Reachability Analysis of Linear Systems with Stepwise Constant Inputs

Paul Hänsch[1] Hilal Diab[1] Ibtissem Ben Makhlouf [1]
Stefan Kowalewski [1]

Embedded Software Laboratory (Informatik 11)
RWTH Aachen University
52074 Aachen, Germany

Abstract

Reachability analysis is one major approach for safety verification of continuous and hybrid dynamical systems. In this paper we present a new approach to calculate the reachable states of linear systems with uncertain inputs under the assumption that the inputs are stepwise constant. The original system S with inputs is transformed into a system S' without inputs such that the reachability problem of S can be reformulated as a problem that involves only S' and thus the inputs need no longer to be considered. Finally, we show that this approach is in accordance with existing ones.

Keywords: Linear systems, reachability analysis, safety verification, uncertain input, zonotopes.

1 Introduction

In safety critical applications, simulations can reveal errors in the system design, but in general they are not sufficient to verify safety. One approach to formal verification, common in the field of continuous and hybrid dynamic systems, is the reachability analysis which aims at computing the reachable states, taking into account all possible initial states and inputs.

In general, the exact set of all reachable states of a dynamic system cannot be computed (see [8,15]). However, for special classes of dynamic systems, approximation techniques are applied to overcome this problem by calculating over- or underapproximations of the reachable states.

[1] Email: [haensch diab makhlouf kowalewski]@embedded.rwth-aachen.de

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Common datastructures used to represent the approximations of reachable states are boxes ([16]), polytopes ([7]), polyhedra ([4,3]), level sets, ellipsoids ([2,10]), zonotopes ([5,1]) and support functions ([12],[6]), each of which has certain advantages and drawbacks. Boxes for example, being very simple datastructures, introduce larger approximation errors than others but they are easy to handle. Zonotopes can represent more complex geometric figures and are still one of the most popular datastructures in reachability analysis, but they are not closed under intersections and computing an approximation of the intersection can be expensive or inaccurate or both. In particular, computational complexity and inaccuracy of approximated intersection increase with the representation size of the zonotope. In the case of linear systems with inputs, the representation size of a zonotope increases due to the input to the system. By reshaping the system matrix we map the reachability problem of a system with inputs to a problem that involves only a system without inputs. This approach is not restricted to a special geometric datastructure. We assume stepwise constant inputs, which, however, is relevant in real-world applications where the input of the system is driven by a constant sample time.

Section 2 contains a precise formulation of the problem. In Section 3.1 we explain our approach using an example and after that, in Sections 3.2 and 3.3, the general approach is given. Section 4 deals with the implementation of the presented method and Section 4.1 shows how to include a check for safety constraints. The example in Section 5 demonstrates the presented techniques. In Section 6 we show the equivalence of our method with existing approaches and Section 7 closes with a summary.

2 Problem Statement

In this paper we consider linear time-invariant non-autonomous systems of the form

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1}$$

where $x(t)$ is the system state at time t, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are constant matrices, $u(t) \in \mathbb{R}^m$ is the input to the system at time t bounded by $u(t) \in [\underline{u}_1, \overline{u}_1] \times \ldots \times [\underline{u}_m, \overline{u}_m] =: \mathcal{U}$, for all $t \geq 0$. We assume the input $u : \mathbb{R}_0^+ \to \mathbb{R}^m$ to be stepwise constant with respect to the time step r. We denote the class of all such bounded and stepwise constant (in short *admissible*) inputs by $\mathfrak{U}$.

Definition 2.1 [Trajectory, Reachability] A *trajectory* of system (1) is a time-valued continuous function $x : \mathbb{R}_0^+ \to \mathbb{R}^n$ which satisfies the following condition: There exists an admissible input $u : \mathbb{R}_0^+ \to \mathbb{R}^m$ such that for all $t \geq 0$ (except for those where u is not continuous) the function x satisfies equa-

tion (1).

A state $\xi \in \mathbb{R}^n$ is *reachable* from $\xi_0 \in \mathbb{R}^n$ after time τ if there exists a trajectory x of the system under consideration such that $x(0) = \xi_0$ and $x(\tau) = \xi$.

In the following we introduce the $\mathcal{R}$-operator which will be used throughout this paper.

Definition 2.2 [$\mathcal{R}$-Operator] By $\mathcal{R}(\mathcal{X}, [t_0, t_1])$ we denote the set of all states $x(t)$ reachable by system (1) after a time $t \in [t_0, t_1]$ with initial condition $x(0) \in \mathcal{X}$ under some admissible input $u \in \mathfrak{U}$. If $\mathcal{X} = \{x_0\}$ is a singleton we write $\mathcal{R}(x_0, [t_0, t_1])$ and for $t_0 = t_1$ we write $\mathcal{R}(\mathcal{X}, t_0)$.

Given a set of potential *initial states* $\mathcal{I} \subseteq \mathbb{R}^n$ and a time horizon T, the *reachability problem* consists in computing a (close) overapproximation of the set $\mathcal{R}(\mathcal{I}, [0, T])$

In general it is hopeless to compute the exact set $\mathcal{R}(\mathcal{I}, [0, T])$. Decidability results have been obtained only for very special classes of linear systems, see e.g. [11]. However, depending on the given problem, good over- or underapproximations are often sufficient. A common task is to verify that a given system does not reach a critical state, which can be accomplished by showing that an overapproximation of the reachable states does not intersect with the critical region. Having this motivation in mind, we look for overapproximations of reachable states.

3 Approach

Our idea is to map the reachability problem of a system with input given by equation (1) to the reachability problem of an autonomous system, i.e. a system without the input Bu. Under the assumption that the inputs are stepwise constant, this can be done by shifting the input Bu into the state space. The price we pay for having a system without inputs is that the number of dimensions increases by the number of inputs.

In the following subsection we explain the idea using a simple example. Subsection 3.2 presents the approach in a general setting.

3.1 Sketch of the Idea

Let us consider the following symbolic example of an n-dimensional system,

$$\dot{x}(t) = Ax(t) + u(t), \quad u(t) \in [\underline{\mu}, \overline{\mu}] \times \{0\} \times \ldots \times \{0\}, \tag{2}$$

where only the first component u_1 of the input u is not zero. For system (2) we define its *accompanying* system by

$$\dot{x}'(t) = \underbrace{\left(\begin{array}{c|c} A & \begin{matrix}1\\0\\\vdots\end{matrix} \\ \hline \begin{matrix}0\dots0\end{matrix} & 0 \end{array}\right)}_{A'} x'(t) \tag{3}$$

which has $n+1$ dimensions. The additional state variable x'_{n+1}, which is the last component of the state vector x', substitutes the input. As you can tell from the last row of A' the variable x'_{n+1} does not change over time. This corresponds to our assumption of stepwise constant input. Further, because of $A'_{1,n+1} = 1$ and $A'_{2,n+1} = \ldots = A'_{n+1,n+1} = 0$ the variable x'_{n+1} has an impact only on $\dot{x}'_1$ just as $u(t)$ has an impact only on $\dot{x}_1$. Finally, if we take for $x'_{n+1}(0)$ all values in $[\underline{\mu}, \overline{\mu}]$ as possible initial values into account, then the variables $x'_1, \ldots, x'_n$ can reach the same states as $x_1, \ldots, x_n$, as long as the input $u(t)$ is a constant value in $[\underline{\mu}, \overline{\mu}]$.

In order to formalize the above paragraph, we introduce some further notations. $\mathcal{R}(\cdot,\cdot)$ denotes the reachable states of system (2), i.e. the original system, and $\mathcal{R}'(\cdot,\cdot)$ denotes those of system (3), i.e. the accompanying system without input. Further we define π_k as the projection of a vector onto its first k components. Then, for example $\pi_2((4,5,6)^T) = (4,5)^T$. We extend this projection to sets of vectors by $\pi_k(V) := \{\pi_k(v) \mid v \in V\}$. Then, for example, we have $\pi_2(\{(4,3,2,1)^T, (5,6,7,8)^T\}) = \{(4,3)^T, (5,6)^T\}$.

Now we can say that, under the assumption that the input u is constant in $[0,r]$,

$$\mathcal{R}(\mathcal{I}, \tau) = \pi_n(\mathcal{R}'(\mathcal{I} \times [\underline{\mu}, \overline{\mu}], \tau))$$

holds for all $\tau \in [0,r]$ and hence also

$$\mathcal{R}(\mathcal{I}, [0,r]) = \pi_n(\mathcal{R}'(\mathcal{I} \times [\underline{\mu}, \overline{\mu}], [0,r])) \tag{4}$$

holds.

Time-invariant dynamic systems (as is the case for (2)) satisfy the following property

$$\mathcal{R}(\mathcal{I}, [a+\delta, b+\delta]) = \mathcal{R}(\mathcal{R}(\mathcal{I}, [a,b]), \delta).$$

Now, we can derive

$$\begin{aligned} \mathcal{R}(\mathcal{I}, [ir, ir+r]) &= \mathcal{R}(\mathcal{R}(\mathcal{I}, [ir-r, ir]), r) \\ &= \pi_n(\mathcal{R}'(\mathcal{R}(\mathcal{I}, [ir-r, ir]) \times [\underline{\mu}, \overline{\mu}], r)), \end{aligned} \tag{5}$$

where the first equation follows immediately from the above property, and the

second equation is true under the assumption that, starting from $R(\mathcal{I}, [ir - r, ir])$ the input remains constant for r time units.

Equations (4) and (5) provide a way to reduce the reachability problem for the original system with inputs to the reachability problem of a system without inputs: First, equation (4) can be used to compute $\mathcal{R}(\mathcal{I}, [0, r])$ if we have a method to compute the reachable states $\mathcal{R}'(\cdot, [0, r])$ of the accompanying autonomous system. Second, equation (5) shows how to compute $\mathcal{R}(\mathcal{I}, [ir, ir+r])$ if we have $\mathcal{R}(\mathcal{I}, [ir-r, ir])$ and a method to compute $\mathcal{R}'(\cdot, r)$.

The necessary technique to compute $\mathcal{R}'(\cdot, \cdot)$ will be given in Subsection 3.3.

3.2 The General Approach

We focus on the general form of linear systems given by equation (1) and consider stepwise constant inputs.

For system (1) we define its *accompanying* system by the state vector $x' = (x'_1, \ldots, x'_{n+m})^T$ and the dynamic behavior

$$\dot{x}'(t) = \underbrace{\left(\begin{array}{c|c} A & B \\ \hline 0 \;\cdots\; 0 & \\ \vdots \quad\; \vdots & \\ 0 \;\cdots\; 0 & \end{array}\right)}_{A' :=} x'(t). \tag{6}$$

The variables $x'_1, \ldots, x'_n$ correspond to the original state variables $x_1, \ldots, x_n$. Whereas the variables $x'_{n+1}, \ldots, x'_{n+m}$ are substitutes for the m different inputs.

As before, $\mathcal{R}(\cdot, \cdot)$ denotes the reachable states of the original system (1) and $\mathcal{R}'(\cdot, \cdot)$ those of its accompanying system (6).

Equations (4) and (5) from the previous subsection can be generalized to

$$\mathcal{R}(\mathcal{I}, [0, r]) = \pi_n(\mathcal{R}'(\mathcal{I} \times \mathcal{U}, [0, r])) \tag{7}$$

$$\mathcal{R}(\mathcal{I}, [ir, ir+r]) = \pi_n(\mathcal{R}'(\mathcal{R}(\mathcal{I}, [ir-r, ir]) \times \mathcal{U}), r). \tag{8}$$

Understanding these equations involves the same arguments as have been given in the previous subsection and will not be repeated here. It remains to compute the reachable states of the autonomous accompanying system, which will be done in the next subsection. Then, equation (7) gives $\mathcal{R}(\mathcal{I}, [0, r])$ which can be plugged into equation (8) and gives $\mathcal{R}(\mathcal{I}, [r, 2r])$, which can be plugged again into the same equation and gives the reachable states for $[2r, 3r]$ and so on, until we reach the desired time horizon.

3.3 Computing Reachable States of Linear Autonomous Systems

Given a linear time-invariant autonomous system

$$\dot{x} = Ax$$

we briefly recapitulate methods to compute or overapproximate the operations $\mathcal{R}'(\cdot, \tau)$ and $\mathcal{R}'(\cdot, [0, \tau])$ which are needed in the computation scheme in equations (7) and (8).

The above system of differential equations has the analytical solution

$$x(t) = e^{At} x_0,$$

where $x_0 = x(0)$ is the desired initial condition. It follows that the set of states reachable from a given set $\mathcal{X}$ after time τ is

$$\begin{aligned} \mathcal{R}'(\mathcal{X}, \tau) &= \bigcup_{x_0 \in \mathcal{X}} e^{A\tau} \cdot x_0 \\ &= e^{A\tau} \cdot \mathcal{X}, \end{aligned} \tag{9}$$

which is simply a linear transformation of $\mathcal{X}$ and the matrix $e^{A\tau}$ can be computed by standard numerical tools with a high degree of accuracy (see [13] for the underlying algorithms).

Accordingly, we have

$$\mathcal{R}'(\mathcal{X}, [0, \tau]) = \bigcup_{t \in [0,\tau]} e^{At} \cdot \mathcal{X}.$$

As proposed in [5], this set can be overapproximated by

$$\begin{aligned} \mathcal{R}'(\mathcal{X}, [0, \tau]) &\subseteq CH(\mathcal{I}, e^{A\tau}\mathcal{I}) \oplus \{x \mid \|x\|_v \leq \alpha\}, \\ \alpha &= (e^{\|A\|_m \tau} - 1 - \|A\|_m \tau) \cdot \sup_{x \in \mathcal{X}} \|x\|_v, \end{aligned} \tag{10}$$

where CH denotes the convex hull of its arguments, $\oplus$ is Minkowski sum of sets defined by $S \oplus T := \{s + t \mid s \in S, t \in T\}$ and $\|\cdot\|_m$ is a matrix norm which has to be submultiplicative [2] and also consistent [3] with the vector norm $\|\cdot\|_v$.

Equation (9) and inequality (10) complete the general computation scheme given in equations (7) and (8). The remaining details depend on the data-structures used in an implementation of this method.

[2] i.e. $\|AB\|_m \leq \|A\|_m \|B\|_m$ for all matrices $A, B \in \mathbb{R}^{n \times n}$
[3] i.e. $\|Ax\|_v \leq \|A\|_m \|x\|_v$ for all $A \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^n$

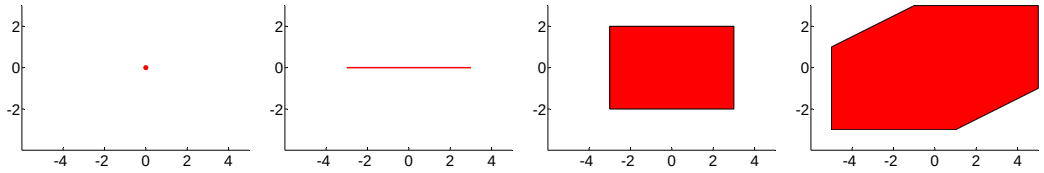

Fig. 1. Some examples for zonotopes in 2-dimensional space. From left to right: $\binom{0}{0}$ has no generators and represents the singleton $\{\binom{0}{0}\}$, $\left(\begin{smallmatrix}0 & 3\\ 0 & 0\end{smallmatrix}\right)$ represents a straight line segment, $\left(\begin{smallmatrix}0 & 3 & 0\\ 0 & 0 & 2\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}0 & 3 & 0 & 2\\ 0 & 0 & 2 & 1\end{smallmatrix}\right)$ both expand over two dimensions.

4 Implementation Using Zonotopes

The computation scheme presented in the previous section can be implemented with different data structures. Here, we show an implementation using zonotopes, i.e. we will overapproximate each of the sets of reachable states $\mathcal{R}(\mathcal{I}, [ir, ir + r])$ with a zonotope. Afterwards we propose a way to efficiently check safety constraints of the form $\wedge_i \Sigma_j \alpha_{ij} x_j \leq c_i$, i.e. conjunctions of linear inequalities over the state variables x_j with arbitrary constant coefficients α_{ij}.

Definition 4.1 [Zonotope] A *zonotope* Z is a tuple $Z = (c, g_1, \ldots, g_k)$ with $c, g_1, \ldots, g_k \in \mathbb{R}^n$ and $k \geq 0$. The semantics of Z is the following set

$$\{c + \Sigma_{i=1}^{k} \alpha_i \cdot g_i \mid -1 \leq \alpha_i \leq 1 \text{ for all } i = 1, \ldots, k\} \subseteq \mathbb{R}^n.$$

The parameter c is called the *center* and $g_1, \ldots, g_k$ are the *generators* of the zonotope.

The term *zonotope* can refer to the tuple and also to the set defined by its semantics, which should be clear from the context.

To simplify the notation, we will sometimes write a zonotope as a matrix, where the first column represents the center and the successing columns represent the generators of the zonotope. Figure 1 shows several examples of zonotopes together with their matrix representation.

In the following we give implementations of the operations needed in the the computation scheme from Section 3.2 (see equations (7) and (8)).

One common way to specify the initial region $\mathcal{I}$ and also the inputs $\mathcal{U}$ is to use an interval[4]. Each interval $\mathcal{I} = [\underline{a}_1, \overline{a}_1] \times \ldots \times [\underline{a}_n, \overline{a}_n]$ is also a zonotope

[4] The term *interval* is used also for higher-dimensional intervals like $[\underline{a}_1, \overline{a}_1] \times \ldots \times [\underline{a}_n, \overline{a}_n]$. Another common name for intervals is *axis aligned box*.

which can be written as

$$\mathcal{I} = \left(\begin{array}{c|ccc} \frac{\overline{a}_1+\underline{a}_1}{2} & \frac{\overline{a}_1-\underline{a}_1}{2} & & \\ \vdots & & \ddots & \\ \frac{\overline{a}_n+\underline{a}_n}{2} & & & \frac{\overline{a}_1-\underline{a}_1}{2} \end{array}\right)$$

where except for the first column, the matrix is diagonal, empty entries are 0. Hence, the implementation starts by transforming $\mathcal{I}$ and $\mathcal{U}$ into a zonotope.

Given two zonotopes $Z_1 = (c_1, g_1^1, \ldots, g_1^k)$ and $Z_2 = (c_2, g_2^1, \ldots, g_2^k)$ the set $Z_1 \times Z_2$ can be represented by

$$Z_1 \times Z_2 = \left(\begin{array}{c|ccc|c} c_1 & g_1^1 & \ldots & g_1^k & 0 \\ \hline c_2 & 0 & & & g_2^1 \ \ldots \ g_2^k \end{array}\right).$$

We use this formula to compute $\mathcal{I} \times \mathcal{U}$ (see equation (7)), after having transformed $\mathcal{I}$ and $\mathcal{U}$ into zonotopes if necessary.

Next, we have to compute $\mathcal{R}'(Z, [0, r])$ for a given zonotope $Z = (c, g_1, \ldots, g_k)$. According to equation (10) we first need the convex hull of Z and $e^{Ar}Z$. Unfortunately, the convex hull of two zonotopes does not need to be a zonotope[5]. In [5] the following zonotope overapproximation is proposed:

$$\begin{aligned} CH(Z, e^{Ar}Z) \subseteq (& \frac{c + e^{Ar}c}{2}, \frac{g_1 + e^{Ar}g_1}{2}, \ldots, \frac{g_k + e^{Ar}g_k}{2}, \\ & \frac{c - e^{Ar}c}{2}, \frac{g_1 - e^{Ar}g_1}{2}, \ldots, \frac{g_k - e^{Ar}g_k}{2}). \end{aligned}$$

Note that it contains $2k+1$ generators. If Z is simply an interval[6] and r is sufficiently small, a closer zonotope overapproximation can be obtained by computing the interval hull[7] of Z and $e^{Ar}Z$:

$$IH(Z, e^{Ar}Z) = [c_1 - \sum_{i=1}^{k} |g_i^1|, c_1 + \sum_{i=1}^{k} |g_i^1|] \times \ldots \times [c_k - \sum_{i=1}^{k} |g_i^k|, c_k + \sum_{i=1}^{k} |g_i^k|]$$

and transforming it into a zonotope Z_H. The resulting zonotope Z_H has only n generators. Back to equation (10), we have to compute the Minkowski sum of Z_H and $\{x \mid \|x\|_v \leq \alpha\}$. Therefore, we have to agree on a vector norm $\|\cdot\|_v$.

[5] By definition, each zonotope is symmetric about its center and hence, the convex hull of e.g. the line segment $\left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and the singleton $\binom{1}{0}$ is not a zonotope.
[6] If $\mathcal{I}$ and $\mathcal{U}$ are both intervals then so is $Z = \mathcal{I} \times \mathcal{U}$.
[7] i.e. a tight interval overapproximation

If using zonotopes, it is handy to use the infinity norm $\|x\|_\infty := \max_i |x_i|$. In that case

$$\{x \mid \|x\|_\infty \leq \alpha\} = [-\alpha, \alpha] \times \ldots \times [-\alpha, \alpha],$$

is an interval (with α as in equation (10)) and thus a zonotope.

The Minkowski sum of two zonotopes $Z_1 = (c_1, g_1^1, \ldots, g_1^k)$ and $Z_2 = (c_2, g_2^1, \ldots, g_2^k)$ is simply $Z_1 + Z_2 = (c_1 + c_2, g_1^1, \ldots, g_1^k, g_2^1, \ldots, g_2^k)$. Altogether, we have a zonotope overapproximation of $\mathcal{R}'(Z, [0, r])$.

The transformation of a zonotope $Z = (c, g_1, \ldots, g_k)$ by a linear map represented by the matrix A is again a zonotope and can be written as $A \cdot Z = (Ac, Ag_1, \ldots, Ag_k)$. This covers the transformation of $\mathcal{R}(\mathcal{I}, [ir-r, ir])$ by $e^{rA'}$.

The projection $\pi_n(Z)$ is obtained by applying π_n to the center and each generator of Z. The matrix representation of $\pi_n(Z)$ is equal to the first n rows of the matrix representation of Z. In general, the projection can produce generators that have all entries equal 0. These generators can be deleted to reduce the size of the zonotope.

To sum it up, all involved operations have polynomial runtime. But the number of generators increases in each iteration by the number of inputs, this is due to the operation $\mathcal{R}(\mathcal{I}, [ir-r, r]) \times \mathcal{U}$.

4.1 Checking Safety Constraints

Given the dynamic system (1) with state variables $x_1, \ldots, x_n$ and a safety constraint $\wedge_i \Sigma_j \alpha_{ij} x_j \leq c_i$ over the state variables, we compute iteratively the sets $\mathcal{R}(\mathcal{I}, [0, r])$, $\mathcal{R}(\mathcal{I}, [r, 2r])$, ... and in each step we can check the safety constraint to verify that the system cannot reach a critical state under any initial condition $x \in \mathcal{I}$ and under any possible input u.

Lemma 4.2 *Given a zonotope $Z = (c, g_1, \ldots, g_k)$ and a safety constraint $\Sigma_i \alpha_i x_i \leq b$. Then there exists $z \in Z$ such that $\sum_i \alpha_i z_i > b$, if and only if $(\alpha_1 \ldots \alpha_n) \cdot c + \sum_i |(\alpha_1 \ldots \alpha_n) \cdot g_i| > b$.*

Proof. Consider $Z' = (\alpha_1 \ldots \alpha_n) \cdot Z$, a linear transformation of Z by a one-row-matrix. The result is a zonotope in $\mathbb{R}$ which is a one-dimensional interval. At first we show: $\exists z \in Z : \sum_i \alpha_i z_i > b \Leftrightarrow \exists z' \in Z' : z' > b$.

"$\Rightarrow$" Assume $z \in Z$ with $\sum_i \alpha_i z_i > b$
$\Rightarrow z' := \sum_i \alpha_i z_i = (\alpha_1 \ldots \alpha_n) \cdot z$ is in Z' and satisfies $z' > b$.

"$\Leftarrow$" Assume $z' \in Z'$ and $z' > b$
$\Rightarrow$ there must be $z \in Z$ such that $z' = (\alpha_1 \ldots \alpha_n) \cdot z$
$\Rightarrow \sum_i \alpha_i z_i = (\alpha_1 \ldots \alpha_n) \cdot z = z' > b$

Thus, it suffices to check whether there is $z' \in Z'$ with $z' > b$. On the other hand, Z' is a one-dimensional interval with maximum value

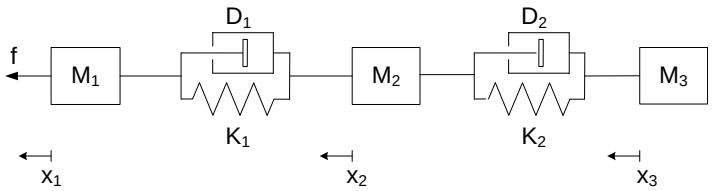

Fig. 2. Mechanical dynamic system consisting of point masses M_i at positions x_i, ideal springs with spring constants K_i and ideal viscous dampers with damping constants D_i.

$(\alpha_1 \dots \alpha_n) \cdot c + \sum_i |(\alpha_1 \dots \alpha_n) \cdot g_i|$. So it suffices to check if this maximum value is $> b$. □

After the computation of each $\mathcal{R}(\mathcal{I}, [ir, ir{+}r])$ we use the above method to check the safety constraints in time $\mathcal{O}(nk)$ with n being the dimension of the system and k the number of generators of the zonotope overapproximating $\mathcal{R}(\mathcal{I}, [ir, ir{+}r])$.

5 Example

We consider a system of three point masses interconnected by springs and dampers (see Figure 2). Here we make two assumptions: The springs follow Hooke's Law and the dampers are ideal viscous dampers. The position variables x_i of the masses M_i are defined such that $x_1 = x_2 = x_3$ holds when the springs are in their idle state (i.e. not compressed and not expanded). The force f applied to M_1 is the input to this dynamic system.

From Newton's Second Law we know that $\Sigma F = M \cdot a$ where ΣF is the sum of forces applied to the mass M and a is the acceleration of M. Here, the sum of forces applied to, for example, M_1, is $f + K_1(x_2 - x_1) + D_1(\dot{x}_2 - \dot{x}_1)$ in the positive direction of x_1. Therefore

$$f + K_1(x_2 - x_1) + D_1(\dot{x}_2 - \dot{x}_1) = M_1 a_1 = M_1 \ddot{x}_1.$$

The dynamic behavior of M_2 and M_3 can be obtained in the same way. After introducing the state vector $x = (x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3)^T$ the dynamic behavior of the whole system can be written as a six-dimensional linear system with input (see equation (3), where, for example, the second line reflects the dynamic behavior of M_1 as given in the above equation).

Mechanic systems of this type can be used to model and verify the safety of controllers for the longitudinal dynamics of vehicles in a platoon (see [9]).

Figure 4 shows the results of a simulation where all constants of the system have been set to 1, carried out with MATLAB/Simulink software. The force f, in other words the input to the system (solid line in figure 4) is one sine wave oscillation followed by constant zero. The system responds with oscillating velocities and positions, where M_1 naturally reacts faster than M_2, which in

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_1}{M_1} & -\frac{D_1}{M_1} & \frac{K_1}{M_1} & \frac{D_1}{M_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{K_1}{M_2} & \frac{D_1}{M_2} & -\frac{K_1+K_2}{M_2} & -\frac{D_1+D_2}{M_2} & \frac{K_2}{M_2} & \frac{D_2}{M_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{K_2}{M_3} & \frac{D_2}{M_3} & -\frac{K_2}{M_3} & -\frac{D_2}{M_3} \end{pmatrix} x + \begin{pmatrix} 0 \\ \frac{1}{M_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} f$$

Fig. 3. State space model of the mechanical system with $x = (x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3)^T$.

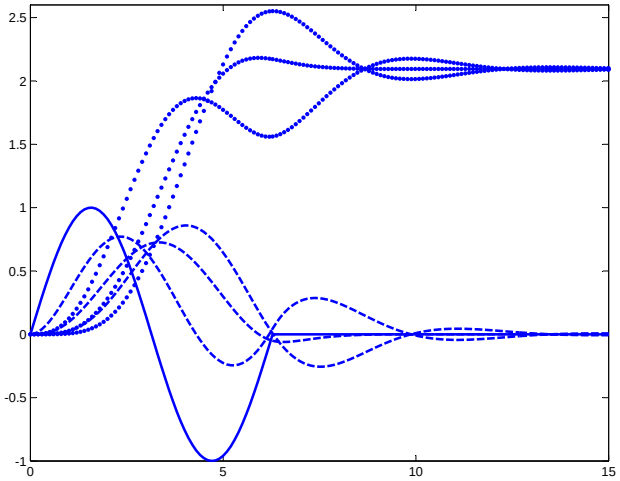

Fig. 4. Simulation of the mechanical system with input force f (solid line), positions x_i (dotted lines) and velocities $\dot{x}_i$ (dashed lines) of the point masses M_i over time t.

turn reacts faster than M_3. After the force f reaches and stays at zero, all positions converge towards the same constant value and the velocities converge to zero.

A safety relevant question might be whether the distances $x_1 - x_2$ and $x_2 - x_3$ are within given bounds, if we assume, e.g., $f \in [-1, 1]$. It does not suffice to run simulations. Instead, we can use the approach given in the previous section to check the safety constraints. However, reachability analysis is only a bounded model checking technique and we have to fix a time horizon T up to which the computations should be carried out.

For this example we assume the input to be bounded by $f \in [-1, 1]$ and to keep it simple, we consider only one initial state, namely $(0, \ldots, 0)^T$. The time step is chosen to be $r = 0.01$ and the time horizon $T = 30$. To analyze the possible behaviors of the system, we plot after each step, i.e. for each computed $\mathcal{R}(\mathcal{I}, [ir, ir + r])$ the biggest possible distance $x_1 - x_2$ among the currently reachable states in $\mathcal{R}(\mathcal{I}, [ir, ir + r])$. According to Lemma 4.2 this biggest

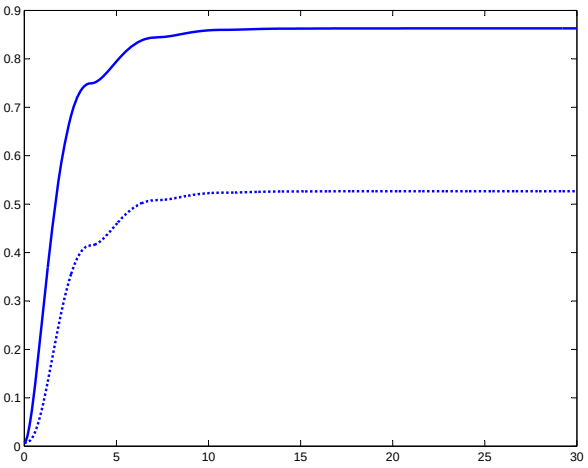

Fig. 5. Result of reachability analysis: Biggest possible distance $x_1 - x_2$ (solid line) and $x_2 - x_3$ (dashed line) over time t.

possible distance is equal to $(1\ 0\ -1\ 0\ 0\ 0) \cdot c + \sum_i |(1\ 0\ -1\ 0\ 0\ 0) \cdot g_i|$ where c is the center and $g_1, \ldots, g_k$ are the generators of the zonotope representing $\mathcal{R}(\mathcal{I}, [ir, ir{+}r])$. Figure 5 shows the result and we see, that the biggest possible distance $x_1 - x_2$ is around 0.85 and $x_2 - x_3$ is around 0.5. In particular, the biggest possible distances can be reached within around 10 time units. Further reachability computations with bigger time horizon (100) and smaller time step (0.001) produced quasi identical results.

6 Comparison with Existing Approaches

We briefly recapitulate the common approach to the reachability problem of linear systems. It is known (see e.g. [14]) that the expression

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau$$

is the solution of the initial value problem $x(0) = x_0$ associated with system (1). From this, one can show in a straight forward manner that

$$\mathcal{R}(\mathcal{I}, [0, r]) \subseteq \bigcup_{t\in[0,r]} e^{At}\mathcal{I} \ \oplus \ \bigcup_{t\in[0,r]} \bigcup_{u\in\mathfrak{U}} \int_0^t e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau$$

holds for system (1) where $\mathfrak{U}$ is the class of allowed inputs. An overapproximation of this set and hence of the reachable states in the first time interval $[0, r]$ can be found for example in [6].

To cover the successing time intervals $[r, 2r]$, $[2r, 3r]$,... an iterative scheme

is needed. For time-invariant systems it holds

$$\begin{aligned}\mathcal{R}(\mathcal{I},[ir,ir+r]) &= \mathcal{R}(\mathcal{R}(\mathcal{I},[ir-r,ir]),r)\\ &\subseteq e^{Ar}\cdot\mathcal{R}(\mathcal{I},[ir-r,ir])\oplus\bigcup_{u\in\mathfrak{U}}\int_0^r e^{A(r-\tau)}u(\tau)\mathrm{d}\tau.\end{aligned}$$

In the case of stepwise constant input, the above formula can be simplified to

$$\mathcal{R}(\mathcal{I},[ir,ir+r])\subseteq e^{Ar}\cdot\mathcal{R}(\mathcal{I},[ir-r,ir])+A^{-1}(e^{Ar}-I)B\mathcal{U}. \tag{11}$$

Basically the standard approach is to plug in the overapproximation of $\mathcal{R}(\mathcal{I},[0,r])$ into equation (11) giving an overapproximation of $\mathcal{R}(\mathcal{I},[r,2r])$ which again can be plugged into equation (11) etc. until the desired time horizon T is reached.

Both approaches, the one above and the one proposed in Section 3.2 consist of two formulae, one giving an initial computation and another one for the iteration. The iterative formulae of both approaches are equivalent:

Theorem 6.1 *Equations (11) and (8) are equivalent.*

Proof. Let $A'=\left(\begin{array}{c|c}A & B\\ \hline 0 & 0\end{array}\right)\in\mathbb{R}^{n+m\times n+m}$ as defined in equation (6). By induction it follows easily that $A'^i=\left(\begin{array}{c|c}A^i & A^{i-1}B\\ \hline 0 & 0\end{array}\right)$. By I_k we denote the $k\times k$ identity matrix. Then

$$\begin{aligned}e^{rA'} &= \sum_{i=0}^{\infty}\frac{A'^i r^i}{i!}\\ &= I_{n+m}+\left(\begin{array}{c|c}\sum_{i=1}^{\infty}\frac{A^i r^i}{i!} & \sum_{i=1}^{\infty}\frac{A^{i-1}Br^i}{i!}\\ \hline 0 & 0\end{array}\right)\\ &= \left(\begin{array}{c|c}\sum_{i=0}^{\infty}\frac{A^i r^i}{i!} & (\sum_{i=1}^{\infty}\frac{A^{i-1}r^i}{i!})B\\ \hline 0 & I_m\end{array}\right)\\ &= \left(\begin{array}{c|c}e^{rA} & A^{-1}(\sum_{i=1}^{\infty}\frac{A^i r^i}{i!})B\\ \hline 0 & I_m\end{array}\right) \qquad \text{provided } \det(A)\neq 0\\ &= \left(\begin{array}{c|c}e^{rA} & A^{-1}(e^{rA}-I_n)B\\ \hline 0 & I_m\end{array}\right).\end{aligned} \tag{12}$$

In the following we use the short hand notation $\mathcal{R} := \mathcal{R}(\mathcal{I}, [ir-r, ir])$ and $\mathcal{U} := [\underline{u}_1, \overline{u}_1] \times \ldots \times [\underline{u}_m, \overline{u}_m]$. For matrices $P \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times m}$ it holds

$$\begin{aligned}\left(\begin{array}{c|c} P & Q \\ \hline 0 & I_m \end{array}\right) \cdot (\mathcal{R} \times \mathcal{U}) &= \left\{ \left(\begin{array}{c|c} P & Q \\ \hline 0 & I_m \end{array}\right) \begin{pmatrix} r \\ u \end{pmatrix} \mid r \in \mathcal{R}, u \in \mathcal{U} \right\} \\ &= \left\{ \begin{pmatrix} Pr + Qu \\ u \end{pmatrix} \mid r \in \mathcal{R}, u \in \mathcal{U} \right\}.\end{aligned}$$

Consequently

$$\pi_n\left(\left(\begin{array}{c|c} P & Q \\ \hline 0 & I_m \end{array}\right) \cdot (\mathcal{R} \times \mathcal{U}) \right) = P\mathcal{R} + Q\mathcal{U}. \tag{13}$$

Our claim follows from equations (12) and (13):

$$\begin{aligned}&\pi_n(e^{rA'}(\mathcal{R} \times \mathcal{U})) \\ &= e^{rA}\mathcal{R} + A^{-1}(e^{rA} - I_n)B\mathcal{U}\end{aligned}$$

□

7 Summary

By reshaping the system matrix we mapped the reachability problem of linear systems with inputs to a problem over autonomous linear systems. We proposed an implementation using zonotopes. However, this approach is not restricted to a certain geometric datastructure. The reachability analysis was demonstrated at a practical example including the verification of safety constraints. Finally, we showed that the performance of this method is in accordance with the standard approach to the reachability analysis of linear systems with stepwise constant inputs.

References

[1] M. Althoff, O. Stursberg, and M. Buss. Verification of uncertain embedded systems by computing reachable sets based on zonotopes. In *Proc. of the 17th IFAC World Congress*, 2008.

[2] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *HSCC. Volume 1790 of LNCS.*, pages 73–88. Springer, 2000.

[3] O. Bournez, O. Maler, and A. Pnueli. Orthogonal polyhedra: Representation and computation, 1999.

[4] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past hytech. In *HSCC 2005, LNCS 3414*, pages 258–273. Springer-Verlag, 2005.

[5] A. Girard. Reachability of uncertain linear systems using zonotopes. In Springer, editor, *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, 2005.

[6] Colas Le Guernic and Antoine Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250 – 262, 2010. IFAC World Congress 2008.

[7] Z. Han and B. H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In *HSCC*, pages 287–301, 2006.

[8] T.A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. *Computer and Systems Sciences*, 57:94–124, 1998.

[9] P. Hänsch H. Diab S. Kowalewski I. Ben Makhlouf, J. Maschuw and D. Abel. Safety verification of a cooperative vehicle platoon with uncertain inputs using zonotopes. In *submitted to 18th IFAC World Congress*, 2010.

[10] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *HSCC '00: Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control*, pages 202–214, London, UK, 2000. Springer-Verlag.

[11] Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. A new class of decidable hybrid systems. In *HSCC*, pages 137–151, 1999.

[12] C. Le Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In *CAV '09: Proceedings of the 21st International Conference on Computer Aided Verification*, pages 540–554, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

[14] Lawrence Perko. *Differential Equations and Dynamical Systems*. Springer; 3rd edition, 2006.

[15] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In *Proceedings of the 6th International Conference on Computer Aided Verification (CAV 1994), LNCS 818*, pages 95–104. Springer-Verlag, 1994.

[16] S. Ratschan and Z. She. Constraints for continuous reachability in the verification of hybrid systems. In *Proc. 8th Int. Conf. on Artif. Intell. and Symb. Comp., AISC'2006*, number 4120 in LNCS, pages 196–210. Springer, 2006.