Avoiding Geometric Intersection Operations in Reachability Analysis of Hybrid Systems

Matthias Althoff
malthoff@ece.cmu.edu

Bruce H. Krogh
krogh@ece.cmu.edu

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

Although a growing number of dynamical systems studied in various fields are hybrid in nature, the verification of properties, such as stability, safety, etc., is still a challenging problem. Reachability analysis is one of the promising methods for hybrid system verification, which together with all other verification techniques faces the challenge of making the analysis scale with respect to the number of continuous state variables. The bottleneck of many reachability analysis techniques for hybrid systems is the geometrically computed intersection with guard sets. In this work, we replace the intersection operation by a nonlinear mapping onto the guard, which is not only numerically stable, but also scalable, making it possible to verify systems which were previously out of reach. The approach can be applied to the fairly common class of hybrid systems with piecewise continuous solutions, guard sets modeled as halfspaces, and urgent semantics, i.e. discrete transitions are immediately taken when enabled by guard sets. We demonstrate the usefulness of the new approach by a mechanical system with backlash which has 101 continuous state variables.

Categories and Subject Descriptors

G.1.0 [**Numerical Analysis**]: General; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Algorithms, Theory, Verification

Keywords

Reachability Analysis, Hybrid Systems, Guard Intersection, Zonotopes, Safety

1. INTRODUCTION

Reachability analysis essentially provides the set of states that a hybrid system can reach in finite or infinite time. In contrast to numerical simulation, reachable set algorithms compute the reachable states starting from sets with infinitely many initial states and in many cases also sets of infinitely many inputs/disturbances and parameters. When the reachable set is obtained in an overapproximative way (all reachable states are enclosed), reachability analysis serves as a formal method for safety verification and other control problems: invariant computation [27], abstraction to discrete systems [9], guaranteed state observation [29], and the like.

Typically, the main difficulty for the hybrid (combined discrete and continuous) reachability problem lies in the continuous part, requiring set operations in continuous space. Over the years, many representations for continuous reachable sets have been introduced, each of them having advantages for specific types of system dynamics: ellipsoids [20], polytopes [8], oriented rectangular hulls [30], zonotopes [12], zonotope bundles [2], support functions [13], level sets [23], and others. For reachability problems with large discrete state spaces, specialized representations are also used for sets of discrete states, see e.g. [10].

Zonotopes [4, 15] and support functions [13] have shown great performance and scalability for purely continuous systems, but are challenging for intersections with guard sets. In this paper, we consider guard sets modeled as polyhedra, which is the most commonly used modeling formalism. Even reachable sets represented by polytopes (bounded polyhedra), which are closed under intersection with polyhedra, are not computationally efficient for guard intersection since partial intersections have to be unified by a convex hull to avoid a combinatorial explosion in the representation size and time required to compute the reachable set. Since the computation of the convex hull is itself costly and in many cases numerically unstable [5], alternative approaches have been proposed for unifying reachable sets with simpler representations, such as two-dimensional projections of zonotopes [14], bundles of parallelotopes (a special case of zonotopes) [2], and template polyhedra [11].

When the guard sets are modeled more generally as level sets, a representation of the intersection of the reachable set with the guard set can be computed by determining the time interval in which an intersection takes place, and using the complete reachable set of this time interval as an overapproximation of the intersection. This approach is often used in so-called *guaranteed integration* methods for hybrid systems, typically enclosing a single trajectory rather than a set of trajectories [24]. To compute tight overapproximations of the reachable set using this method, the set of initial states

HSCC'12, April 17–19, 2012, Beijing, China.

has to be partitioned and the reachable states are computed from each set in the partition, resulting in a possibly large computational load [28].

A completely different approach for reachability analysis of hybrid systems is suggested in [16]. There, only the reachable set propagating from guard intersection to guard intersection is computed by solving linear matrix inequalities (LMIs) embedded in a convex optimization routine, where guards are modeled as hyperplanes. A disadvantage of this approach is that it does not consider uncertain inputs and it is semi-formal, since the LMIs have to be fulfilled for time intervals, whereas only a finite number of points in time can be checked. In this work, we propose an approach which also computes guard set intersections without explicitly computing set intersections. We use hyperplanes to define guard sets, but in contrast to [16], we (i) compute the reachable set for all times in between guard intersections, (ii) present a formal approach, and (iii) use operations that break down to simple operations (addition, multiplication, and division) on reals, instead of solving LMIs. Our algorithm has a complexity of $\mathcal{O}(n^6)$ with respect to the number of continuous state variables n, which can be reduced to $\mathcal{O}(n^3)$ for "simple" dynamics and/or "small" sets of states. Due to the simple operations involved, the algorithm is numerically stable. A requirement for the proposed approach, which is shared with the approach in [16], is that the time interval of a guard intersection for all trajectories within the reachable tube has to be bounded. If the reachable tube only partially intersects a guard set, a case which can be automatically detected, one has to use the classical geometric intersection technique.

2. PROBLEM FORMULATION

We present a technique to compute the reachable set of a hybrid automaton with urgent transition semantics, i.e., a transition is taken as soon as a guard is hit [17]. Due to the urgent transitions, guard sets modeled as hyperplanes or halfspaces enforce the same behavior, but halfspaces are used in order to remove ambiguity when a reachable set hits several guard sets (see Sec. 5.5).

Definition 1 (Hybrid Automaton). *The hybrid automaton used in this work is defined as a tuple* $\mathrm{HA} = (\mathcal{V}, \mathcal{X}, \mathcal{U}, \mathrm{T}, \mathrm{g}, \mathrm{h}, \mathrm{f})$ *with:*

- *the discrete state space* $\mathcal{V} = \{v_1, \ldots, v_\xi\}$. *Elements of* $\mathcal{V}$ *are referred to as locations.*
- *the continuous state space* $\mathcal{X} \subseteq \mathbb{R}^n$ *and input space* $\mathcal{U} \subset \mathbb{R}^m$.
- *the set of discrete transitions* $\mathrm{T} \subseteq \mathcal{V} \times \mathcal{V}$. *A transition from* v_i *to* v_j *is denoted by* (v_i, v_j).
- *the guard function* $\mathrm{g} : \mathrm{T} \to 2^{\mathcal{X}}$ *(*$2^{\mathcal{X}}$ *denotes the power-set of* $\mathcal{X}$*), which associates a guard set* $\mathrm{g}((v_i, v_j))$ *with each transition* (v_i, v_j).
- *the jump function* $\mathrm{h} : \mathrm{T} \times \mathcal{X} \to \mathcal{X}$, *which returns the next continuous state when a transition is taken.*
- *the flow function* $\mathrm{f} : \mathcal{V} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$, *which defines a vector field for the time derivative of* x: $\dot{x} = \mathrm{f}(v, x, u)$.

The guard sets g *are modeled as halfspaces* $\mathcal{H} = \{x | n^T x \leq d;\ x, n \in \mathbb{R}^n;\ d \in \mathbb{R}\}$. *The jump function is restricted to an affine map* $x' = K_{(v_i,v_j)} x + l_{(v_i,v_j)}$, *where* x' *denotes the state after the transition is taken and* $K_{(v_i,v_j)} \in \mathbb{R}^{n \times n}$, $l_{(v_i,v_j)} \in \mathbb{R}^n$ *are given for each transition* (v_i, v_j).

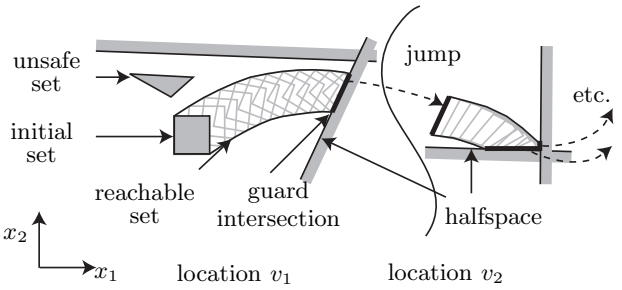


Figure 1: Illustration of the evolution of a reachable set of a hybrid automaton.

In contrast to other definitions of hybrid automata, we do not include invariants since a discrete transition is already forced by the guard itself. Assuming that the hybrid automaton has no Zeno behavior, the combined discrete and continuous state $(v(t), x(t))$ starts from (v^0, x^0) and $x(t)$ changes according to the flow function, while $v(t)$ remains constant. If the continuous state hits a guard set, the corresponding transition is taken immediately. In the event of hitting more than one guard set at the same time, the transition is chosen non-deterministically. After the transition from the previous location v_i to the next location v_j is taken (in zero time), the continuous state is updated according to the jump function and the continuous evolution is continued in the next location. This procedure is also illustrated in Fig. 1. Given the behavior of the hybrid automaton, we are interested in the reachable continuous states:

Definition 2 (Exact Reachable Set). *Given an initial location* v^0 *and a set of initial continuous states* $\mathcal{X}^0$, *the continuous reachable set* $\mathcal{R}^e_{v_i}(r)$ *of a hybrid automaton as specified in Def. 1 at time* r *in location* v_i *is:*

$$\mathcal{R}^e_{v_i}(r) = \Big\{\tilde{x} \big| \exists \text{ some trajectory } (v(t), x(t)) \text{ of HA with } v(0) = v^0, x(0) \in \mathcal{X}^0 \subset \mathcal{X}, \text{ such that } v(r) = v_i,\ x(r) = \tilde{x}\Big\}.$$

The exact reachable set can be computed for only a limited class of hybrid automata [21]. Therefore, we compute over-approximations $\mathcal{R}_{v_i}(t) \supseteq \mathcal{R}^e_{v_i}(t)$. In order to compute for all times, we compute reachable sets for consecutive time intervals $[t_{k-1}, t_k]$, where $t_k = k\,r$, $r \in \mathbb{R}^+$ is the time increment, and $k \in \mathbb{N}$. Next, we explain the basic procedure of computing reachable sets $\mathcal{R}_{v_i}([t_{k-1}, t_k])$ for consecutive time intervals in a single location plus the subsequent computation for determining the initial set of the next location. Since this process is executed repeatedly, it is sufficient to focus on one location, so we drop the location index for $\mathcal{R}(t)$ from now on for simplicity of notation.

3. BASIC PROCEDURE

We begin with the reachable set computation for the continuous evolution and briefly describe the extension to hybrid systems with the classical and the new approach.

3.1 Continuous Dynamics

We first explain the reachable set computation for linear dynamics ($\dot{x} = Ax(t) + u(t)$, $A \in \mathbb{R}^{n \times n}$, $x, u \in \mathbb{R}^n$) and later discuss the extension to nonlinear dynamics. For linear

systems, the reachable set of the first time interval $[t_0, t_1]$ is computed as shown in Fig. 2:

1. Compute the reachable set at $t = t_1$, neglecting uncertain inputs (the homogeneous solution, $\mathcal{R}^h(t_1)$);
2. Generate the convex hull of the solution at $t = t_1$ and the initial set; and
3. Enlarge the convex hull to ensure enclosure of all trajectories for the time interval $t \in [t_0, t_1]$, including the effects of uncertain inputs.

The computation of further time intervals is performed as in [15], which is similar to the computation of the first time interval, but no further convex hull computations are required and further computations are carried out without the wrapping effect, i.e. without accumulating overapproximation errors.

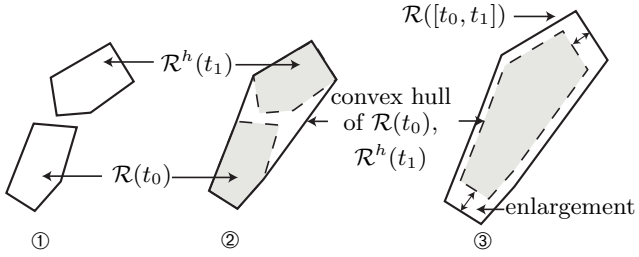


Figure 2: Steps for the computation of an overapproximation of the reachable set.

For hybrid systems, it is necessary to also consider discrete transitions activated by guards. Once these sets are determined, the mapping due to possible jumps is performed and the continuous reachable set computation can be continued. Two methods for guard intersection are presented: geometric guard intersection and the new technique mapping reachable sets onto guard sets.

3.2 Geometric Guard Intersection

The classical approach to computing reachable sets across discrete transitions in hybrid systems considers all possible states hitting a guard by computing the intersection of the reachable set with the guard set, see e.g. [7, 8, 11, 14, 18]. This is done by first intersecting all reachable sets of individual time intervals $[t_{k-1}, t_k]$ with the guard set in an exact or overapproximative way. In a second step, the individual intersections are unified into one or a few sets in order to bound the number of initial sets for continuing the reachability computations in the newly reached location, see [11]. This procedure is illustrated in Fig. 3(a) for polyhedral sets, where $\mathcal{R}_g$ is the intersection with the guard set and the displayed vertices indicate individual intersections.

When using representations other than general polyhedra, such as ellipsoids [7], multidimensional intervals [18], zonotopes [14], or template polyhedra [11], the intersection with polyhedral guard sets (which is the most common type) might result in large overapproximations. This problem is avoided by general polyhedra [8], but there are two problems with polyhedral computations: (i) the result is not numerically stable unless infinite precision arithmetic is used [6], and (ii) the unification of individual intersections by a convex hull is computationally expensive [5].

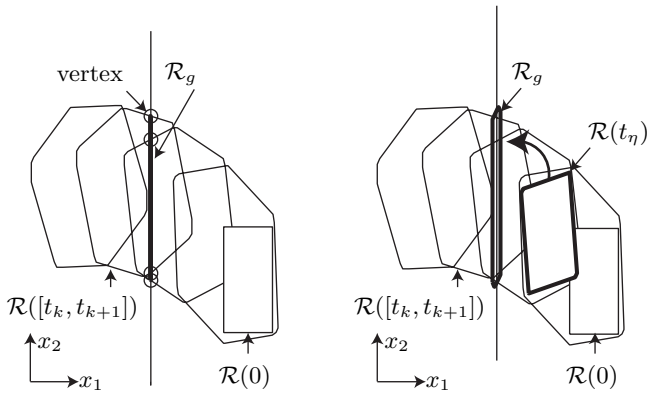


Figure 3: Guard intersection using the classical and the new approach.

3.3 Mapping onto Guard Sets

In the following, we propose a new technique to avoid geometric operations and directly map reachable sets to sets enclosing guard intersections. We compute this mapping from the last reachable set at a point in time t_η that does not intersect the guard. This point in time is determined by the first reachable set of the time interval $[t_\eta, t_{\eta+1}]$ which intersects the guard set. It is obvious that although for $[t_\eta, t_{\eta+1}]$ there is an intersection, there is no intersection for t_η since there was no intersection for $[t_{\eta-1}, t_\eta]$. In Fig. 3(b), the mapping from $\mathcal{R}(t_\eta)$ (bold border) to the overapproximative guard intersection $\mathcal{R}_g$ is indicated by a curved arrow.

We motivate our new technique by a simple example, where the flow is constant $\dot{z} = b$; $z(0) = z^0$; $z, b, z^0 \in \mathbb{R}^n$, such that the solution is $z(t) = z^0 + b\,t$. We use z to denote the state of the constant-flow system to distinguish it from the continuous state x for general continuous dynamics (such as linear systems). Given a halfspace $\mathcal{H} = \{z | n^T z \leq d\}$, the time t_h for hitting the halfspace for the constant flow is computed by solving $n^T(z^0 + b\,t_h) = d$, which implies $t_h = -\frac{n^T}{n^T b} z^0 + \frac{d}{n^T b}$. Inserting t_h into the solution $z(t)$ results in a map from z^0 to the state z_h on the bordering hyperplane:

$$z_h = \underbrace{(I - \frac{bn^T}{n^T b})}_{=:C} z^0 + \underbrace{\frac{bd}{n^T b}}_{=:w},$$

where I is the identity matrix and $z_h = Cz^0 + w$ is an affine map. For any initial state $z^0 \in \mathcal{R}(0)$, this mapping can be easily performed for most set representations (ellipsoids, zonotopes, polytopes, support functions, etc.).

A possible extension from constant flow to linear dynamics ($\dot{x} = Ax(t) + u(t)$, $x(0) = x^0$, $A \in \mathbb{R}^{n \times n}$, $x, u \in \mathbb{R}^n$) is to first assume constant flow, e.g. $\dot{z} = b = Ax^0 + u(0)$. We can then add a set that bounds the error between the constant flow and the trajectories of the linear dynamic system. This error $\mathcal{E} = \{e(t) = x(t) - z(t) | t \in [0, t_{h,\max}]\}$ is added via Minkowski addition ($\mathcal{A} \oplus \mathcal{B} := \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$) to $z(t)$, which assures $x(t) \in z(t) \oplus \mathcal{E}$ for $t \in [0, t_{h,\max}]$, where $t_{h,\max}$ is the time until all initial states have hit the guard set. Thus, $t_h \in -\frac{n^T}{n^T b}(x^0 \oplus \mathcal{E}) \oplus \frac{d}{n^T b}$, which after insertion

into $x(t) \in z(t) \oplus \mathcal{E}$ yields

$$x_h \in \left(\left(I - \frac{bn^T}{n^T b}\right)x^0 + \frac{bd}{n^T b}\right) \oplus \left(-\frac{bn^T}{n^T b} \otimes \mathcal{E}\right) \oplus \mathcal{E}, \quad (1)$$

where x_h denotes a state of the linear dynamic system when a trajectory has hit the guard set and $\mathcal{A} \otimes \mathcal{B} := \{ab | a \in \mathcal{A}, b \in \mathcal{B}\}$ is referred to as set-based multiplication. In the following, the operator $\otimes$ is sometimes omitted for simplicity.

We have used the constant flow to motivate the approach described in the next section using a more general class of dynamics we call *state-dependent constant flow*, which gives a much tighter approximation than (1) to the guard set intersection for linear dynamics. If the dynamics is nonlinear, we propose to first abstract to linear dynamics as shown in [4], where also a set of errors is computed in order to preserve the overapproximation of the result. Since an abstraction to linear dynamics already exists, we focus on the abstraction of linear dynamics to state-dependent constants flows.

4. STATE-DEPENDENT CONSTANT FLOW

We define state-dependent constant flow using the system matrix $A \in \mathbb{R}^{n \times n}$, the initial state $y^0 = y(0) \in \mathbb{R}^n$, and a given vector b as

$$\dot{y} = Ay^0 + b. \quad (2)$$

For a given y^0, this is simply a constant-flow system. The difference is that by including the initial state explicitly, we will be able to compute the mapping as a function of the initial state. The solution of (2) is

$$y(t) = y^0 + (Ay^0 + b)t. \quad (3)$$

A nice property of state-dependent constant flow is that the hitting time of a hyperplane has a closed-form solution as derived in the previous section.

We compute the error made when abstracting a linear system to state-dependent constant flow, where the linear system has constant input u_c:

$$\dot{x}(t) = Ax(t) + u_c. \quad (4)$$

The solution of the linear system for $x(0) = x^0$ is

$$x(t) = e^{At}x^0 + \underbrace{\int_0^t e^{A(t-\tau)} \mathrm{d}\tau \, u_c}_{=:\Gamma(t)}. \quad (5)$$

When A is invertible, the matrix $\Gamma(t)$ can be computed as $\Gamma(t) = A^{-1}(e^{At} - I)$, where I is the identity matrix. However, $\Gamma(t)$ is not always invertible, so we compute $\Gamma(t)$ by integrating the Taylor series of e^{At} with remainder term

$$e^{At} = \sum_{i=0}^{\infty} \frac{(At)^i}{i!} \in \sum_{i=0}^{\eta} \frac{(At)^i}{i!} \oplus \hat{\mathcal{E}}(t). \quad (6)$$

The error term for the finite Taylor series is modeled as an interval matrix $\hat{\mathcal{E}}(t) = [-W(t), W(t)]$, where

$$W(t) = \left| \sum_{i=\eta+1}^{\infty} \frac{A^i}{i!} t^i \right| \leq \sum_{i=\eta+1}^{\infty} \frac{|A|^i t^i}{i!} = e^{|A|t} - \sum_{i=0}^{\eta} \frac{|A|^i t^i}{i!}.$$

Integration yields

$$\Gamma(t) = \sum_{i=0}^{\infty} \frac{A^i t^{i+1}}{(i+1)!} \in \sum_{i=0}^{\eta} \frac{A^i t^{i+1}}{(i+1)!} \oplus \tilde{\mathcal{E}}(t), \quad (7)$$

where one can choose $\tilde{\mathcal{E}}(t) = \hat{\mathcal{E}}(t)t$, see [3]. The approximation error made for computing the hitting state based on state-dependent constant flow can be made arbitrarily small for a single initial state:

PROPOSITION 1 (MINIMIZING THE STATE DIFFERENCE). *The error $e(\tilde{t}_h) = x(\tilde{t}_h) - y(\tilde{t}_h)$, where $\tilde{t}_h$ is the hitting time of the linear dynamics and $x(0) = y(0) = x^0$, is minimized by choosing*

$$b = \Theta(\tilde{t}_h)x^0 + \Gamma^*(\tilde{t}_h)u_c$$
$$\Theta(t) = \sum_{i=2}^{\eta} \frac{A^i t^{i-1}}{i!}, \quad \Gamma^*(t) = \sum_{i=0}^{\eta} \frac{A^i t^i}{(i+1)!}$$

such that $\lim_{\eta \to \infty} e(\tilde{t}_h) = \mathbf{0}$ and $\mathbf{0}$ is a vector of zeros of proper dimension.

PROOF. We compute the error under the assumption that $x(0) = y(0) = x^0$, and $\xi \neq x^0$ is used for obtaining b according to the proposition. Using (5) and (3) we have for the proposed choice of b

$$\begin{aligned} e(t) &= x(t) - y(t) \\ &= \left(\sum_{i=0}^{\eta} \frac{A^i t^i}{i!} \oplus \hat{\mathcal{E}}(t)\right)x^0 \oplus \left(\sum_{i=0}^{\eta} \frac{A^i t^{i+1}}{(i+1)!} \oplus \tilde{\mathcal{E}}(t)\right)u_c \\ &\quad - \left(x^0 + Ax^0 t + \left(\sum_{i=2}^{\eta} \frac{A^i \tilde{t}_h^{i-1}}{i!}\xi + \sum_{i=0}^{\eta} \frac{A^i \tilde{t}_h^i}{(i+1)!}u_c\right)t\right) \quad (8) \\ &= t\left(\sum_{i=2}^{\eta} \frac{A^i}{i!}(t^{i-1}x^0 - \tilde{t}_h^{i-1}\xi) \oplus \sum_{i=0}^{\eta} \frac{A^i(t^i - \tilde{t}_h^i)}{(i+1)!}u_c\right) \\ &\quad \oplus \hat{\mathcal{E}}(t)x^0 \oplus \hat{\mathcal{E}}(t)\, t\, u_c. \end{aligned}$$

Since $x^0 + Ax^0 t$ cancels out, the error of the initial state x^0 is only $\mathcal{O}(t^2)$, whereas it would be $\mathcal{O}(t)$ when one chooses constant flow. For $x^0 = \xi$ and $t = \tilde{t}_h$, the error reduces to $e(\tilde{t}_h) = \hat{\mathcal{E}}(\tilde{t}_h)x^0 \oplus \hat{\mathcal{E}}(\tilde{t}_h)\tilde{t}_h u_c$, so that $\lim_{\eta \to \infty} e(\tilde{t}_h) = \mathbf{0}$. □

When $x(0), y(0) \in \mathcal{R}(0)$, the heuristics of computing b as presented above with x^0 as the center of $\mathcal{R}(0)$ has proven successful in our numerical experiments. Thus far, we have approximated the intersection of reachable states with the guard set for a given initial state. The following section extends this approximation to a set of initial states.

5. SET-BASED COMPUTATIONS

In this section, we compute (i) the set of states y_h and (ii) the error $e(t)$ when the initial state is within a set of states $x(0) = y(0) \in \mathcal{R}(0)$. Thereto, we first introduce matrix zonotopes, which are used for intermediate results, and vector zonotopes, which are used for the representation of the reachable set.

DEFINITION 3 (MATRIX ZONOTOPE). *Given a matrix center $C \in \mathbb{R}^{n \times n}$ and matrix generators $G^{(i)} \in \mathbb{R}^{n \times n}$, a matrix zonotope is defined as*

$$\mathcal{Z} = \left\{C + \sum_{i=1}^{p} \beta_i G^{(i)} \middle| \beta_i \in [-1, 1];\, C, G^{(i)} \in \mathbb{R}^{n \times n}\right\} \quad (9)$$

We write in short $\mathcal{Z} = (C, G^{(1)}, \ldots, G^{(p)})$ and define the *order* as $\rho = \frac{p}{n}$, where p is the number of generators. By replacing the matrix center with a vector center $c \in \mathbb{R}^n$ and

the matrix generators with vector generators $g^{(i)} \in \mathbb{R}^n$, one obtains the standard vector zonotope, which is simply called a *zonotope*. Zonotopes are a compact way of representing sets in high dimensions, which, as is shown in the following, scale well for the operations required in our procedure.

5.1 Operations on Zonotopes

We start with the addition of two zonotopes $\mathcal{Z}_1 = (c,\, g^{(1)},\, \ldots,\, g^{p_1})$ and $\mathcal{Z}_2 = (d,\, h^{(1)},\, \ldots,\, h^{(p_2)})$, and the multiplication with a matrix $L \in \mathbb{R}^{o\times n}$, where both operations are a direct consequence of the zonotope definition (see [19]):

$$\begin{aligned} \mathcal{Z}_1 \oplus \mathcal{Z}_2 &= (c+d, g^{(1)}, \ldots, g^{(p_1)}, h^{(1)}, \ldots, h^{(p_2)}) \\ L \otimes \mathcal{Z}_1 &= (L\,c, L\,g^{(1)}, \ldots, L\,g^{(p_1)}) \end{aligned} \tag{10}$$

Additionally, we require the scalar interval multiplication

$$[-1,1] \otimes \mathcal{Z}_1 \subseteq (\mathbf{0}, c, g^{(1)}, \ldots, g^{(p_1)}), \tag{11}$$

where $\mathbf{0}$ is a vector of zeros of proper dimension. We also require the quadratic map of a zonotope, which is newly derived.

LEMMA 1 (QUADRATIC MAP). *Given a zonotope $\mathcal{Z} = (c, g^{(1)}, \ldots, g^{(p)})$ and a discrete set of matrices $Q^{(i)} \in \mathbb{R}^{n\times n}$, $i = 1 \ldots n$, the set*

$$\mathcal{Z}_Q = \{\lambda | \lambda_i = x^T Q^{(i)} x,\, x \in \mathcal{Z}\}$$

is overapproximated by a zonotope $(d, h^{(1)}, \ldots, h^{(\sigma)})$, where $\sigma = \binom{p+2}{2} - 1$. The center is computed as

$$d_i = c^T Q^{(i)} c + 0.5 \sum_{s=1}^{p} {g^{(s)}}^T Q^{(i)} g^{(s)},$$

and the generators are computed as

$$\begin{aligned} j = 1 \ldots p: \quad & h_i^{(j)} = c^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} c \\ j = 1 \ldots p: \quad & h_i^{(p+j)} = 0.5 {g^{(j)}}^T Q^{(i)} g^{(j)} \\ l = \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} 1: \quad & h_i^{(2p+l)} = {g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)} \end{aligned}$$

The complexity of constructing this zonotope overapproximation with respect to the dimension n is $\mathcal{O}(n^5)$.

PROOF. Inserting the definition of a zonotope into the set $\mathcal{Z}_Q = \{\lambda | \lambda_i = x^T Q^{(i)} x,\, x \in \mathcal{Z}\}$ yields

$$\Big\{\lambda \Big| \lambda_i = (c + \sum_{j=1}^{p} \beta_j g^{(j)})^T Q^{(i)} (c + \sum_{j=1}^{p} \beta_j g^{(j)}),\, \beta_j \in [-1,1]\Big\},$$

which can be rearranged to

$$\begin{aligned} \mathcal{Z}_Q = \Big\{\lambda \Big| \lambda_i = & \underbrace{c^T Q^{(i)} c + \sum_{j=1}^{p} 0.5 {g^{(j)}}^T Q^{(i)} g^{(j)}}_{d_i} \\ & + \sum_{j=1}^{p} \beta_j \underbrace{(c^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} c)}_{h_i^{(j)}} \\ & + \sum_{j=1}^{p} (2\beta_j^2 - 1) \underbrace{0.5 {g^{(j)}}^T Q^{(i)} g^{(j)}}_{h_i^{(p+j)}} \\ & + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \beta_j \beta_k \underbrace{({g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)})}_{h_i^{(2p+l)}}, \\ & \beta_i \in [-1,1]\Big\} \\ \subseteq & \Big(d, h^{(1)}, \ldots, h^{(\sigma)}\Big). \end{aligned}$$

The obtained zonotope is an overapproximation since $\beta_j \in [-1,1]$, $(2\beta_j^2 - 1) \in [-1,1]$, and $\beta_j \beta_k \in [-1,1]$ for $j \neq k$. The number of new generators is obtained from the fact that the new generators $h^{(j)}$ are computed by picking two elements from the set containing all generators and the center, where replacement is allowed and order does not matter. By subtracting the possibility that one can choose two centers, one obtains $\sigma = \binom{p+2}{2} - 1$ generators.

It remains to derive the complexity. Quadratic operations such as ${g^{(j)}}^T Q^{(i)} g^{(k)}$ have complexity $\mathcal{O}(n^2)$. The number p of generators of $\mathcal{Z}$ can be expressed by its order as $\rho\, n$, such that the resulting zonotope has $\binom{(\rho\, n)+2}{2} - 1$ generators, a number which can be bounded by $\mathcal{O}(n^2)$, so we have $\mathcal{O}(n^4)$ for all generator computations for each dimension and $\mathcal{O}(n^5)$ for all dimensions. □

The quadratic map can be generalized to the case when all $Q^{(i)}$ are elements of a matrix zonotope:

THEOREM 1 (MATRIX ZONOTOPE MAP). *We have $\mathcal{Z} = (c, g^{(1)}, \ldots, g^{(p)})$ and the matrices $Q^{(i)} \in \mathcal{Q}^{(i)} = (D^{(i)},\, K^{(i,1)},\, \ldots,\, K^{(i,\nu)})$. The zonotope enclosing the set $\{\lambda | \lambda_i = x^T Q^{(i)} x,\, x \in \mathcal{Z},\, Q^{(i)} \in \mathcal{Q}^{(i)}\}$ can be overapproximated by*

$$\mathcal{Z}_D \oplus ([-1,1] \otimes \mathcal{Z}_{K^{(1)}}) \oplus \ldots \oplus ([-1,1] \otimes \mathcal{Z}_{K^{(\nu)}}),$$

where the addition and interval multiplication is performed as in (10) and (11), and the partial solutions are computed as in Lemma 1, where

$$\begin{aligned} \mathcal{Z}_D &\supseteq \{\lambda | \lambda_i = x^T D^{(i)} x,\, x \in \mathcal{Z}\} \\ \mathcal{Z}_{K^{(j)}} &\supseteq \{\lambda | \lambda_i = x^T K^{(i,j)} x,\, x \in \mathcal{Z}\} \end{aligned}$$

The complexity with respect to the dimension n is $\mathcal{O}(n^6)$.

PROOF. After inserting the definition of the matrix zonotope into $\lambda_i = x^T Q^{(i)} x$, one obtains

$$\begin{aligned} \lambda_i &= x^T (D^{(i)} + \sum_{j=1}^{\nu} \beta_j K^{(i,j)}) x \\ &= \underbrace{x^T D^{(i)} x}_{\in I_i \otimes \mathcal{Z}_D} + \beta_1 \underbrace{x^T K^{(i,1)} x}_{\in I_i \otimes \mathcal{Z}_{K^{(1)}}} + \ldots + \beta_\nu \underbrace{x^T K^{(i,\nu)} x}_{\in I_i \otimes \mathcal{Z}_{K^{(\nu)}}}, \end{aligned} \tag{12}$$

where I_i is the i^{th} row of the identity matrix. Since each β_j is within $[-1,1]$, we obtain $\mathcal{Z}_D \oplus ([-1,1] \otimes \mathcal{Z}_{K^{(1)}}) \oplus \ldots \oplus ([-1,1] \otimes \mathcal{Z}_{K^{(\nu)}})$.

Given that the complexity for each partial zonotope $\mathcal{Z}_{K^{(j)}}$ is $\mathcal{O}(n^5)$ from the previous lemma, and that $\tilde{\rho}$ is the order of the matrix zonotopes $\mathcal{Q}^{(i)}$, we have $\tilde{\rho}\, n + 1$ partial zonotopes, giving an overall complexity of $\mathcal{O}(n^6)$. □

5.2 Hitting Times

Before we compute the mapping onto guard sets, we have to find a bound on the times when the guard set is hit. Determining when a zonotope $\mathcal{Z}$ intersects a halfspace $\mathcal{H} = \{x | n^T x \leq d\}$ is computationally efficient since it only evolves

checking if $(n^T \otimes \mathcal{Z}) \oplus (-d) \leq 0$. By interpreting d as a zonotope with no generators, this expression can be evaluated by (10), resulting in a zonotope of dimension 1, which we express as $\omega_c \oplus ([-1,1] \otimes \omega^{(1)}) \oplus \ldots \oplus ([-1,1] \otimes \omega^{(p)})$; $\omega_c, \omega^{(i)} \in \mathbb{R}$. The interval $[\underline{\omega}, \overline{\omega}]$ of zonotope values is obtained by

$$\underline{\omega} = \omega_c - \sum_{i=1}^{p} |\omega^{(i)}|, \quad \overline{\omega} = \omega_c + \sum_{i=1}^{p} |\omega^{(i)}|.$$

If $\underline{\omega} \geq 0$, no intersection occurs and if $\overline{\omega} \leq 0$, the reachable set is completely in the guard set. The union of all consecutive time intervals $[t_k, t_{k+1}]$, for which $\overline{\omega} \geq 0$ and $\underline{\omega} \leq 0$, forms the interval of hitting times. This interval is used below to bound the error of the state-dependent constant flow assumption. Note that in the event that no guard set is hit, only the continuous evolution has to be considered.

5.3 Set of Abstraction Errors

For a given initial state, the error between the state-dependent constant flow solution $y(t)$ and the linear system solution $x(t)$ has already been presented in (8). Now, we consider a set of initial states $\mathcal{R}(0) = \xi \oplus \mathcal{Y}$, where $\mathcal{Y}$ is the deviation from ξ, and for simplicity we reset the time such that the interval of hitting times is $\mathcal{T} = [0, t_{\max}]$. The set of errors based on (8) is

$$\begin{aligned}\mathcal{E} =& \{e(t) = x(t) - y(t) | t \in \mathcal{T}; x^0, y^0 \in (\xi \oplus \mathcal{Y})\} \\ =& \mathcal{T}\Big(\underbrace{\bigoplus_{i=2}^{\eta} \frac{A^i}{i!}\Big(\mathcal{T}^{i-1}(\xi \oplus \mathcal{Y}) \oplus (-\tilde{t}_h^{i-1}\xi)\Big)}_{=\sum_{i=2}^{\eta} \frac{A^i}{i!}\left(\mathcal{T}^{i-1}\mathcal{Y} \oplus (\mathcal{T}^{i-1} \oplus (-\tilde{t}_h^{i-1}))\xi\right)} \\ & \oplus \bigoplus_{i=0}^{\eta} \frac{A^i(\mathcal{T}^i \oplus (-\tilde{t}_h^i))}{(i+1)!} u_c \Big) \oplus \hat{\mathcal{E}}(t_{\max}) x^0 \oplus \hat{\mathcal{E}}(t_{\max}) t_{\max} u_c, \end{aligned} \tag{13}$$

which is computed with complexity $\mathcal{O}(n^3)$. Note that due to the monotonic growth of $\hat{\mathcal{E}}(t)$, it is sufficient to use the latest time $t_{\max}$ for $\hat{\mathcal{E}}(t)$. Since all values of $\mathcal{T}$ are positive, we have $\mathcal{T}^i = [0, t_{\max}^i]$ such that we can bound $\mathcal{T}^i$ by a zonotope (T, T), where $T = 0.5 t_{\max}^i$. The multiplication $\mathcal{T}^i \otimes \mathcal{Y}$ in (13) is performed similar to Theorem 1 as $\mathcal{T}^i \otimes \mathcal{Y} \supseteq T\mathcal{Y} \oplus ([-1,1] \otimes T\mathcal{Y})$. Other operations in (13) are performed similarly, by e.g. considering ξ as a zonotope with no generators.

5.4 Set of State-Dependent Constant Flow Solutions

Using the previously introduced operations on zonotopes and matrix zonotopes, we compute the set of states x_h hitting the halfspace $\mathcal{H} = \{x | n^T x \leq d\}$ by bounding $x(t)$ with constant individual flow and the error set $\mathcal{E}$:

$$x(t) \in x^0 + (Ax^0 + b)t \oplus \mathcal{E} \text{ for } t \in [0, t_{\max}], \tag{14}$$

By replacing b with $(Ax^0 + b)$ in (1), one obtains the set of states $\mathcal{R}_h(x^0)$ enclosing x_h:

$$\mathcal{R}_h(x^0) := x^0 + (Ax^0 + b)\frac{d - n^T(x^0 \oplus \mathcal{E})}{n^T(Ax^0 + b)} \oplus \mathcal{E}, \tag{15}$$

which is split into the hitting state $y_h(x^0)$ of $y(t)$ and $\mathcal{R}_{h,\mathcal{E}}(x^0)$, such that $\mathcal{R}_h(x^0) = y_h(x^0) \oplus \mathcal{R}_{h,\mathcal{E}}(x^0)$ and

$$y_h(x^0) = x^0 + (Ax^0 + b)\frac{d - n^T x^0}{n^T(A\,x^0 + b)},$$

$$\mathcal{R}_{h,\mathcal{E}}(x^0) = (Ax^0 + b)\frac{-n^T\mathcal{E}}{n^T(A\,x^0 + b)} \oplus \mathcal{E}.$$

We will first focus on $y_h(x^0)$. The set of states $y_h(x^0)$ for $x^0 \in \mathcal{R}(0)$ is

$$\Big\{ x^0 + (Ax^0 + b)\frac{d - n^T x^0}{n^T(A\,x^0 + b)} \Big| x^0 \in \mathcal{R}(0) \Big\},$$

which requires to divide by $n^T(A\,x^0 + b)$ with $x^0 \in \mathcal{R}(0)$. This could be resolved by first computing the interval

$$\mathcal{I} = \{n^T(Ax^0 + b) | x^0 \in \mathcal{R}(0)\}, \tag{16}$$

and then computing the set of y_h as $\{x^0 + (Ax^0 + b)(d - n^T x^0)/a | x^0 \in \mathcal{R}(0), a \in \mathcal{I}\}$. However, this approach neglects the dependency of x^0 and thus is too conservative. We capture the dependency in a much better way by applying a Taylor series to the expression for y_h, for which we first need the partial derivatives.

PROPOSITION 2 (PARTIAL DERIVATIVES OF y_h). *We use*

$$\Lambda(y) := n^T(Ay + b), \qquad \Upsilon := n^T A,$$
$$\Theta(y) := -n\Lambda(y) - (d - n^T y)\Upsilon^T, \quad \Omega := -n\Upsilon + \Upsilon^T n^T.$$

for a concise notation of the partial derivatives with respect to x^0 in index notation:

$$\begin{aligned} L_{il}(x^0) :=& \frac{\partial y_{h,i}}{\partial x_l^0} = I_{il} + \frac{A_{il}}{\Lambda(x^0)}(d - \sum_{j=1}^{n} n_j x_j^0) \\ & + (\sum_{j=1}^{n} A_{ij} x_j^0 + b_i)\frac{\Theta_l(x^0)}{\Lambda(x^0)^2}, \\ \tilde{Q}_{lm}^{(i)}(x^0) :=& \frac{\partial^2 y_{h,i}}{\partial x_l^0 \partial x_m^0} = \frac{1}{\Lambda(x^0)^2} Q_{lm}^{(i)}(x^0), \text{ where} \\ Q_{lm}^{(i)}(x^0) =& A_{il}\Theta_m(x^0) + A_{im}\Theta_l(x^0) \\ & + (\Omega_{lm} - \frac{\Theta_l(x^0)}{\Lambda(x^0)} 2\Upsilon_m)(\sum_{j=1}^{n} A_{ij} x_j^0 + b_i). \end{aligned}$$

The derivatives are obtained via standard derivation.

PROPOSITION 3 (TAYLOR SERIES OF y_h). *The first-order Taylor series with Lagrange remainder of the state y_h is*

$$y_{h,i}(x^0) \in \Big(y_{h,i}(\xi) + L_i(\xi)\nu\Big) \oplus \Big(\frac{1}{2}\nu^T \tilde{\mathcal{Q}}^{(i)} \nu\Big), \tag{17}$$

where $\nu = x^0 - \xi$, L_i is the i^{th} row of the matrix L, $\tilde{\mathcal{Q}}^{(i)} = [\underline{\varphi}, \overline{\varphi}] \otimes \mathcal{Q}^{(i)}$, $[\underline{\varphi}, \overline{\varphi}]$ is the interval of $\{1/\Lambda(x^0)^2 | x^0 \in \mathcal{R}(0)\}$,

$$\mathcal{Q}^{(i)} = \{Q^{(i)}(x^0) | x^0 \in \mathcal{R}(0)\} \subseteq (\tilde{C}^{(i)}, \tilde{G}^{(i,1)}, \ldots, \tilde{G}^{(i,1+2p)}),$$

where

$$\begin{aligned} \tilde{C}_{lm}^{(i)} =& A_{il}\Theta_{c,m} + A_{im}\Theta_{c,l} \\ & + (\Omega_{lm} - \psi_{c,l} 2\Upsilon_m)(\textstyle\sum_{j=1}^{n} A_{ij} c_{x,j} + b_i), \\ \tilde{G}_{lm}^{(i,1)} =& -\psi_{g,l} 2\Upsilon_m (\textstyle\sum_{j=1}^{n} A_{ij} c_{x,j} + b_i), \\ \tilde{G}_{lm}^{(i,1+\alpha)} =& A_{il}\Theta_{g,m}^{(\alpha)} + A_{im}\Theta_{g,l}^{(\alpha)} \\ & + (\Omega_{lm} - \psi_{c,l} 2\Upsilon_m) \textstyle\sum_{j=1}^{n} A_{ij} g_{x,j}^{(\alpha)}, \\ \tilde{G}_{lm}^{(i,1+p+\alpha)} =& -\psi_{g,l} 2\Upsilon_m \textstyle\sum_{j=1}^{n} A_{ij} g_{x,j}^{(\alpha)}. \end{aligned}$$

PROOF. The Lagrange remainder in (17) encloses all higher order terms if $\{\tilde{Q}^{(i)}(x^0)|x^0 \in \mathcal{R}(0)\} \subseteq \tilde{\mathcal{Q}}^{(i)}$. Based on the computation of $\tilde{Q}^{(i)}(x^0)$ in Proposition 2, we first compute the interval of $\{1/\Lambda(x^0)^2|x^0 \in \mathcal{R}(0)\}$ and then $\{Q_{lm}^{(i)}(x^0)|x^0 \in \mathcal{R}(0)\}$, which we enclose by the proposed matrix zonotope. Since the initial set is a zonotope, we have that $x^0 = c_x + \sum_{\alpha=1}^{p} \beta_\alpha g_x^{(\alpha)}$, and $\Theta_l(x^0) = \Theta_{c,l} + \sum_{\alpha=1}^{p} \beta_\alpha \Theta_{g,l}^{(\alpha)}$, where

$$\begin{aligned} \Theta_{c,l} &= -n(n^T(Ac_x + b)) - (d - n^T c_x)\Upsilon^T, \\ \Theta_{g,l}^{(\alpha)} &= -n(n^T A g_x^{(\alpha)}) + n^T g_x^{(\alpha)} \Upsilon^T. \end{aligned} \tag{18}$$

Next, we obtain the overapproximation

$$\{\Theta_l(x^0)/\Lambda(x^0)|x^0 \in \mathcal{R}(0)\} \subseteq \psi_{c,l} \oplus [-1,1]\psi_{g,l} \tag{19}$$

using interval arithmetic. The center and the generators of the matrix zonotope $(\tilde{C}^{(i)}, \tilde{G}^{(i,1)}, \ldots, \tilde{G}^{(i,1+2p)})$ result from inserting (18), (19), and x^0 in zonotope form into $Q^{(i)}(x^0)$ in Proposition 2. □

It remains to compute $\mathcal{R}_{h,\mathcal{E}}(\mathcal{R}(0))$, which is much smaller compared to the set of y_h. Thus, it is sufficient to use (16) and compute $\mathcal{R}_{h,\mathcal{E}}(\mathcal{R}(0)) = (A\mathcal{R}(0) \oplus b)(-n^T\mathcal{E})(1/\mathcal{I}) \oplus \mathcal{E}$.

Using Proposition 3, the zonotope $\mathcal{R}_g$ enclosing the guard intersection is computed as

$$\mathcal{R}_g = y_h(\xi) \oplus L \otimes (\mathcal{R}(0) \oplus (-\xi)) \oplus \frac{1}{2}[\underline{\varphi}, \overline{\varphi}] \otimes \mathcal{R}_{\text{quad}} \oplus \mathcal{R}_{h,\mathcal{E}}(\mathcal{R}(0)),$$

where $[\underline{\varphi}, \overline{\varphi}] \otimes \mathcal{R}_{\text{quad}} \subseteq 0.5(\underline{\varphi} + \overline{\varphi})\mathcal{R}_{\text{quad}} \oplus 0.5(\overline{\varphi} - \underline{\varphi}) \otimes [-1,1] \otimes \mathcal{R}_{\text{quad}}$ is computed using (10), (11), and $\mathcal{R}_{\text{quad}} \supseteq \{\lambda|\lambda_i = x^T Q^{(i)} x, x \in (\mathcal{R}(0) \oplus (-\xi)), Q^{(i)} \in \mathcal{Q}^{(i)}\}$ is computed as in Theorem 1. The overall complexity of computing $\mathcal{R}_g$ is determined by the computation of $\mathcal{R}_{\text{quad}}$, such that it is $\mathcal{O}(n^6)$. When the constant flow approximation is sufficient, the complexity reduces to $\mathcal{O}(n^3)$.

Note that in case of uncertain inputs $u(t) \in \mathcal{U}$, one needs an additional Minkowski addition to $\mathcal{R}_g$ due to uncertain inputs as described in [3].

5.5 Hitting Several Guard Sets

A problem, which is not addressed in [16], is how to deal with reachable sets that hit several guard sets. A straightforward way would be to separately compute the intersection for each guard using the presented approach, as illustrated by $\tilde{\mathcal{R}}_{g,1}, \tilde{\mathcal{R}}_{g,2}$ in Fig. 4. Without further intersection of other overlapping guard sets, one could continue the computation with $\tilde{\mathcal{R}}_{g,1}, \tilde{\mathcal{R}}_{g,2}$ as new initial sets. However, this often leads to a substantial overapproximation. In this case we suggest to intersect the mapped guard set with all other guards that have been hit, resulting in $\mathcal{R}_{g,1} = \tilde{\mathcal{R}}_{g,1} \cap \mathcal{H}_2$, $\mathcal{R}_{g,2} = \tilde{\mathcal{R}}_{g,2} \cap \mathcal{H}_1$ for the example in Fig. 4.

As mentioned earlier, zonotopes are not closed under intersection, but intersection can be tightly overapproximated using zonotopes or zonotope bundles [2]. Although hitting several guard sets requires a classical intersection operation, one has the huge advantage that the guard intersection for all times is represented by a single set $\mathcal{R}_g$, while for the classical approach, one first has to enclose partial intersections by a single set, which can be a costly or even unstable operation.

6. NUMERICAL EXAMPLE

We present the usefulness and scalability of the presented approach on instances of a mechanical system with backlash.

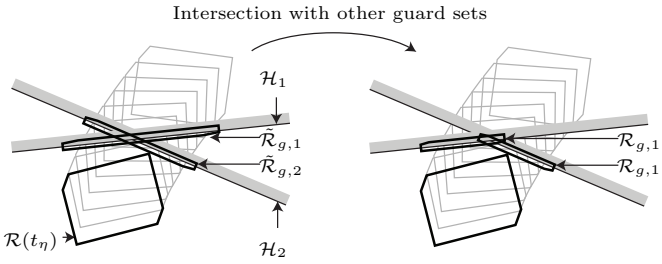


Figure 4: Procedure for hitting several halfspaces.

Methods for analyzing such systems is of great interest since backlash drastically reduces control performance [25]. The considered system in Fig. 5 is taken from an automotive drivetrain problem [22] and enhanced by additional rotating masses. Similar versions of this problem can be found in robotics, automation, and production machines. We first derive the differential equations of the system and then present the results in comparison with other approaches.

6.1 System Equations

The indices m and l refer to the motor and the load, numbered indices refer to the numbering of additional rotating masses, which are sometimes generalized by i. Moments of inertia are denoted by J [kg m^2], viscous friction constants by b [Nm s/rad], shaft stiffness by k [Nm/rad], angular positions by Θ [rad], and torque by T [Nm] (see Fig. 5).

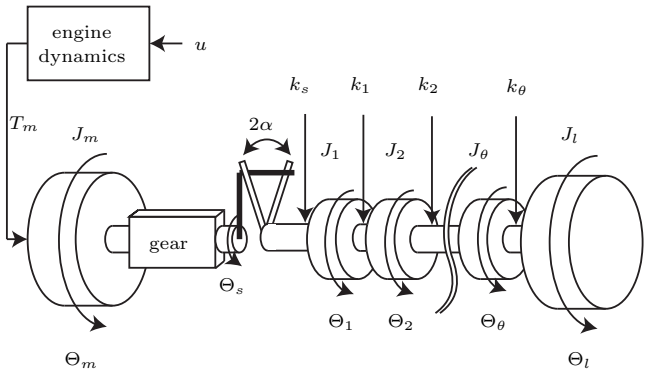


Figure 5: Powertrain model.

The differential equations of the rotating masses are

$$\begin{aligned} J_m\ddot{\Theta}_m + b_m\dot{\Theta}_m &= T_m - T_g \\ J_1\ddot{\Theta}_1 + b_1\dot{\Theta}_1 &= T_s - T_1 \\ &\cdots \\ J_n\ddot{\Theta}_\theta + b_\theta\dot{\Theta}_\theta &= T_{\theta-1} - T_\theta \\ J_l\ddot{\Theta}_l + b_l\dot{\Theta}_l &= T_\theta \end{aligned}$$

The dead-zone nonlinearity of the backlash is

$$T_s = k_s \begin{cases} \Theta_s - \Theta_1 - \alpha \text{ if } \Theta_s - \Theta_1 \geq \alpha \\ 0 \text{ if } |\Theta_s - \Theta_1| < \alpha \\ \Theta_s - \Theta_1 + \alpha \text{ if } \Theta_s - \Theta_1 \leq -\alpha \end{cases}$$

and the engine dynamics is modeled as $\dot{T}_m = (v - T_m)/\tau_{eng}$, where v [Nm] is the requested engine torque and τ_{eng} is the time constant. The remaining torques T_i with indices $i =$

$1 \dots (\theta-1)$ and index θ are computed as $T_i = k_i(\Theta_i - \Theta_{i+1})$ using the shaft stiffness, and the last torque is $T_\theta = k_\theta(\Theta_\theta - \Theta_l)$. The gearbox ratio $\gamma = 12$ constrains the torques and angles to $T_g = T_s/\gamma$, $\Theta_s = \Theta_m/\gamma$ and a PID controller is used to control the motor velocity, such that

$$v = k_P(\dot{\Theta}_{ref} - \dot{\Theta}_m) + k_I(\Theta_{ref} - \Theta_m) + k_D(\ddot{\Theta}_{ref} - \ddot{\Theta}_m).$$

In order to write the equations in state space form, we introduce the state variables $x_1 = \Theta_s - \Theta_1 = \Theta_m/i - \Theta_1$, $x_2 = T_m$, $x_3 = \Theta_{ref}$, $x_4 = \dot{\Theta}_{ref}$, $x_5 = \Theta_l$, $x_6 = \dot{\Theta}_l$, $x_7 = \dot{\Theta}_m$, $x_8 = \Theta_1$, $x_9 = \dot{\Theta}_1$, …, $x_{2\theta+6} = \Theta_\theta$, $x_{2\theta+7} = \dot{\Theta}_\theta$. We also define the input $u = \ddot{\Theta}_{ref}$, which is the acceleration of the reference angle. For $x_1 \geq \alpha$, which we refer to as location 1, the system dynamics in state space form results from the previous equations as

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{\gamma}x_7 - x_9 \\
\dot{x}_2 &= \frac{1}{\tau_{eng}}\Big((k_P(\gamma x_4 - x_7) + k_I(\gamma x_3 - \gamma(x_1 + x_8)) \\
&\quad + k_D\big(\gamma u - \frac{1}{J_m}(x_2 - \frac{1}{\gamma}k_s(x_1 - \alpha) - b_m x_7)\big) - x_2\Big) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= u \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \frac{1}{J_l}(k_\theta(x_{2\theta+6} - x_5) - b_l x_6) \\
\dot{x}_7 &= \frac{1}{J_m}(x_2 - \frac{1}{\gamma}k_s(x_1 - \alpha) - b_m x_7) \\
\dot{x}_8 &= x_9 \\
\dot{x}_9 &= \frac{1}{J_1}(k_s(x_1 - \alpha) - k_1(x_8 - x_{10}) - b_1 x_9) \\
&\dots \\
\dot{x}_{2\theta+6} &= x_{2\theta+7} \\
\dot{x}_{2\theta+7} &= \frac{1}{J_n}(k_{\theta-1}(x_{2\theta+4} - x_{2\theta+6}) - k_\theta(x_{2\theta+6} - x_5) - b_\theta x_{2\theta+7}).
\end{aligned}$$

The above dynamics can be written in linear form as $\dot{x} = A_1 x(t) + b_1 u(t) + c_1$, where the index refers to the location and $A_1 \in \mathbb{R}^{n\times n}$; $b_1, c_1 \in \mathbb{R}^n$, where $n = 2\theta + 7$. By setting $k_s = 0$ in the system equations, one obtains the dynamics of location 2 when the system is in the dead-zone, and by changing the sign of α, one obtains the dynamics of location 3 when the system is on the other side of the contact zone than location 1.

The parameters of the system taken from [22] are listed in Table 1. Parameters from the additional masses, which can be interpreted as rotating elements in a gearbox and further drivetrain elements, are taken from [26]. The PID parameters tuned by the authors are $k_P = 0.5$, $k_I = 0.5$, $k_D = 0.5$.

Table 1: Powertrain parameters in SI units.

α	τ_{eng}	b_l	b_m	b_i	k_s	k_i	J_l	J_m	J_i
0.03	0.1	5.6	0	1	10^4	10^5	140	0.3	0.01

6.2 Reachable Set Computation

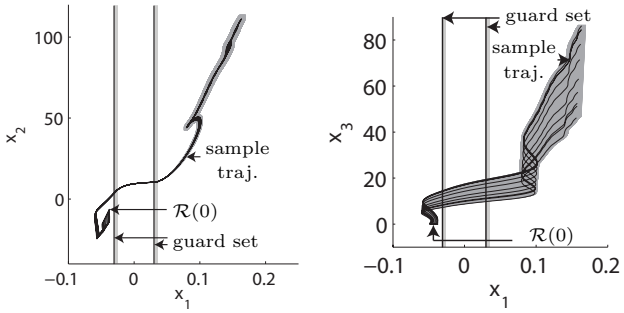
Many engineering questions of the considered powertrain can be solved by computing the reachable set, such as guaranteeing upper bounds on the torques for a set of initial states in order to ensure that no shaft will brake. Other problems could be the verification of a maximum settling time, or showing that the system does not reach the dead-zone for certain maneuvers.

In this work, we choose a benchmark maneuver from an assumed maximum negative acceleration of $\ddot{\Theta}_{ref} = -5$ [rad/s^2] to maximum positive acceleration of 5 [rad/s^2], where the first acceleration command lasts 0.2 s and the second one 1.8 s. We consider a wide range of possible initial angular velocities in the interval [20, 40] [rad/s], which would correspond to a range of [2292, 4584] RPM of the motor for the given gear ratio. Based on the range of initial angular velocities, the other initial states are chosen as the steady state solution for an external load of 300 [Nm], resulting in a zonotope with center $c = [-0.0432, -11, 0, 30, 0, 30, 360, -0.0013, 30, \dots, -0.0013, 30]^T$ and generator $g = [0.0056, 4.67, 0, 10, 0, 10, 120, 0.0006, 10, \dots, 0.0006, 10]^T$. The time step size is chosen as $r = 5 \cdot 10^{-4}$ s, and the maximum zonotope order for quadratic evaluations is limited to $\rho = 1.2$ using the reduction technique in [12].

We computed the reachable sets for different problem instances ($\theta = \{0, 1, \dots, 47\}$), resulting in up to 101 continuous state variables. The overall computational time, as well as the individual intersection times with guard sets (there are 2 intersections) are listed in Tab. 2. Computations were done in MATLAB on an i7 Processor and 6GB memory. Different projections of reachable sets for the instance with 101 continuous state variables are shown in Fig. 6, which also displays simulations from sampled initial states. Since the continuous dynamics can be computed wrapping-free [15], the overapproximation is small for all times, which can be seen by comparison with sample trajectories.

Table 2: Computational times in seconds ($n = 2\theta+7$).

dim. n	11	21	31	41	51	101
CPU time	8.122	14.31	23.72	31.83	53.74	1550
1^{st} guard	0.087	0.413	2.620	4.858	11.40	663.7
2^{nd} guard	0.094	0.467	2.704	4.774	11.71	522.4



(a) Projection onto x_1, x_2. (b) Projection onto x_1, x_3.

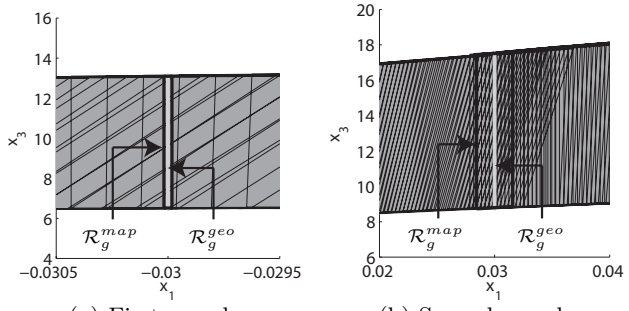
Figure 6: Reachable set of the powertrain for $n = 101$ ($\theta = 47$). Black lines show sampled trajectories and the gray region shows the reachable set.

For problem instances with $\theta = \{0, 1\}$ we compare the results with a classical geometrical approach [1, Chap. 3.5]

in which different enclosure techniques are discussed; in this paper, we use the one based on principal component analysis (see also [30]). Simpler techniques, such as a box enclosure result in unacceptably large overapproximations. The approximation in [30] is accurate, but computed based on vertices, which is not feasible for larger instances of the powertrain problem. The comparison of reachable sets at both guard sets for $\theta = 1$ is shown in Fig. 7. One can see that both approaches have small overapproximation error, with an even smaller error for the classical geometric approach, but the new approach scales much more favorably. For both guards and $\theta = 0$, the classical approach consumes 12.56 s (new approach: 0.133 s), and 286.7 s (new approach: 0.154 s) for $\theta = 1$, while the classical approach is infeasible for $\theta = 2$.

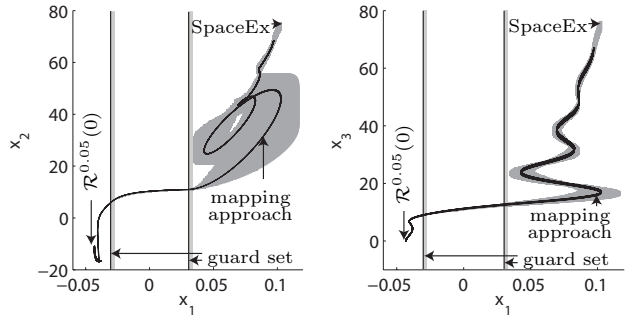
Additionally, we compared the results to the reachability tool *SpaceEx* [11]. There, the geometric intersection is computed using linear programming, by which one can solve higher-dimensional problems compared to vertex-based geometrical approaches, but one needs good normal vectors for bounding halfspaces, while their absence might result in substantial overapproximations. For the powertrain example the overapproximation is substantial so that we could only compute rather tight results for $\theta = 0$ and an initial set $\mathcal{R}^{0.05}(0)$, which is 5% of the original set in each direction. We used the following SpaceEx parameters: 0.01 flowpipe tolerance, 100% clustering, and the template directions are chosen octogonal plus 500 random directions. The results are shown together with the mapping approach in Fig. 8. Due to the large number of required directions, the computational time of SpaceEx is 10023 s compared to 0.133 s for the mapping approach. However, we emphasize that SpaceEx is a general-purpose tool, while our approach requires that all trajectories in a location actually hit the guard set.

One reason why the mapping approach outperforms the classical geometric approach is that the guard intersection of the presented example is bounded by a zonotope with 2020 generators for 101 continuous state variables, which is a compact representation of a polyhedral set bounded by $2\binom{2020}{100} = 3.04{\cdot}10^{171}$ halfspaces, a number that is out of reach for classical approaches. However, classical approaches are still required when guard sets are hit by only some of the trajectories.



(a) First guard. (b) Second guard.

Figure 7: Guard intersection using the geometrical (geo) and the mapping (map) approach for $n = 9$ ($\theta = 1$). The gray area shows the reachable set using the geometrical approach, black lines indicate the bounds of the reachable set of the mapping approach.



(a) Projection onto x_1, x_2. (b) Projection onto x_1, x_3.

Figure 8: Guard intersection for $n = 7$ ($\Theta = 0$) using SpaceEx and the mapping approach. The gray and black areas show the reachable set obtained from SpaceEx and the mapping approach.

7. CONCLUSIONS

We present a new approach for avoiding geometric intersection operations in reachability analysis of hybrid systems by overapproximating the intersection with a nonlinear map for guard sets modeled as halfspaces. The approach scales well with the system dimension, making it possible to verify systems which were previously out of reach. An important factor for the accuracy of the presented approach is the time interval of possible guard intersection times. When this time interval is too large, it might be necessary to split the reachable set used as the initial set for the mapping onto the guard, in order to refine the computation. As mentioned in the introduction, if the time interval is unbounded, a classical geometrical approach has to be applied.

Intermediate results of the presented approach can be used for other reachability problems. For example, the linearization error of nonlinear reachability problems based on the Lagrangian remainder (see [4]) can be drastically tightened when computed as in Theorem 1. Also, the linear part of the Taylor series in Proposition 3 can be used to map normal vectors of polyhedral reachable sets to distinctive normal vectors of the guard intersection, which is useful for the classical geometric approach when bounding the intersection using linear programming.

Acknowledgments

This research was supported in part by U.S. National Science Foundation grant number CCF-0926181 and the U.S. Air Force Office of Scientific Research grant number FA9550-06-1-0312.

8. REFERENCES

[1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. Dissertation, Technische Universität München, 2010. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[2] M. Althoff and B. H. Krogh. Zonotope bundles for the efficient computation of reachable sets. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 6814–6821, 2011.

[3] M. Althoff, C. Le Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Hybrid Systems: Computation and Control*, pages 93–102, 2011.

[4] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[5] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry: Theory and Applications*, 7:265–301, 1997.

[6] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72:3–21, 2008.

[7] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 73–88. Springer, 2000.

[8] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003.

[9] E. Clarke, A. Fehnker, Z. Han, B. Krogh, O. Stursberg, and M. Theobald. *Tools and Algorithms for the Construction and Analysis of Systems*, chapter Verification of Hybrid Systems based on Counterexample-Guided Abstraction Refinement, pages 192–207. LNCS 2619. Springer, 2003.

[10] W. Damm, S. Disch, H. Hungar, S. Jacobs, J. Pang, F. Pigorsch, C. Scholl, U. Waldmann, and B. Wirtz. Exact state set representations in the verification of linear hybrid systems with large discrete state space. In *Proc. of the 5th Int. Symposium on Automated Technology for Verification and Analysis*, pages 425–440, 2007.

[11] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, LNCS 6806, pages 379–395. Springer, 2011.

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

[13] A. Girard and C. Le Guernic. Efficient reachability analysis for linear systems using support functions. In *Proc. of the 17th IFAC World Congress*, pages 8966–8971, 2008.

[14] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proc. of Hybrid Systems: Computation and Control*, LNCS 4981, pages 215–228. Springer, 2008.

[15] 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*, LNCS 3927, pages 257–271. Springer, 2006.

[16] A. Hamadeh and J. Goncalves. Reachability analysis of continuous-time piecewise affine systems. *Automatica*, 44(12):189–3194, 2008.

[17] T. A. Henzinger and P.-H. Ho. HyTech: The Cornell Hybrid Technology Tool. In *Hybrid Systems II*, LNCS 999, pages 265–294. Springer, 1995.

[18] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HyTech: Hybrid systems analysis using interval numerical methods. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 130–144. Springer, 2000.

[19] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

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

[21] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 1569, pages 137–151. Springer, 1999.

[22] A. Lagerberg. A benchmark on hybrid control of an automotive powertrain with backlash. Technical Report R005/2007, Signals and Systems, Chalmers University of Technology, 2007.

[23] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50:947–957, 2005.

[24] N. S. Nedialkov and M. von Mohrenschildt. Rigorous simulation of hybrid dynamic systems with symbolic and interval methods. In *Proc. of the American Control Conference*, pages 140–147, 2002.

[25] M. Nordin and P.-O. Gutman. Controlling mechanical systems with backlash – a survey. *Automatica*, 38:1633 – 1649, 2002.

[26] E.-A. M. A. Rabeih. *Torsional Vibration Analysis of Automotive Drivelines*. PhD thesis, University of Leeds, 1997.

[27] S. V. Raković, P. Grieder, M. Kvasnica, D. Q. Mayne, and M. Morari. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *Proc. of the 43rd IEEE Conference on Decision and Control*, pages 1418–1423, 2004.

[28] N. Ramdani and N. S. Nedialkov. Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2010.

[29] F. M. Schlaepfer and F. C. Schweppe. Continuous-time state estimation under disturbances bounded by convex sets. *IEEE Transactions on Automatic Control*, 17(2):197–205, 1972.

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