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

Efficient Reachability Analysis for Linear Systems using Support Functions

Antoine Girard * **Colas Le Guernic** **

* *Laboratoire Jean Kuntzmann, Université Joseph Fourier, 51 rue des Mathématiques, B.P. 53, 38041 Grenoble Cedex 9, France (e-mail: Antoine.Girard@imag.fr).*
** *VERIMAG, Université Joseph Fourier, Centre Equation - 2, avenue de Vignate, 38610 Gières, France (e-mail: Colas.Le-Guernic@imag.fr).*

Abstract: This work is concerned with the algorithmic reachability analysis of linear systems with constrained initial states and inputs. In this paper, we present a new approach for the computation of tight polyhedral over-approximations of the reachable sets of a linear system. The main contribution over our previous work is that it makes it possible to consider systems whose sets of initial states and inputs are given by arbitrary compact convex sets represented by their support functions. We first consider the discrete-time setting and then we show how our algorithm can be extended to handle continuous-time linear systems. Finally, the effectiveness of our approach is demonstrated through several examples.

Keywords: Reachability analysis, support functions, computational methods, linear systems.

1. INTRODUCTION

Computers have become ubiquitious in control systems design, offering the opportunity for the development of new techniques for synthesis and analysis. One of these approaches, inspired by the algorithmic verification of discrete systems (i.e. model checking (Clarke et al. (2000))), has emerged from hybrid systems research and is based on reachability analysis. It consists in computing the reachable sets of a system; thus making it possible to examine all its possible behaviours. This information can then be used either for algorithmic verification or controller synthesis (see e.g. Dang (2000); Tomlin et al. (2003)).

Numerous techniques have been developped in the latest decade for reachability analysis of hybrid systems (see e.g. Bemporad and Morari (1999); Chutinan and Krogh (1999); Asarin et al. (2000); Kurzhanski and Varaiya (2000); Mitchell and Tomlin (2000)). The standard hybrid reachability algorithm alternates computations of the sets reachable under the discrete dynamics and of the sets reachable under the continuous dynamics. Reachability under the continuous dynamics is often the most challenging part of the job and it has been the main focus of the work on hybrid system reachability.

In this paper, we consider the computation of the reachable sets of discrete-time and continuous-time linear systems with constrained initial states and inputs. These classes of systems have been considered in several previous papers including Varaiya (1998); Bemporad and Morari (1999); Kurzhanski and Varaiya (2000); Girard (2005); Girard et al. (2006); Kurzhanskiy and Varaiya (2007).

⋆ This work was partially supported by the ANR project VAL-AMS.

Given a discrete-time linear system of the form:

$$x_{k+1} = Ax_k + Bu_k,\ x_0 \in I,\ u_k \in U \tag{1}$$

where I and U are compact convex sets. We denote by Ω_k the set of reachable states at time k. It is straighforward to verify that it satisfies the recurrence relation:

$$\Omega_{k+1} = A\Omega_k \oplus BU,\ \Omega_0 = I \tag{2}$$

where $\oplus$ denotes the Minkowski sum [1]. For a given time-horizon N, we are interested in computing the sequence of sets $\Omega_0, \dots, \Omega_N$. The relation (2) can be implemented exactly for the classes of sets that are closed under linear transformation and Minkowski sum (e.g. polytopes or zonotopes). However, at each step of the computation, the size of the representation of the set Ω_k grows and the problems becomes rapidly intractable for long time horizons.

The usual turnaround is to make an over-approximation at each step in order to limit the size of the representation (Stursberg and Krogh (2003); Girard (2005)), leading to a relation of the following type:

$$\tilde{\Omega}_{k+1} = \text{Approx}\left(A\tilde{\Omega}_k \oplus BU\right),\ \tilde{\Omega}_0 = I.$$

However, this scheme is subject to the so called *wrapping effect*: at each step, starting points for new trajectories are added to $\tilde{\Omega}_k$, which can lead eventually to dramatic over-approximations (see Kühn (1999)). In (Girard et al. (2006)), we proposed an algorithm avoiding this wrapping effect problem for linear systems whose sets of initial states and inputs are zonotopes.

[1] The Minkowski sum of two sets $U, V \subseteq \mathbb{R}^d$ is the set defined by $U \oplus V = \{u + v : u \in U, v \in V\}$.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.0569

In this paper, we extend our approach in order to handle arbitrary compact convex sets. We represent convex sets by their support functions, a standard tool in convex analysis (see e.g. Bertsekas et al. (2003); Boyd and Vandenberghe (2004)). The use of support functions for reachability analysis has been considered previously in Varaiya (1998) and we include in this paper a comparison between the two approaches.

The paper is organized as follows. In section 2, we introduce the notion of support function. In section 3, we present our reachability algorithm for discrete-time linear systems. We propose additional improvements to increase its efficiency, we show how it can be used for control synthesis and discuss its relation to the work presented in Varaiya (1998). In section 4, we extend our algorithm to handle continuous-time linear systems. The effectiveness of our approach is shown in section 5 where several examples are considered.

2. SUPPORT FUNCTIONS

In this section, we present the notion of support function that we will use to represent convex sets. The properties of support functions are stated here without the proofs that are quite straightforward and can be found in several textbooks on convex analysis (see e.g. Bertsekas et al. (2003); Boyd and Vandenberghe (2004)).

Definition 1. The support function of a compact convex set $\Omega \subseteq \mathbb{R}^d$, denoted ρ_Ω, is defined as:

$$\begin{aligned} \rho_\Omega : \mathbb{R}^d &\to \mathbb{R} \\ \ell &\mapsto \max_{x \in \Omega} \ell \cdot x \end{aligned}$$

We also introduce the related notion of support vectors:

Definition 2. A support vector of a compact convex set $\Omega \subseteq \mathbb{R}^d$, in the direction $\ell \in \mathbb{R}^d$, denoted $\nu_{\Omega,\ell}$, is a vector of $\mathbb{R}^d$ such that

$$\nu_{\Omega,\ell} \in \Omega \text{ and } \ell \cdot \nu_{\Omega,\ell} = \rho_\Omega(\ell).$$

Let us remark that, generally, the support vector of Ω in the direction ℓ is not unique. It is to be noted that a compact convex set is uniquely determined by its support function as the following equality holds:

$$\Omega = \bigcap_{\ell \in \mathbb{R}^d} \{x \in \mathbb{R}^d : \ell \cdot x \leq \rho_\Omega(\ell)\}. \tag{3}$$

From equation (3), it is easy to see that a tight polyhedral over-approximation of an arbitrary compact convex set can be obtained by "sampling" its support function.

Proposition 1. Let Ω be a compact convex set and $\ell_1, \ldots, \ell_r \in \mathbb{R}^d$ be arbitrary chosen vectors, we define the following halfspaces:

$$\mathcal{H}_i = \{x \in \mathbb{R}^d : \ell_i \cdot x \leq \rho_\Omega(\ell_i)\},\ i = 1, \ldots, r.$$

Let us define the polyhedron $\tilde{\Omega} = \bigcap_{i=1}^r \mathcal{H}_i$. Then, $\Omega \subseteq \tilde{\Omega}$. Moreover, the over-approximation is tight as Ω touches the faces of $\tilde{\Omega}$ at the points $\nu_{\Omega,\ell_1}, \ldots, \nu_{\Omega,\ell_r}$.

For most classes of sets that are commonly used in the context of reachability analysis, the support function and a support vector can be easily computed:

- If Ω is the unit ball then
$$\rho_\Omega(\ell) = \|\ell\|_2 \text{ and } \nu_{\Omega,\ell} = \frac{\ell}{\|\ell\|_2}$$
where $\|.\|_2$ denotes the usual Euclidean norm in $\mathbb{R}^d$.
- If Ω is an ellipsoid: there exists a positive definite symmetric matrix Q such that
$$\Omega = \left\{x : x^T Q^{-1} x \leq 1\right\},$$
then
$$\rho_\Omega(\ell) = \sqrt{\ell^T Q \ell} \text{ and } \nu_{\Omega,\ell} = \frac{Q\ell}{\sqrt{\ell^T Q \ell}}.$$
- If Ω is the hyper-rectangle $[-h_1; h_1] \times \ldots \times [-h_d; h_d]$, then
$$\rho_\Omega(\ell) = \sum_{i=1}^d |h_i \ell_i| \text{ and } \nu_{\Omega,\ell} = (\sigma(l_1)h_1, \ldots, \sigma(l_d)h_d)^T$$
where σ denotes the sign function:
$$\forall x \in \mathbb{R},\ \sigma(x) = \begin{cases} 1 & \text{iff } x > 0 \\ 0 & \text{iff } x = 0 \\ -1 & \text{iff } x < 0 \end{cases}$$
- If Ω is a zonotope: there exists $g_1, \ldots, g_r \in \mathbb{R}^d$ such that
$$\Omega = \left\{\sum_{i=1}^r \alpha_i g_i : -1 \leq \alpha_i \leq 1 \ \ i = 1, \ldots, r\right\},$$
then
$$\rho_\Omega(\ell) = \sum_{i=1}^r |g_i \cdot \ell| \text{ and } \nu_{\Omega,\ell} = \sum_{i=1}^r \sigma(g_i \cdot \ell) g_i$$
- More generally, if Ω is a polytope then computing $\rho_\Omega(\ell)$ and $\nu_{\Omega,\ell}$ is equivalent to solving a linear program.

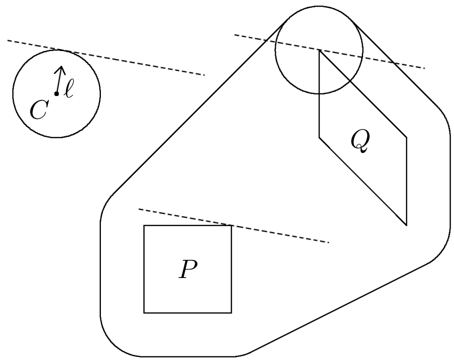

Fig. 1. Computation of the support function of the Minkowski sum of the convex hull of two parallelograms with a circle. $\rho_{\mathrm{CH}(P,Q) \oplus C}(\ell) = \max(\rho_P(\ell), \rho_Q(\ell)) + \rho_C(\ell)$.

Thus, support functions and support vectors can be computed efficiently for a large class of sets. Further, more complex sets can be represented easily by combining elementary support functions using the following properties:

Proposition 2. For all matrices A, all compact convex sets $U, V \subseteq \mathbb{R}^d$, and all non-zero vectors $\ell \in \mathbb{R}^d$, we have:

$$\begin{aligned} \rho_{\mathrm{CH}(U,V)}(\ell) &= \max(\rho_U(\ell), \rho_V(\ell)) \\ \rho_{U \oplus V}(\ell) &= \rho_U(\ell) + \rho_V(\ell) \\ \rho_{AU}(\ell) &= \rho_U(A^T \ell) \end{aligned}$$

where $\mathrm{CH}(U, V)$ denotes the convex hull of U and V.

Using these properties, one can easily consider convex sets of unusual shape without really worrying about their internal representation. Figure 1 illustrates how the support function of the Minkowski sum of the convex hull of two parallelograms with a circle can be computed.

3. REACHABILITY OF DISCRETE-TIME SYSTEMS

We now move to the main contribution of the paper. Let us consider a discrete-time linear system of the form:

$$x_{k+1} = Ax_k + v_k,\ x_0 \in I,\ v_k \in V \tag{4}$$

where $I \subseteq \mathbb{R}^d$ and $V \subseteq \mathbb{R}^d$ are compact convex sets. Equation (4) is equivalent to (1) by setting $V = BU$. We denote by Ω_k the subset of states reachable at time k. For a given time-horizon N, we are interested in computing the sequence of sets $\Omega_0, \dots, \Omega_N$. Equivalent to equation (2), we have the following recurrence relation:

$$\Omega_{k+1} = A\Omega_k \oplus V,\ \Omega_0 = I. \tag{5}$$

The exact computation of these sets, though sometimes possible, is often intractable as the complexity of the representation of Ω_k increases at each iteration. For that reason, we are interested in computing simple over-approximations of the reachable sets.

Using Proposition 1, we will compute a tight polyhedral over-approximation $\tilde{\Omega}_k$ of the reachable set Ω_k. Given a certain number of arbitrarily chosen directions $\ell_1, \dots, \ell_r$, the polyhedron $\tilde{\Omega}_k$ will be defined as the intersection of halfspaces:

$$\mathcal{H}_{k,i} = \{x : \ell_i \cdot x \le \rho_{\Omega_k}(\ell_i)\},\ i = 1, \dots, r.$$

Then, computing the over-approximation $\tilde{\Omega}_k$ of Ω_k is equivalent to evaluating the support function ρ_{Ω_k} at $\ell_1, \dots, \ell_r$.

3.1 Computing the support function

Let $\ell \in \mathbb{R}^d$, in this part we propose an efficient algorithm for the computation of $\rho_{\Omega_0}(\ell), \dots, \rho_{\Omega_N}(\ell)$.

Proposition 3. For all $k \in \{0, \dots, N\}$,

$$\rho_{\Omega_k}(\ell) = \rho_I\left((A^T)^k \ell\right) + \sum_{i=0}^{k-1} \rho_V\left((A^T)^i \ell\right). \tag{6}$$

Proof The proof is done by induction. Since $\Omega_0 = I$, we have $\rho_{\Omega_0} = \rho_I$ and therefore, equation (6) holds for $k = 0$. Let us assume that it holds for some $k \in \{0, \dots, N\}$ and show that it holds for $k+1$. From equation (5) and Proposition 2, it follows that

$$\rho_{\Omega_{k+1}}(\ell) = \rho_{A\Omega_k \oplus V}(\ell) = \rho_{\Omega_k}\left(A^T \ell\right) + \rho_V(\ell).$$

Then, from the induction assumption,

$$\begin{aligned}\rho_{\Omega_{k+1}}(\ell) &= \rho_I\left((A^T)^k A^T \ell\right) + \sum_{i=0}^{k-1} \rho_V\left((A^T)^i A^T \ell\right) + \rho_V(\ell) \\ &= \rho_I\left((A^T)^{k+1} \ell\right) + \sum_{i=0}^{k} \rho_V\left((A^T)^i \ell\right).\end{aligned}$$

Hence, equation (6) holds for $k+1$ as well. By induction, the proposition is proved. ■

In order to compute efficiently $\rho_{\Omega_0}(\ell), \dots, \rho_{\Omega_N}(\ell)$, we introduce the following auxiliary sequences $r_0, \dots, r_N \in \mathbb{R}^d$ and $s_0, \dots, s_N \in \mathbb{R}$:

$$\begin{aligned} r_0 &= \ell,\ \ r_{k+1} = A^T r_k, \\ s_0 &= 0,\ \ s_{k+1} = s_k + \rho_V(r_k). \end{aligned} \tag{7}$$

Equivalently, we have

$$r_k = (A^T)^k \ell \text{ and } s_k = \sum_{i=0}^{k-1} \rho_V\left((A^T)^i \ell\right).$$

Therefore,

$$\rho_{\Omega_k}(\ell) = \rho_I(r_k) + s_k.$$

Algorithm 1 implements the evaluation of the support functions $\rho_{\Omega_0}, \dots, \rho_{\Omega_N}$.

Algorithm 1 Evaluation of the support functions $\rho_{\Omega_0}, \dots, \rho_{\Omega_N}$.

Input: The matrix A, the support functions ρ_I and ρ_V, the vector ℓ and an integer N.
Output: $Y_k = \rho_{\Omega_k}(\ell)$ for k in $\{0, \dots, N\}$

```
r_0 ← ℓ
s_0 ← 0
Y_0 ← ρ_I(r_0)
for k from 0 to N − 1 do
    r_{k+1} ← A^T r_k
    s_{k+1} ← s_k + ρ_V(r_k)
    Y_{k+1} ← ρ_I(r_{k+1}) + s_{k+1}
end for
return {Y_0, ..., Y_N}
```

Algorithm 1 performs, at each of its N iterations, a linear transformation on a vector and the evaluation of the support functions ρ_I and ρ_V. The global complexity of Algorithm 1 is therefore $\mathcal{O}(N(d^2 + \mathcal{C}_I + \mathcal{C}_V))$ where $\mathcal{C}_I$ and $\mathcal{C}_V$ denote the complexity of evaluating ρ_I and ρ_V, respectively [2]. Let us remark that the complexity of Algorithm 1 is linear in the time horizon N and polynomial in d; this is comparable to the complexity of the most competitive algorithms (Girard et al. (2006); Kurzhanskiy and Varaiya (2007)).

Then, the computation of the tight over-approximations of the reachable sets $\tilde{\Omega}_0, \dots, \tilde{\Omega}_N$ defined as intersections of r halfspaces has overall complexity

$$\mathcal{O}(rN(d^2 + \mathcal{C}_I + \mathcal{C}_U)).$$

In the following, we show how the efficiency of the algorithm can be further improved.

3.2 Improvements

An important advantage of Algorithm 1 is that it can be trivially parallelized. Indeed, the support function can be evaluated independently in the different directions $\ell_1, \dots, \ell_r$. Thus, running the reachability analysis on α processors makes the overall complexity drops to:

$$\mathcal{O}\left(\left\lceil \frac{r}{\alpha} \right\rceil N(d^2 + \mathcal{C}_I + \mathcal{C}_U)\right).$$

The following improvement is more sophisticated. Let us assume that the different directions of approximation $\ell_1, \dots, \ell_r$ have been chosen such that:

$$\ell_i = (A^T)^{j_i} \ell,\ i = 1, \dots, r$$

[2] If I and U are low order full dimensional zonotopes or ellipsoids, the complexity becomes $\mathcal{O}(Nd^2)$.

where $0 = j_1 < j_2 < \cdots < j_r$. Then, from Proposition 3, it follows that for all $i = 1, \ldots, r$:

$$\begin{aligned}\rho_{\Omega_k}(\ell_i) &= \rho_I\left((A^T)^k \ell_i\right) + \sum_{p=0}^{k-1} \rho_V\left((A^T)^p \ell_i\right)\\ &= \rho_I\left((A^T)^k (A^T)^{j_i} \ell\right) + \sum_{p=0}^{k-1} \rho_V\left((A^T)^p (A^T)^{j_i} \ell\right)\\ &= \rho_I\left((A^T)^{k+j_i} \ell\right) + \sum_{p=0}^{k-1} \rho_V\left((A^T)^{p+j_i} \ell\right)\\ &= \rho_I\left((A^T)^{k+j_i} \ell\right) + \sum_{p=j_i}^{k+j_i-1} \rho_V\left((A^T)^{p} \ell\right)\\ &= \rho_I(r_{k+j_i}) + s_{k+j_i} - s_{j_i}.\end{aligned}$$

Thus, it is sufficient to compute the sequences $r_0, \ldots, r_{N+j_r}$ and $s_0, \ldots, s_{N+j_r}$. Then, it can be shown that the complexity of the reachability analysis drops to

$$\mathcal{O}((N + j_r)(d^2 + \mathcal{C}_I + \mathcal{C}_U + r)).$$

Let us remark that in that case, the reachability algorithm cannot be parallelized any more.

3.3 Control synthesis using support vectors

We would like to point out a relation of our approach with a class of optimal control problems. Indeed, a slight modification of Algorithm 1 allows us to synthesize optimal control inputs for the linear system (4). When computing the support function of a set in a direction ℓ, it is generally easy to get also an associated support vector (see Section 2). These support vectors can be then used as control inputs to solve the optimal control problem:

$$\text{Maximize } \ell \cdot x_N \qquad (8)$$

under the dynamics of equation (4). Then, a straightforward application of Pontryagin maximum principle (see e.g. Bertsekas (2000)) leads us to the following result:

Proposition 4. The trajectory solving the optimal control problem (8) is obtained for the initial condition $x_0 = \nu_{I,r_N}$ and the sequence of inputs:

$$v_k = \nu_{V,r_{N-k-1}},\ k = 0, \ldots, N-1.$$

Let us remark that Algorithm 1 computes $\rho_I(r_N)$ and $\rho_V(r_0), \ldots, \rho_V(r_{N-1})$. Computing, in addition the associated support vectors allows us to solve the optimal control problem (8).

3.4 Comparison with a similar approach (Varaiya (1998))

Reachability analysis based on the use of support vectors has already been proposed in Varaiya (1998). We would like to discuss here the differences between the two approaches. In Varaiya (1998), the support functions of the reachable sets $\Omega_0, \ldots, \Omega_N$ are computed recursively using the relation:

$$\rho_{\Omega_{k+1}}(\ell) = \rho_{\Omega_k}(A^T \ell) + \rho_V(\ell).$$

Then, the polyhedral over-approximation $\tilde{\Omega}_k$ is defined as the intersection of the halfspaces:

$$\mathcal{H}_{k,i} = \{x : \ell_{k,i} \cdot x \leq \rho_{\Omega_k}(\ell_{k,i})\},\ i = 1, \ldots, r$$

where $\ell_{k,i} = (A^T)^{N-k} \ell_{N,i}$. Then, the directions used for the approximation are not the same for all Ω_k. There are two reasons that makes this point potentially problematic.

The first reason is numerical. Let us fix $\ell_{N,i}$, then the directions used for the approximation of Ω_k are $\ell_{k,i} = (A^T)^{N-k} \ell_{N,i}$. For simplicity, we assume that the eigenvalue of A^T with largest modulus is real and denote ℓ^* the associated eigenvector. Then for a long time horizon N, all the vectors $\ell_{k,1}, \ldots, \ell_{k,r}$ tends to point towards the direction of ℓ^* when k approaches 0. This means that the polyhedral over-approximation $\tilde{\Omega}_k$ is likely to be ill-conditioned for small values of k. The second reason is more practical. Sometimes, we are not interested in approximating the reachable sets but rather the projection of the reachable sets on an output space. Let us consider, for instance the single output system:

$$\begin{cases} x_{k+1} &= Ax_k + v_k,\ x_0 \in I,\ v_k \in V \\ y_k &= cx_k \end{cases}$$

where $c^T \in \mathbb{R}^d$. Then, in order to compute a tight over-approximation of the sets reachable by $y_0, \ldots, y_N$ it is sufficient to run Algorithm 1 with $\ell = c^T$. Similarly, when dealing with hybrid systems with switching conditions given by hyperplanes, it is interesting to choose the directions of approximation given by the normal vectors to the hyperplanes. Indeed, in that case the over-approximation $\tilde{\Omega}_k$ intersects the hyperplane if and only if Ω_k intersects it (see Girard et al. (2006)).

The main advantage of the algorithm presented in (Varaiya (1998)) over Algorithm 1 is that it can be extended very easily to time-varying linear systems. Algorithm 1 does not extend to this class of systems as Proposition 3 holds only for time-invariant linear systems.

4. REACHABILITY OF CONTINUOUS-TIME SYSTEMS

In this section, we show how a similar approach can be used for reachability analysis of continuous-time systems of the form:

$$\dot{x}(t) = Ax(t) + v(t),\ x(0) \in I,\ v(t) \in V \qquad (9)$$

where $I \subseteq \mathbb{R}^d$ and $V \subseteq \mathbb{R}^d$ are compact convex sets. We denote by $\Omega(t)$ the set of reachable points at time t. Similar to the discrete-time case, we want to express $\rho_{\Omega(t)}$ as a function of ρ_I and ρ_V.

Proposition 5. For all $t \in \mathbb{R}^+$,

$$\rho_{\Omega(t)}(\ell) = \rho_I\left(e^{tA^T} \ell\right) + \int_0^t \rho_V\left(e^{\tau A^T} \ell\right) d\tau. \qquad (10)$$

Proof One trajectory of system (9) is given by:

$$x(t) = e^{tA} x(0) + \int_0^t e^{(t-\tau)A} v(\tau) d\tau.$$

Then,

$$\begin{aligned}\ell \cdot x(t) &= \ell \cdot e^{tA} x(0) + \ell \cdot \int_0^t e^{(t-\tau)A} v(\tau) d\tau\\ &= \ell \cdot e^{tA} x(0) + \ell \cdot \int_0^t e^{\tau A} v(t-\tau) d\tau\\ &= x(0) \cdot e^{tA^T} \ell + \int_0^t v(t-\tau) \cdot e^{\tau A^T} \ell d\tau.\end{aligned}$$

Then $\rho_{\Omega(t)}(\ell)$ is obtained by maximizing $\ell \cdot x(t)$ over the initial condition $x(0) \in I$ and input function $v : [0, t] \to V$. Then,

$$\begin{aligned}\rho_{\Omega(t)}(\ell) &= \max_{x(0)} x(0) \cdot e^{tA^T}\ell + \max_{v} \int_0^t v(t-\tau) \cdot e^{\tau A^T}\ell d\tau \\ &= \rho_{\Omega_0}\left(e^{tA^T}\ell\right) + \max_{v} \int_0^t v(t-\tau) \cdot e^{\tau A^T}\ell d\tau\end{aligned}$$

For all $v : [0, t] \to V$, for all $t \in \mathbb{R}^+$ and $\tau \in [0, t]$, we have that

$$v(t-\tau) \cdot e^{\tau A^T}\ell \leq \rho_V\left(e^{\tau A^T}\ell\right)$$

Therefore,

$$\max_{v} \int_0^t v(t-\tau) \cdot e^{\tau A^T}\ell d\tau \leq \int_0^t \rho_V\left(e^{\tau A^T}\ell\right) d\tau$$

Let us consider the input function $v^* : [0, t] \to V$ that associate to $\tau \in [0, t]$, a support vector of V in the direction $e^{(t-\tau)A^T}\ell$. Then, for all τ in $[0, t]$:

$$v^*(t-\tau) \cdot e^{\tau A^T}\ell = \rho_V\left(e^{\tau A^T}\ell\right)$$

It follows that

$$\max_{v} \int_0^t v(t-\tau) \cdot e^{\tau A^T}\ell d\tau \geq \int_0^t \rho_V\left(e^{\tau A^T}\ell\right) d\tau$$

and equation (10) holds. ■

For the practical computation of the support function of the reachable sets, we introduce, similar to the discrete-time case, auxiliary functions $r : \mathbb{R}^+ \to \mathbb{R}^d$ and $s : \mathbb{R}^+ \to \mathbb{R}$ defined by the differential equations:

$$\begin{aligned}\dot{r}(t) &= A^T r(t), \quad r(0) = \ell, \\ \dot{s}(t) &= \rho_V(r(t)), \quad s(0) = 0.\end{aligned} \tag{11}$$

Equivalently, we have

$$r(t) = e^{tA^T}\ell \text{ and } s(t) = \int_0^t \rho_V\left(e^{\tau A^T}\ell\right) d\tau$$

Using Proposition 5, it follows that the support function of the reachable set $\Omega(t)$ can be computed using the following equation

$$\rho_{\Omega(t)}(\ell) = \rho_I(r(t)) + s(t).$$

Hence, the computation of a tight polyhedral over-approximation of the reachable sets can be done by simulating the differential equations (11) for several initial conditions given by the directions used for approximation.

5. EXPERIMENTAL RESULTS

In this section, we show the effectiveness of our approach on some examples. Both discrete-time and continuous-time algorithms have been implemented in OCaml. All computations were performed on a Pentium IV 3.2GHz with 1GB RAM.

5.1 RLC model of a transmission line

The first example we consider is a verification problem for a transmission line borrowed from Han (2005). The goal is to check that the transient behavior of a long transmission line is acceptable both in terms of overshoot and of response time. Figure 2 shows a model of the transmission line, which consists of a number of RLC components (R: resistor, L: inductor, C: capacitor) modelling segments of the line. The left side is the sending end and the right side is the receiving end of the transmission line.

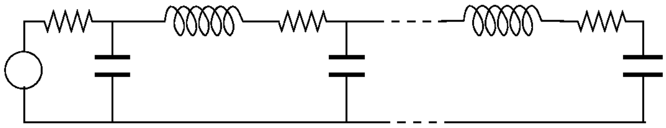

Fig. 2. RLC model of a transmission line

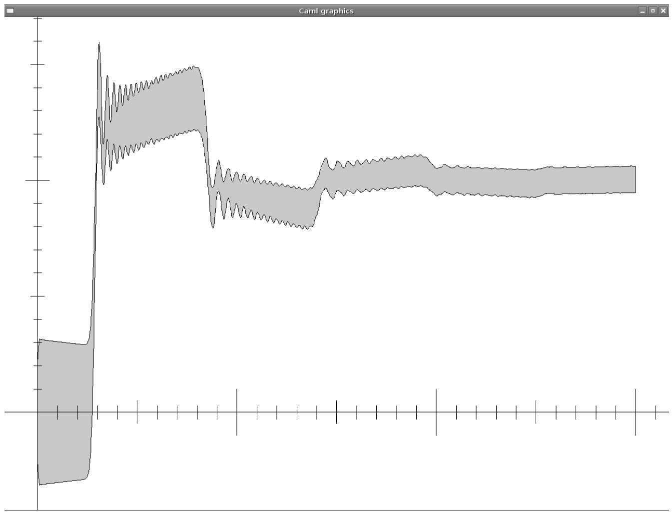

Fig. 3. Reachable tube of $u_{out}(t)$

The dynamics of the system are given by the single-input single-output linear dynamical system

$$\begin{cases} \dot{x}(t) &= Ax(t) + bu_{in}(t), \;\; x(0) \in I, \;\; u_{in}(t) \in U \\ u_{out}(t) &= cx(t) \end{cases}$$

where $x(t) \in \mathbb{R}^d$ is the state vector containing the voltage of the capacitors and the current of the inductors and $u_{in}(t) \in U \subseteq \mathbb{R}$ is the voltage at the sending end. The output of the system is the voltage $u_{out}(t) \in \mathbb{R}$ at the receiving end.

Initially, the system is supposed to be in an ϵ-neighborhood (with $\epsilon = 0.01$) of its steady state for an input voltage inside $[-0.2; 0.2]$. Then, at time $t = 0$, the input voltage is switched to a value in $[0.99; 1.01]$:

$$I = -A^{-1}b[-0.2; 0.2] \oplus \mathcal{B}_\epsilon \qquad U = [0.9; 1.1]$$

The model we considered has dimension 81. Figure 3 shows the reachable tube of the output voltage for a time horizon of 3ns, it was computed in 0.10s using 0.234MB.

5.2 Extensive experiments

Our implementation has also been tested on randomly generated discrete-time examples of different dimension. Tables 1 and 2 summarize the results of our experimentations. We computed a tight over-approximation of the reachable sets $\Omega_0, \ldots, \Omega_{100}$, for random matrices A of dimension d. We either used the algorithm from Girard et al. (2006) (denoted as *direct* in the tables), or Algorithm 1 (*sf*); initial and inputs set were given either as zonotopes of order 1 (*Z*) or ellipsoids (*E*); and the computed tight over-approximation consisted in the intersection of 1, d, or d^2 half-spaces. The program was terminated after 90s.

$d =$	10	20	50	100	200	500
direct Z 1	< 0.01	0.01	0.13	1.00	5.44	85.9
sf Z 1	< 0.01	< 0.01	0.01	0.01	0.05	0.28
direct E 1	< 0.01	0.02	0.27	1.71	11.8	
sf E 1	< 0.01	< 0.01	< 0.01	0.02	0.05	0.31
direct Z d	< 0.01	0.02	0.27	1.86	11.4	
sf Z d	< 0.01	0.02	0.23	1.5	11.1	
direct E d	0.01	0.04	0.41	2.82	21.9	
sf E d	< 0.01	0.02	0.19	1.48	8.98	
direct Z d^2	0.04	0.35	7.38	90.6		
sf Z d^2	0.04	0.36	9.83			
direct E d^2	0.03	0.26	6.69			
sf E d^2	0.03	0.32	9.16			

Table 1. Execution time (in seconds) for $N = 100$ for several linear systems of different dimensions

$d =$	10	20	50	100	200	500
direct Z 1	0.234	0.234	0.234	0.703	2.258	13.43
sf Z 1	0.234	0.234	0.234	0.469	1.480	8.707
direct E 1	0.234	0.234	0.469	1.172	4.961	
sf E 1	0.234	0.234	0.234	0.703	2.332	12.53
direct Z d	0.234	0.234	0.469	1.172	3.195	
sf Z d	0.234	0.234	0.234	0.703	2.184	
direct E d	0.234	0.469	0.937	3.281	7.332	
sf E d	0.234	0.234	0.234	0.703	3.035	
direct Z d^2	0.703	2.812	18.28	77.81		
sf Z d^2	0.234	0.469	3.75			
direct E d^2	0.703	3.047	18.98			
sf E d^2	0.234	0.469	3.75			

Table 2. Memory consumption (in MB) for $N = 100$ for several linear systems of different dimensions

We can see that the new algorithm has great performances for systems with a single output: it can compute exact bounds on this output for the first 100 timesteps in less than a third of a second for a 500 dimensionnal system, while the fastest previoulsy known algorithm, to the best of the authors knowledge, takes more than a minute. For a larger number of directions of tightness, Algorithm 1 compares well to one of the most competitive algorithms.

6. CONCLUSION

We have presented an efficient algorithm for computing reachable sets of discrete-time and continuous-time linear systems with constrained initial states and inputs. We showed that it can handle arbitrary compact convex sets, and can be used to solve a class of optimal control problem. For single ouput systems, it is faster than other algorithms by a factor d, which allows it to handle hundreds of dimensions in a fraction of a second. For more general problems, it is one of the fastest algorithms and can even be further improved by parallelization and by carefully choosing the directions of approximations.

ACKNOWLEDGEMENTS

The authors would like to thank Thao Dang, Goran Frehse and Oded Maler for numerous valuable discussions on reachability analysis and Zhi Han for providing them with transmission line example.

REFERENCES

E. Asarin, T. Dang, O. Maler, and O. Bournez. Approximate reachability analysis of piecewise-linear dynamical systems. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 20–31. Springer, 2000.

A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In *Hybrid Systems: Computation and Control*, volume 1569 of *LNCS*, pages 31–45. Springer, 1999.

D.P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, 2000.

D.P. Bertsekas, A. Nedic, and A.E. Ozdaglar. *Convex analysis and optimization.* Athena Scientific, 2003.

S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

A. Chutinan and B.H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid Systems: Computation and Control*, volume 1569 of *LNCS*, pages 76–90. Springer, 1999.

E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* MIT Press, 2000.

T. Dang. *Vérification et Synthèse des Systèmes Hybrides.* PhD thesis, Institut National Polytechnique de Grenoble, 2000.

A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, pages 291–305. Springer, 2005.

A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 257–271. Springer, 2006.

Z. Han. *Formal Verification of Hybrid Systems using Model Order Reduction and Decomposition.* PhD thesis, Department of ECE, Carnegie Mellon University, 2005.

W. Kühn. Towards an optimal control of the wrapping effect. In *SCAN 98, Developments in Reliable Computing*, pages 43–51. Kluwer, 1999.

A.B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 202–214. Springer, 2000.

A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete time linear systems. *IEEE Trans. Automatic Control*, 52(1):26–38, 2007.

I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 310–323. Springer, 2000.

O. Stursberg and B.H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*, pages 482–497. Springer, 2003.

C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification and control of hybrid systems. *Proc. of the IEEE*, 91(7):986–1001, 2003.

P. Varaiya. Reach set computation using optimal control. In *Proc. KIT Workshop on Verification of Hybrid Systems.* Verimag, Grenoble, 1998.