position
$m \leftarrow \infty$, $M \leftarrow -\infty$
for i from 1 to r **do**
  if $y_{g_i} < 0$ or ($y_{g_i} = 0$ and $x_{g_i} < 0$) **then** ▷ $g_i = (x_{g_i}, y_{g_i})$
    $g_i \leftarrow -g_i$ ▷ Ensure all generators are pointing upward
  end if
  $P \leftarrow P - g_i$ ▷ Drives P toward the lowest vertex of Z
end for
$g_{i_1}, \dots, g_{i_r} \leftarrow \mathrm{SORT}(g_1, \dots, g_r)$ ▷ Sort the generators in trigonometric order
for j from 1 to r **do**
  if $[P; P + 2g_{i_j}]$ intersects L_γ **then**
    $(x, y) \leftarrow [P; P + 2g_{i_j}] \cap L_\gamma$
    $m \leftarrow \min(m, y)$
    $M \leftarrow \max(M, y)$
  end if
  $P \leftarrow P + 2g_{i_j}$
end for ▷ Only half of the vertices of the zonogon have been scanned
for j from 1 to r **do**
  if $[P; P - 2g_{i_j}]$ intersects L_γ **then**
    $(x, y) \leftarrow [P; P - 2g_{i_j}] \cap L_\gamma$
    $m \leftarrow \min(m, y)$
    $M \leftarrow \max(M, y)$
  end if
  $P \leftarrow P - 2g_{i_j}$
end for ▷ we are back in $P = e$
return $[m; M]$

All the generators are taken poiting upward for simplicity, this does not change the zonogon since replacing a generator g by it opposite $-g$ does not modify the shape of a zonogon. Then, we compute the lowest vertex of Z, and sort the generators according to the trigonometric order. Scanning the generators in that order allows us to scan the vertices of Z. While scanning these vertices, we check for the intersection with the line L_γ. This leads to an algorithm for the intersection between a line and a zonogon with r generators whose complexity is $\mathcal{O}(r \log r)$. The most time consuming part is to sort the generators.

In practice, the number of generators r can be very large (remember that the zonogon we want to intersect comes from the reachable set Ω_k computed by algorithm 1; Ω_k has about kd generators). Further, each time a discrete transition occurs, this procedure is called several times by algorithm 2 (one call for each direction of approximation). Thus, we need it to be as fast as possible. Hence, instead of scanning all the vertices of Z, we look directly for the two edges that intersect the line L_γ with a dichotomic search.

Dichotomic search of the intersecting edges. We start again from the lowest vertex of Z. At each step of the algorithm, P is a vertex of the zonogon representing the current position and G is a set of generators. We know that the segment $[P; P+\sum_{g\in G} 2g]$ intersects the line L_γ. We choose a pivot vector s and split the generators in G into two sets $G_<$ and $G_>$, the set of generators respectively smaller and bigger (according to the trigonometric order) than s. Then, it is clear that L_γ intersects either $[P; P + \sum_{g\in G_<} 2g]$ or $[P + \sum_{g\in G_<} 2g; P + \sum_{g\in G} 2g]$. We continue either with P and $G_<$ or $P + \sum_{g\in G_<} 2g$ and $G_>$. When the lowest vertex of the intersection is found, we start again from the highest vertex of Z in order to find the highest vertex of the intersection. Algorithm 4 implements this approach. Figure 1 illustrates the execution of the algorithm, both from the lowest and the highest point at the same time.

Algorithm 4. BOUND_INTERSECT_2D

Input: A zonogon $Z = \langle c; g_1, \dots, g_r\rangle$ and a line $L_\gamma = \{(x, y) : x = \gamma\}$
Output: A segment $[m; M]$ such that $\{\gamma\} \times [m; M] = Z \cap L_\gamma$.

$P \leftarrow c$ ▷ current position $P = (x_P, y_P)$
$m \leftarrow \infty$, $M \leftarrow -\infty$
for i from 1 to r **do**
  if $y_{g_i} < 0$ or ($y_{g_i} = 0$ and $x_{g_i} < 0$) **then** ▷ $g_i = (x_{g_i}, y_{g_i})$
    $g_i \leftarrow -g_i$ ▷ Ensure all generators are pointing upward
  end if
  $P \leftarrow P - g_i$ ▷ Drives P toward the lowest vertex of Z
end for
if $x_p < \gamma$ **then**
  $G \leftarrow \{g_1, \dots, g_r\} \cap (\mathbb{R}^+ \times \mathbb{R})$ ▷ We should look right
else
  $G \leftarrow \{g_1, \dots, g_r\} \cap (\mathbb{R}^- \times \mathbb{R})$ ▷ or left
end if
$s \leftarrow \sum_{g\in G} 2g$
while $|G| > 1$ **do**
  $(G_1, G_2) \leftarrow$ SPLIT_PIVOT(G, s)
  $s_1 \leftarrow \sum_{g\in G_1} 2g$
  if $[P; P + s_1]$ intersects L_γ **then**
    $G \leftarrow G_1$
    $s \leftarrow s_1$
  else
    $G \leftarrow G_2$
    $s \leftarrow s - s_1$
    $P \leftarrow P + s_1$
  end if
end while ▷ Only one generator remains
$(x, y) \leftarrow [P; P + s] \cap L_\gamma$
$m \leftarrow y$
... ▷ Same thing for M, starting from the upper vertex of Z
return $[m; M]$

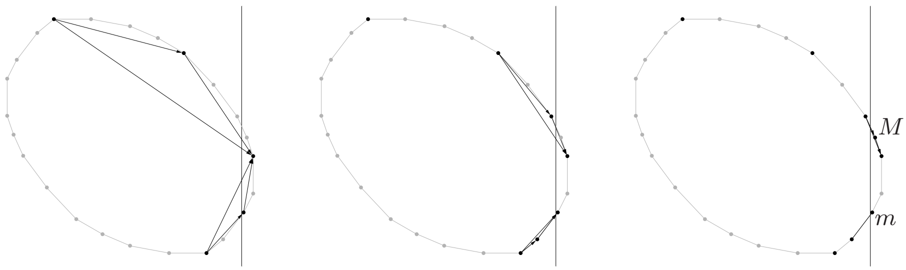

Fig. 1. Dichotomic search of the intersecting edges

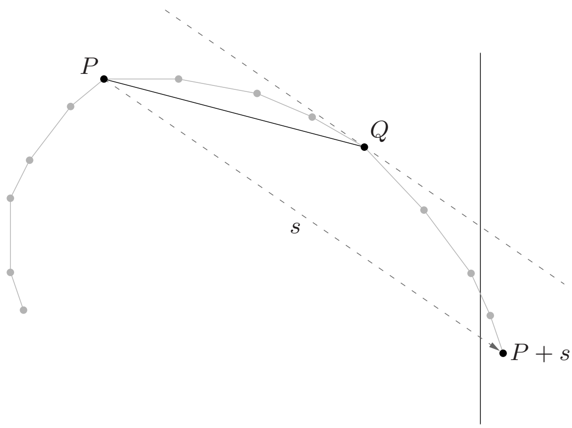

Fig. 2. A good choice for the pivot allows a smart enclosure of the intersection point

With a good pivot selection algorithm [18], the dichotomic search has a linear complexity. For our problem, we choose the sum of the remaining generators as the pivot. Even though this leads to a quadratic theoretical worst case complexity, it improves the practical behavior. Indeed, the sum of the remaining generators is already available and it has a nice geometric interpretation, as illustrated in Figure 2. At each step, P and $P + \sum_{g \in G} 2g$ are both the closest computed vertex to the line $\{(x, y) : x = \gamma\}$, each on a different side of this line, thus defining the best computed under-approximation of the interval $[m; M]$ at this step. A pivot s defines a vertex $Q = P + \sum_{g \in G_<} 2g$ between P and $P + \sum_{g \in G} 2g$. The line of direction s going through Q is "tangent" to Z and its intersection with L_γ defines an over-approximation of the interval $[m; M]$. Choosing $s = \sum_{g \in G} g$ as the pivot ensures that the distance between the over-approximation and the under-approximation of the interval $[m; M]$ is not correlated with γ, the position of the intersecting line.

Remark 2. Algorithm 4 is similar to Bamas and Zemel's algorithm for the fractional Knapsack problem [19]. Eppstein[2] suggested in a talk [20] that one could

[2] The authors wish to thank an anonymous reviewer for pointing out this reference.

maximize a linear function on the intersection of a zonotope and a hyperplane by adapting the greedy algorithm for the fractional Knapsack problem. This is actually what algorithm 3 does.

3.3 Intersection of the Reachable Set and a Guard

Now that we know how to intersect a zonogon with a line, we can approximate the intersection of a zonotope with a hyperplane, using Algorithm 2. In the context of reachability analysis, the intersection between the reachable sets $\Omega_0, \dots, \Omega_N$ with a guard G generally occurs at several steps. Let $\mathcal{I}_G$ be the set of indices i for which Ω_i intersects the guard G. One can approximate the intersection between each Ω_i and G independently, and then compute the union of these intersections in order to get an approximation of the intersection of the reachable set with the guard G. Using this approach, we do not exploit the fact that the reachable sets Ω_i have a special structure. They actually share a lot of generators. Let us assume that $\mathcal{I}_G$ is a set of $k+1$ consecutive integers i, $i+1$, …, $i+k$. With the notations of Algorithm 1, the zonotopes intersecting the guards are:

$$\begin{aligned}
\Omega_i &= X_i \oplus S_i, \\
\Omega_{i+1} &= X_{i+1} \oplus V_{i+1} \oplus S_i, \\
&\vdots \\
\Omega_{i+k} &= X_{i+k} \oplus V_{i+k} \oplus \ldots \oplus V_{i+1} \oplus S_i
\end{aligned}$$

They all share the generators in S_i. Actually each zonotope Ω_j shares all its generators but the ones in X_j with the zonotopes of greater index. Consequently, when approximating the intersection at step j in $\mathcal{I}_G$, it is possible to reuse most of the computations already done for smaller indices. Not only the projections of most of the generators of Ω_j have already been computed, but they are also partially sorted. Moreover, at each step of Algorithm 4, one can easily compute an under-approximation and an over-approximation of the intersection, as explained at the end of the previous subsection and on Figure 2. It is then possible to modify Algorithm 4 in order to compute all the intersection concurrently. Since we are interested in $(\cup_{i\in\mathcal{I}_G}\Omega_i)\cap G$ and not in each $\Omega_i \cap G$, we can, at each step, drop the computation of the $\Omega_i \cap G$ whose over-approximation is included in the under-approximation of $(\cup_{i\in\mathcal{I}_G}\Omega_i)\cap G$.

3.4 From Polytopes to Zonotopes

Let us remark that the tight over-approximation of the intersection between the reachable set and the guard G which is computed using Algorithm 2 is a polytope. In order to process the continuous reachability analysis using Algorithm 1 in the next discrete mode, we need this over-approximation to be expressed as a zonotope. To the best of the authors knowledge, there is no known efficient algorithm for the approximation of a general polytope by a zonotope (except in small dimension [21]). In Algorithm 2, we have the choice of the normal vectors to the facets of the approximating polytope. Then, we can choose these vectors such that the resulting polytope can be easily approximated by a zonotope. Even better, we can choose these vectors such that the approximating polytope is a zonotope. Indeed, some polytopes are easily expressed as zonotopes; this is the case for the class of parallelotopes and particularly for hyper-rectangles. Hence, we choose the normal vectors to the facets such that the over-approximation of the intersection of the reachable set with the the guard is a hyper-rectangle.

Initially, we do not have much information on the intersection, so we generate at random (in a way similar to [22]) a set of directions, and only keep the direction ℓ_0 that induces the thinner approximation. Then, we randomly generate a set of directions orthogonal to the directions already chosen, and again we only keep the one for which $M_\ell - m_\ell$ is minimal, until we get a hyper-rectangle, after $d-2$ steps.

4 Examples

The algorithms presented in this paper have been implemented in Ocaml [23]. In this section, we show the effectiveness of our approach on some examples. All computations were performed on a Pentium IV 3.2GHz with 1GB RAM.

4.1 5-Dimensional Benchmark

To evaluate the error introduced by our method, and its usability in a hybrid reachability toolchain, we would like to compare the computed reachable set with the exact one. As explained before, the computation of the exact intersection is untractable. This is why we artificially add a switching hyperplane to a continuous linear system. This guard will allow a transition between two states with the same dynamic (see figure 3).

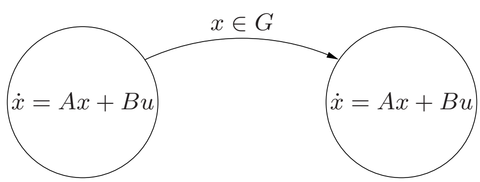

Fig. 3. A *(not so)* hybrid system on which approximate reachable sets can be compared to the exact reachable sets (computed with a non-hybrid view of the system)

The exact (up to initial time discretization errors) reachability analysis of this hybrid system can be done with algorithm 1, by removing the guard. This analysis can then be compared to the one done using our algorithm for approximating the intersection with the guard on the hybrid view of the system. We applied our methods on such a hybrid system contructed on a five dimensional linear system [7,8]. The projection on the first two variables of the computed reachables sets can be seen on figure 4.

The exact reachable set, computed by algorithm 1 on the non-hybrid system, has been plotted in black. After the intersection the error introduced by the

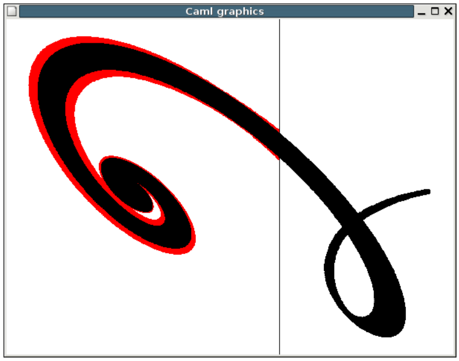

Fig. 4. Error introduced by approximating the intersection (in red)

approximation appears in red. The directions of approximation were chosen as explained in section 3.4, 16 at each step. We only kept 4 out of the 49 computed constraints on the intersection, in order to be able to express it as a zonotope. This first approximation was improved by adding 4 generators, introducing 8 new facets. The whole computation, including intersection and reachability, took 0.2 seconds and 1.4 MB. If we try to compute exactly the intersection, by expressing the intersecting zonotopes as polytopes, we have to deal with more than 10^{11} vertices. Only storing these vertices would require more than 1.8 terabyte, more than one million times what we need for approximate intersection and reachability.

4.2 Thermostat

As a second example, we consider a high dimensional hybrid system with two discrete states. A heat source can be switched on or off at one end of a metal rod of length 1, the switching is driven by a sensor placed at the middle of the rod. The temperature at each point x of the rod, $T(x,t)$ is driven by the Heat equation:

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}.$$

When the heat source is *ON*, we have $T(0,t) = 1$, and when it is *OFF*, $T(0,t) = 0$. We approximate this partial differential equation by a linear ordinary differential equation using a finite difference scheme on a grid of size $\frac{1}{90}$.

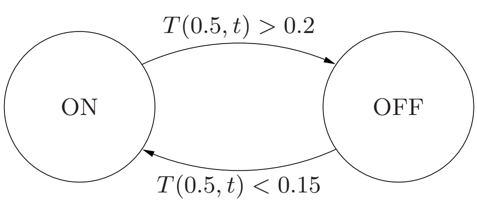

Fig. 5. Hybrid model of a thermostat

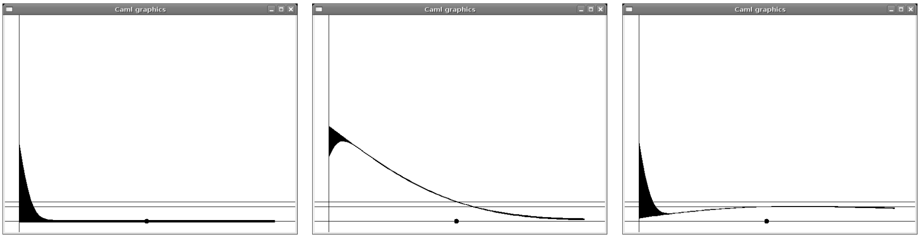

Fig. 6. Reachable set at three different times. The x-axis represents the position in the metal rod, and the y-axis the temperature. The dot on the middle of the x-axis specify the position of the sensor, and the two horizontal line the switching temperatures. The heat source is on the left.

The resulting hybrid system has 89 continuous variables and 2 discrete states (see figure 5). We computed the reachable sets of this system for 1000 times step, during which 10 discrete transitions occured, in 71.6s using 406MB of memory. Figure 6 shows the reachable sets at three different time, each after a discrete transition.

5 Conclusion

In this paper, we presented an efficient algorithm for computing a tight over-approximation of the intersection between a zonotope and a hyperplane. We showed that it can be used in conjunction with a reachability analysis algorithm for continuous linear systems to effectively analyze hybrid systems with high dimensional linear continuous dynamic.

The use of the zonotope representation can be seen as a trick allowing us not to compute the full dimensional Minkowski sum, this trick can in fact be applied to more complex objects and it is possible to adapt our algorithm so that it can handle intersection between a hyperplane and the Minkowski sum of a set of ellipsoids and zonotopes. An other extension should allow us to compute an under-approximation of the intersection. This under-approximation might be useful for the choice of the directions of approximation.

Future work also includes the approximation of a polytope by a zonotope, to avoid loosing most of the computed constraints (in the 5-dimensional example, we only kept 8 out of the 53 computed constraints).

References

1. Varaiya, P.: Reach set computation using optimal control. In: Proc. KIT Workshop on Verification of Hybrid Systems, Verimag, Grenoble (1998)
2. Chutinan, A., Krogh, B.H.: Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In: Vaandrager, F.W., van Schuppen, J.H. (eds.) HSCC 1999. LNCS, vol. 1569, pp. 76–90. Springer, Heidelberg (1999)

3. Asarin, E., Dang, T., Maler, O., Bournez, O.: Approximate reachability analysis of piecewise-linear dynamical systems. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 20–31. Springer, Heidelberg (2000)
4. Kurzhanski, A.B., Varaiya, P.: Ellipsoidal techniques for reachability analysis. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 202–214. Springer, Heidelberg (2000)
5. Mitchell, I., Tomlin, C.: Level set methods for computation in hybrid systems. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 310–323. Springer, Heidelberg (2000)
6. Stursberg, O., Krogh, B.H.: Efficient representation and computation of reachable sets for hybrid systems. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 482–497. Springer, Heidelberg (2003)
7. Girard, A.: Reachability of uncertain linear systems using zonotopes. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 291–305. Springer, Heidelberg (2005)
8. Girard, A., Le Guernic, C., Maler, O.: Efficient computation of reachable sets of linear time-invariant systems with inputs. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 257–271. Springer, Heidelberg (2006)
9. Kurzhanskiy, A.A., Varaiya, P.: Ellipsoidal techniques for reachability analysis of discrete-time linear systems. IEEE Trans. Automatic Control 52, 26–38 (2007)
10. Asarin, E., Bournez, O., Dang, T., Maler, O., Pnueli, A.: Effective synthesis of switching controllers for linear systems. Proc. of the IEEE 88(7), 1011–1025 (2000)
11. Dang, T.: Vérification et Synthèse des Systèmes Hybrides. PhD thesis, Institut National Polytechnique de Grenoble (2000)
12. Maler, O.: Control from computer science. IFAC Annual Reviews in Control (2003)
13. Tomlin, C., Mitchell, I., Bayen, A., Oishi, M.: Computational techniques for the verification and control of hybrid systems. Proc. of the IEEE 91(7), 986–1001 (2003)
14. Ziegler, G.M.: Lectures on Polytopes. In: Graduate Texts in Mathematics, vol. 152, Springer, Heidelberg (1995)
15. Avis, D., Fukuda, K.: Reverse search for enumeration. Discrete Appl. Math. 65(1-3), 21–46 (1996)
16. Zaslavsky, T.: Facing Up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes. In: Memoirs of the American Mathematical Society. American Mathematical Society, vol. 154 (1975)
17. Jarvis, R.A.: On the identification of the convex hull of a finite set of points in the plane. Inf. Process. Lett. 2(1), 18–21 (1973)
18. Blum, M., Floyd, R.W., Pratt, V.R., Rivest, R.L., Tarjan, R.E.: Time bounds for selection. J. Comput. Syst. Sci. 7(4), 448–461 (1973)
19. Balas, E., Zemel, E.: An algorithm for large zero-one knapsack problems. Operations Research 28(5), 1130–1154 (1980)
20. Bern, M.W., Eppstein, D.: Optimization over zonotopes and training support vector machines (2001) Talk given at WADS
21. Guibas, L.J., Nguyen, A., Zhang, L.: Zonotopes as bounding volumes. In: SODA 2003: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 803–812 (2003)
22. Muller, M.E.: A note on a method for generating points uniformly on n-dimensional spheres. Commun. ACM 2(4), 19–20 (1959)
23. Leroy, X.: The Objective Caml system. In: INRIA (2007)