# Reachability Analysis for Hybrid Systems with Nonlinear Guard Sets

Niklas Kochdumper
niklas.kochdumper@tum.de
Technische Universität München

Matthias Althoff
althoff@tum.de
Technische Universität München

## ABSTRACT

Reachability analysis is one of the most important methods for formal verification of hybrid systems. The main difficulty for hybrid system reachability analysis is to calculate the intersection between reachable set and guard sets. While there exist several approaches for guard sets defined by hyperplanes or polytopes, only few methods are able to handle nonlinear guard sets. In this work we present a novel approach to tightly enclose the intersections of reachable sets with nonlinear guard sets. One major advantage of our method is its polynomial complexity with respect to the system dimension, which makes it applicable for high-dimensional systems. Furthermore, our approach can be combined with different reachability algorithms for continuous systems due to its modular design. We demonstrate the advantages of our novel approach compared to existing methods with numerical examples.

## CCS CONCEPTS

• **General and reference → Verification**; • **Theory of computation → Timed and hybrid models**

.

## KEYWORDS

hybrid systems, nonlinear level sets, Taylor models polynomial zonotopes

**ACM Reference Format:**
Niklas Kochdumper and Matthias Althoff. 2020. Reachability Analysis for Hybrid Systems with Nonlinear Guard Sets. In *23rd ACM International Conference on Hybrid Systems: Computation and Control (HSCC '20), April 22–24, 2020, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3365365.3382194

## 1 INTRODUCTION

For many safety-critical systems like autonomous vehicles, robots collaborating with humans, and automated medical systems, it is often necessary to prove correct functionality using formal verification techniques. In particular, formal verification of hybrid systems has huge practical relevance since most systems exhibit mixed discrete and continuous dynamics due to the interplay between physical behavior and digital control. One commonly-used method for formal verification is reachability analysis, which computes the states reachable by the system. In this work, we introduce a novel technique for over-approximative reachability analysis of hybrid systems with nonlinear guard sets. Our approach is applicable to a broad class of systems, it enables the computation of tight over-approximations of reachable sets, and it scales well with the number of system dimensions.

### 1.1 State of the Art

Reachability analysis for hybrid systems typically relies on the computation of reachable sets for continuous dynamics. Most reachability algorithms for linear continuous systems are based on the propagation of sets [21, 23, 24, 32, 45]. Typical set representations are polytopes [21], zonotopes [23], ellipsoids [32], support functions [24], star sets [10], level sets [40], Taylor models [15], and polynomial zonotopes [1]. Other approaches compute reachable sets based on simulations [10, 20]. Reachability algorithms for nonlinear continuous systems can be categorized into four main groups: invariant generation [31, 35, 39], optimization-based approaches [17, 40], abstraction in solution space [15, 20, 42], and abstraction in state space [1, 3, 7, 19].

The main challenge in reachability analysis for hybrid systems is the computation of the intersection between the reachable set and guard sets. For guards sets given by polyhedra or hyperplanes, several methods for intersection computation have been developed. A straightforward approach is to compute the intersection between the reachable set and the guard set geometrically, which is done by the tools Flow* [16], SpaceEx [21], HyDRA [44], and Julia Reach [12]. The method in [22] computes the intersection between reachable sets represented by support functions and the guard set by solving several convex minimization problems. To avoid an explosion in computation time, the sets resulting from partial intersections are often unified by computing convex hulls [21, 22]. Since the computation of convex hulls is computationally demanding, many approaches unify partial intersections by simpler sets or completely avoid the unification: In [4], the union of the partial intersections is enclosed by bundles of parallelotopes. The work in [25] shows how the intersection between multiple zonotopes and a hyperplane can be efficiently enclosed by a template polyhedron. The tool Hylaa [9] reduces the over-approximation resulting from the unification by applying a backtracking scheme that splits previously computed reachable sets. The method in [8] completely avoids the need for unification by scaling the system dynamics in such a way that only the reachable set for one time step intersects the guard set. For high-dimensional systems not only the unification, but also the computation of geometric intersections is computationally expensive. The technique in [5] avoids both unification and geometric intersection computation, by directly mapping the reachable set

*HSCC '20, April 22–24, 2020, Sydney, NSW, Australia*

ACM ISBN 978-1-4503-7018-9/20/04…$15.00
https://doi.org/10.1145/3365365.3382194

onto the guard set. The tool CORA [2] implements the methods [4], [25], and [5]. The tool Isabelle/HOL [27] applies the method in [25].

Currently, only a few approaches exist for a more general class of hybrid systems which model guard sets as nonlinear level sets. For some simple cases, nonlinear guard sets can be enclosed by multiple polytopes, which makes it possible to use the approaches in [21] and [4]. The method in [18], implemented in the tool Ariadne [11], calculates the intersection by adding constraints to the initial set when hitting a guard set. Similarly, the approach in [15] uses the constraints imposed by the guard intersection to contract the set of initial states, which then yields a Taylor model that encloses the intersection with the guard set. Another strategy is to determine the time interval in which the reachable set intersects the guard set, and then take the whole reachable set for the time interval as an over-approximation of the intersection with the guard set. While this technique works well for *guaranteed integration* methods that enclose only a single trajectory rather then a set of trajectories [41], it is often too conservative for reachability analysis. To compute tight enclosures of reachable sets, the approach in [43] uses a technique similar to [41], but additionally creates partitions in time until a user-defined precision is achieved; however, propagating the reachable sets for all partitions is computationally expensive. For this reason, [37] improves the approach in [43] by unifying the resulting sets from all partitions with an enclosing box. Since box enclosures result in large over-approximation errors, the approach in [36] unifies parallel sets with an enclosing zonotope instead. However, since [36] requires the computation of zonotope vertices, the approach has exponential complexity with respect to the system dimension.

## 1.2 Contribution

We present a novel approach to tightly enclose the intersection between the reachable set and nonlinear guard sets. One major advantage of our method compared to previous approaches is that the computational complexity is only polynomial with respect to the number of system dimensions, which enables the verification of high-dimensional systems. Due to its modular design, our method can be combined with different reachability algorithms for continuous system. As we demonstrate with several numerical examples, our novel method reduces the computation time and improves the accuracy of the reachable set compared to previous approaches.

## 1.3 Notation

Sets are denoted by calligraphic letters, matrices by uppercase letters, vectors by lowercase letters, lists by bold uppercase letters, and set operations by typewriter font (e.g., `center`). Given a vector $b \in \mathbb{R}^n$, $b_{(i)}$ refers to the $i$-th entry. Given a matrix $A \in \mathbb{R}^{n \times m}$, $A_{(i,\cdot)}$ represents the $i$-th matrix row, $A_{(\cdot,j)}$ the $j$-th column, and $A_{(i,j)}$ the $j$-th entry of matrix row $i$. The concatenation of two matrices $C$ and $D$ is denoted by $[C\ D]$. Vectors of zeros and ones are denoted by $\mathbf{0}_n \in \mathbb{R}^n$ and $\mathbf{1}_n \in \mathbb{R}^n$, and the empty matrix is denoted by $[\ ]$. The Minkowski addition of two sets $\mathcal{S}_1 \subset \mathbb{R}^n$ and $\mathcal{S}_2 \subset \mathbb{R}^n$ is defined as $\mathcal{S}_1 \oplus \mathcal{S}_2 = \left\{ s_1 + s_2 \;\middle|\; s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2 \right\}$, and the Cartesian product of two sets $\mathcal{S}_1 \subset \mathbb{R}^n$ and $\mathcal{S}_2 \subset \mathbb{R}^m$ is defined as $\mathcal{S}_1 \times \mathcal{S}_2 = \left\{ [s_1\ s_2]^T \;\middle|\; s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2 \right\}$. We introduce an $n$-dimensional box as $\mathcal{I} := [l, u]$, $\forall i\ l_{(i)} \leq u_{(i)}$, $l, u \in \mathbb{R}^n$, and $\emptyset$ denotes the empty set. The Nabla operator is defined as $\nabla = \sum_{i=1}^{n} e_i \frac{\partial}{\partial x_{(i)}}$, with $x \in \mathbb{R}^n$ and $e_i \in \mathbb{R}^n$ being orthogonal unit vectors.

## 2 PROBLEM FORMULATION

We present a novel technique to compute the reachable set for a hybrid system with nonlinear guard sets. Hybrid systems are modeled as hybrid automata in this work:

Definition 1. *(Hybrid Automaton) A hybrid automaton $H$ with $p$ discrete modes consists of:*

- *A list $\mathbf{F} = (f_1(\cdot), \dots, f_p(\cdot))$ of differential equations $\dot{x}(t) = f_i(\cdot)$ describing the continuous dynamics in each mode $i = 1, \dots, p$.*
- *A list $\mathbf{S} = (\mathcal{S}_1, \dots, \mathcal{S}_p)$ of invariant sets $\mathcal{S}_i \subset \mathbb{R}^n$ for each mode $i = 1, \dots, p$.*
- *A list $\mathbf{T} = (T_1, \dots, T_q)$ of transitions $T_i = \langle \mathcal{G}_i, r_i(\cdot), s_i, d_i \rangle_T$, $i = 1, \dots, q$ between discrete modes, where $\mathcal{G}_i \subset \mathbb{R}^n$ is a guard set, $r_i : \mathbb{R}^n \to \mathbb{R}^n$ is a reset function, and $s_i, d_i \in \{1, \dots, p\}$ are indices of the source and target modes, respectively.*

□

The state of a hybrid automaton consists of the continuous state $x(t) \in \mathbb{R}^n$ and the discrete state $m(t) \in \{1, \dots, p\}$. The evolution of a hybrid automaton is described informally as follows: Given an initial continuous state $x_0 = x(0)$ and an initial discrete state $m_0 = m(0)$ with $x_0 \in \mathcal{S}_{m_0}$, the continuous state $x(t)$ evolves according to the flow function $f_{m_0}(\cdot)$ of the mode $m_0$. If $x(t)$ is within the guard set $\mathcal{G}_i$ of a transition $T_i = \langle \mathcal{G}_i, r_i(\cdot), s_i, d_i \rangle_T \in \mathbf{T}$ with $s_i = m_0$, the transition to the mode $d_i$ is taken and the continuous state $x(t)$ is updated according to the reset function $r_i(\cdot)$. Afterward, the evolution of the continuous state continues according to the flow function $f_{d_i}(\cdot)$ of mode $d_i$ until the next transition is taken. We denote the trajectory of the continuous state for the evolution of the hybrid automaton described above by $\xi(t, x_0, m_0, u(\cdot))$, where $u(\cdot) \in \mathbb{R}^m$ is an input trajectory.

Given the behavior of a hybrid automaton, we are interested in the set of reachable states:

Definition 2. *(Reachable Set) The reachable set at time $t$ of a hybrid automaton $H$ for a set of initial continuous states $\mathcal{X}_0 \subset \mathbb{R}^n$, an initial mode $m_0$, and a set of uncertain inputs $\mathcal{U} \subset \mathbb{R}^m$ is*

$$\mathcal{R}^e(t) := \Big\{ \xi(t, x_0, m_0, u(\cdot)) \;\Big|\; x_0 \in \mathcal{X}_0, \\ \forall \tau \in [0, t] : \; u(\tau) \in \mathcal{U} \Big\}.$$

□

The superscript $e$ on $\mathcal{R}^e(t)$ denotes the exact reachable set, which can be computed for only a limited class of hybrid automata [33]. We therefore compute a tight over-approximation $\mathcal{R}(t) \supseteq \mathcal{R}^e(t)$.

In this work we consider hybrid automata for which the guard sets $\mathcal{G} \subset \mathbb{R}^n$ are given by nonlinear level sets defined as

$$\mathcal{G} = \left\{ x \in \mathbb{R}^n \;\middle|\; g(x) = 0 \right\}, \tag{1}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a Lipschitz continuous function. We use the shorthand $\mathcal{G} = \langle g(\cdot) \rangle_G$. The invariant sets $\mathcal{S} \subset \mathbb{R}^n$ are defined by an intersection of several nonlinear inequality constraints

$$\mathcal{S} = \left\{ x \in \mathbb{R}^n \;\middle|\; s(x) \leq \mathbf{0}_w \right\}, \tag{2}$$

where $s : \mathbb{R}^n \to \mathbb{R}^w$ is a Lipschitz continuous function and the operator $\leq$ in (2) denotes that all entries $s_{(i)}(x)$ of $s(x)$ satisfy $s_{(i)}(x) \leq 0$. We use the shorthand $\mathcal{S} = \langle s(\cdot) \rangle_S$. The reset functions $r : \mathbb{R}^n \to \mathbb{R}^n$ are nonlinear, Lipschitz continuous functions.

To tightly enclose intersections with a nonlinear guard set, a non-convex set representation is required. We therefore use Taylor models or polynomial zonotopes for representing reachable sets, which allows us to combine our approach with many different reachability algorithms for continuous systems:

Definition 3. *(Taylor Model) Given a polynomial vector field $p : \mathbb{R}^n \to \mathbb{R}^n$, a box domain $\mathcal{I} \subset \mathbb{R}^n$, and a box remainder $\mathcal{B} \subset \mathbb{R}^n$, a Taylor model $\mathcal{T}(x) \subset \mathbb{R}^n$ is defined as*

$$\forall x \in \mathcal{I} : \; \mathcal{T}(x) := \left\{ p(x) + b \;\middle|\; b \in \mathcal{B} \right\}.$$

□

We use the shorthand $\mathcal{T}(x) = \langle p(x), \mathcal{B}, \mathcal{I} \rangle_{TM}$. Taylor models are often defined for a normalized domain $\mathcal{I} = [-\mathbf{1}_n, \mathbf{1}_n]$. We do not use a normalized domain since this simplifies the proof of our main theorem presented later.

For polynomial zonotopes, we use the sparse representation from [30]:

Definition 4. *(Polynomial Zonotope) Given a generator matrix of dependent generators $G \in \mathbb{R}^{n \times h}$, a generator matrix of independent generators $G_I \in \mathbb{R}^{n \times l}$, and an exponent matrix $E \in \mathbb{Z}_{\geq 0}^{v \times h}$, a polynomial zonotope is defined as*

$$\mathcal{PZ} := \left\{ \sum_{i=1}^{h} \left( \prod_{k=1}^{v} \alpha_k^{E_{(k,i)}} \right) G_{(\cdot,i)} + \sum_{j=1}^{l} \beta_j G_{I(\cdot,j)} \;\middle|\; \alpha_k, \beta_j \in [-1, 1] \right\}.$$

□

We use the shorthand $\mathcal{PZ} = \langle G, G_I, E \rangle_{PZ}$. Any Taylor model can be equivalently represented by a polynomial zonotope [30, Prop. 4], but not every polynomial zonotope can be represented by a Taylor model [30, Corrollary 1]. For instance, a polytope can be represented by a polynomial zonotope [30, Thm. 1], but not necessarily by a Taylor model.

## 3 BASIC PROCEDURE

We briefly explain the computation of the reachable set for the continuous evolution, and then summarize the main steps of our novel approach for handling discrete transitions.

### 3.1 Continuous Dynamics

One of the major advantages of our novel approach is its modular design, which allows us to combine the approach with many different reachability analysis algorithms for continuous systems. The only requirement is that the used reachability algorithm can compute with reachable sets represented as Taylor models or polynomial zonotopes. Since polynomial zonotopes are closed under nearly all relevant set representations [30], most algorithms fulfill this requirement. The modularity allows one to consider linear continuous dynamics using the algorithm in [23], nonlinear continuous dynamics using the algorithms in [1], [3] or [15], and nonlinear continuous dynamics with algebraic constraints using the algorithm in [6]. All these algorithms compute the reachable set for consecutive time intervals $\tau_s = [t_s, t_{s+1}]$ with $t_{s+1} = t_s + \Delta t$ so that the reachable set for a time horizon $t_f$ is given as $\mathcal{R}([0, t_f]) = \bigcup_{s=0}^{t_f/\Delta t - 1} \mathcal{R}(\tau_s)$. For the numerical examples in Sec. 6, we use the algorithm in [3].

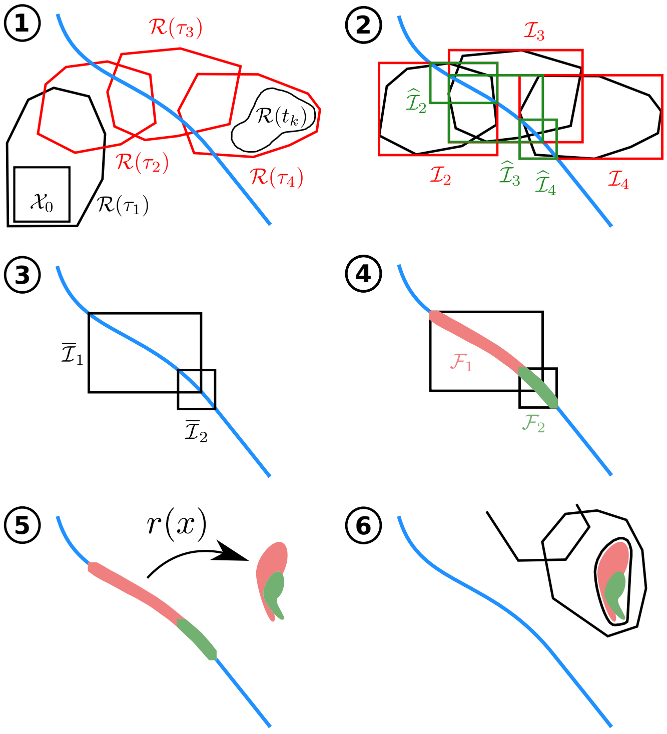

**Figure 1: Visualization of the procedure applied to calculate guard intersections.**

### 3.2 Discrete Transitions

The difficulty in reachability analysis for hybrid systems is the computation of reachable sets across discrete transitions. We follow the steps visualized in Fig. 1:

① We first compute the reachable set as described in Sec. 3.1 until the reachable set is completely located outside the current invariant set or the final time $t_f$ is reached. In addition, we determine the time steps for which the corresponding reachable set intersects the guard set. Our approach also works when only the reachable set within an invariant is propagated, which in some cases significantly reduces the conservatism, but is computationally more expensive.

② To obtain a rough over-approximation of the guard intersection using computationally cheap methods, we first enclose each reachable set intersecting the guard set with a box $\mathcal{I}$. Afterward, we contract the obtained boxes so that they tightly enclose the intersection with the guard set.

③ To avoid the computationally expensive parallel propagation of reachable sets, we enclose the union of the contracted boxes $\widehat{\mathcal{I}}$ by a single box $\overline{\mathcal{I}}$. We introduce an upper bound $\mu$ for the number of boxes that are unified to reduce the conservatism. For the example shown in Fig. 1, a value of $\mu = 2$ is used.

④ We tightly enclose the intersection of the guard set with the previously-obtained box $\overline{\mathcal{I}}$ by a Taylor model or polynomial zonotope $\mathcal{F}$.

⑤ Afterward, we apply the reset function $r(x)$ to the previously obtained Taylor model or polynomial zonotope $\mathcal{F}$.

⑥ Due to the upper bound $\mu$, we might obtain parallel sets, which we unify by a single set to avoid propagating several sets in parallel. The reason for using the upper bound $\mu$ in Step 3 followed by the unification of parallel sets in Step 6 is that the unification in Step 6 significantly increases the representation size of the resulting set if many parallel sets are unified. With the early partial unification in Step 3, we avoid this issue.

The steps of this procedure are explained in detail in the next section. For the case that the reachable set intersects several guard sets, we compute the intersection separately for each guard set using the presented approach. If the computation of the contracted boxes in step 2 is adapted appropriately, this does not lead to large over-approximations.

## 4 GUARD INTERSECTION

In this section we present the steps for our novel computation of guard intersections.

### 4.1 Intersection Detection

First, we describe Step 1 of the procedure from Sec. 3.2. We apply range-bounding to detect intersections with invariant and guard sets:

Definition 5. *(Range Bounding) Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a set $\mathcal{X} \in \mathbb{R}^n$, the range-bounding operation*

$$\mathsf{bound}\Big(f(\cdot), \mathcal{X}\Big) \supseteq \left[\min_{x \in \mathcal{X}} f(x),\ \max_{x \in \mathcal{X}} f(x)\right]$$

*returns an over-approximation of the exact bounds.* □

To check if the reachable set $\mathcal{R}(t_k)$ of the current time step $k$ is located outside of the current invariant set $\mathcal{S} = \langle s(\cdot) \rangle_S$, we apply the range-bounding operation to each subfunction $s_{(i)}(\cdot)$ of $s(\cdot)$ to obtain the bounds

$$[l_i, u_i] = \mathsf{bound}\Big(s_{(i)}(\cdot), \mathcal{R}(t_k)\Big),\ i = 1, \dots, w.$$

According to the definition of the invariant set in (2), it is guaranteed that the reachable set is located outside the invariant set if $\forall i \in \{1, \dots, w\} :\ l_i > 0$ holds.

To determine the time steps in which the reachable set intersects a guard set $\mathcal{G} = \langle g(\cdot) \rangle_G$, we iterate over $\mathcal{R}(\tau_j)$ for all time steps $j = 1, \dots, k$ and compute

$$[l, u] = \mathsf{bound}\Big(g(\cdot), \mathcal{R}(\tau_j)\Big).$$

According to the definition of the guard set in (1), the reachable set potentially intersects the guard set if $l \leq 0 \wedge u \geq 0$.

The conservatism of the intersection detection depends solely on the range-bounding technique used. The simplest method is to enclose the reachable set by a box $\mathcal{I} \supseteq \mathcal{R}(t_k)$, and then use interval arithmetic [29] for range-bounding. Another approach is to enclose the reachable set by a Taylor model, and than apply the approach in [38, Alg. 1] for range-bounding.

### 4.2 Box Contraction

Next, we consider Step 2 of the procedure from Sec. 3.2. The contraction of a box domain to tightly enclose the solutions of a nonlinear constraint is a well-studied problem. A contractor is defined as follows:

Definition 6. *(Contractor) Given a box $\mathcal{I} \subset \mathbb{R}^n$ and a nonlinear function $c : \mathbb{R}^n \rightarrow \mathbb{R}$ which defines the constraint $c(x) = 0$, the operation* `contract` *returns a box that satisfies*

$$\texttt{contract}(\mathcal{I}) \subseteq \mathcal{I}$$

*and*

$$\forall x \in \mathcal{I},\ c(x) = 0 \Rightarrow x \in \texttt{contract}(\mathcal{I}).$$

□

Many different implementations of contractors exist: The *Box* algorithm in [26] updates the bounds for each dimension using an univariate interval Newton iterations. The work in [46] improves the approach from [26] for polynomial constraints by considering extremal functions. The *HC4revise* algorithm in [34] applies forward-backward traversion of the syntax tree to contract the domains. An overview of contractor programming is provided in [13]. For the numerical examples in Sec. 6 we use the *HC4revise* algorithm since it is exact for single-use expressions.

### 4.3 Intersection Computation

We now describe step 4 of the procedure from Sec. 3.2. Our guard intersection approach is based on the novel finding that a specific type of polynomial level set intersected with a box can be represented as a Taylor model or polynomial zonotope. We demonstrate this with an example:

Example 1. *The intersection between the box $\mathcal{I} = [-1, 1] \times [0, 2]$ and the polynomial level set*

$$\mathcal{LS} = \Big\{ x \in \mathbb{R}^2 \mid x_{(2)} = 2x_{(1)}^2 \Big\}$$

*can be equivalently represented by the Taylor model*

$$\mathcal{I} \cap \mathcal{LS} = \Big\{ \mathcal{T}(x) \,\Big|\, x \in \mathcal{I} \Big\},\ \mathcal{T}(x) = \left\langle \begin{bmatrix} x_{(1)} \\ 2x_{(1)}^2 \end{bmatrix}, \emptyset, \mathcal{I} \right\rangle_{TM}$$

*or the polynomial zonotope*

$$\mathcal{I} \cap \mathcal{LS} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0 \\ 2 \end{bmatrix} \alpha_1^2 \,\middle|\, \alpha_1 \in [-1, 1] \right\}.$$

This is generalized in the main theorem of our paper:

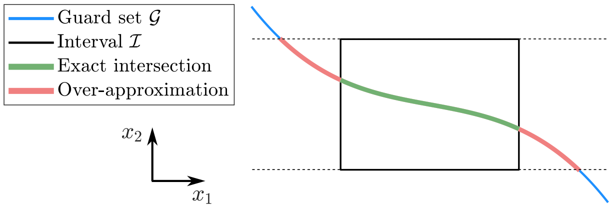

**Figure 2: Over-approximative Taylor model (red) calculated according to Thm. 1 for the case that condition (4) is not fulfilled. With a proper contraction of $\mathcal{I}$ in $x_2$-direction (see Sec. 4.2) the result would be exact.**

Theorem 1. *The intersection between a box $\mathcal{I} = [l, u] \subset \mathbb{R}^n$ and a nonlinear level set*

$$\mathcal{LS} = \left\{ x \in \mathbb{R}^n \,\middle|\, x_{(k)} = p_k(x) \right\}$$

*with $k \in \{1, \dots, n\}$ can be tightly enclosed by the Taylor model*

$$\mathcal{I} \cap \mathcal{LS} \subseteq \left\{ \mathcal{T}(x) \,\middle|\, x \in \mathcal{I} \right\}$$

$$\text{with } \mathcal{T}(x) = \left\langle \underbrace{\begin{bmatrix} \begin{bmatrix} x_{(1)} & \dots & x_{(k-1)} \end{bmatrix}^T \\ p_k(x) \\ \begin{bmatrix} x_{(k+1)} & \dots & x_{(n)} \end{bmatrix}^T \end{bmatrix}}_{h(x)}, \emptyset, \mathcal{I} \right\rangle_{TM}. \tag{3}$$

*If the polynomial function $p_k(x)$ defining the level set $\mathcal{LS}$ satisfies*

$$\min_{x \in [l,u]} p_k(x) \geq l_{(k)} \text{ and } \max_{x \in [l,u]} p_k(x) \leq u_{(k)} \tag{4}$$

*the calculated Taylor model $\mathcal{T}(x)$ exactly represents the intersection $\mathcal{I} \cap \mathcal{LS}$; otherwise, $\mathcal{T}(x)$ encloses the intersection $\mathcal{I} \cap \mathcal{LS}$.*

Proof. The idea of the proof is to replace variable $x_{(k)}$ with the function $p_k(x)$, which defines the level set. If condition (4) is fulfilled the intersection $\mathcal{I} \cap \mathcal{LS}$ can equivalently be expressed as

$$\mathcal{I} \cap \mathcal{LS} = \left\{ x \in \mathcal{I} \,\middle|\, x_{(k)} = p_k(x) \right\} \overset{(4)}{=}$$

$$\left\{ \begin{bmatrix} \begin{bmatrix} x_{(1)} & \dots & x_{(k-1)} \end{bmatrix}^T \\ p_k(x) \\ \begin{bmatrix} x_{(k+1)} & \dots & x_{(n)} \end{bmatrix}^T \end{bmatrix} \,\middle|\, x \in \mathcal{I} \right\} \overset{(3)}{=} \left\{ \mathcal{T}(x) \mid x \in \mathcal{I} \right\}.$$

If condition (4) is not fulfilled, the calculated Taylor model encloses the intersection $\mathcal{I} \cap \mathcal{LS}$ since

$$\left\{ p_k(x) \,\middle|\, x \in \underbrace{[l, u]}_{\mathcal{I}} \right\} \supseteq [l_{(k)}, u_{(k)}].$$

A visualization of the over-approximation error is shown in Fig. 2. □

For a concise notation, we introduce the shorthand $h(x) \leftarrow$ `intersect`$(\mathcal{I}, \mathcal{LS})$ for the intersection $\mathcal{I} \cap \mathcal{LS}$ according to Thm. 1, where the function $h(x)$ is defined as in (3). Thm. 1 equally holds for polynomial zonotopes since according to [30, Prop. 4] the set defined by every Taylor model can equivalently be represented by a polynomial zonotope.

Next, we introduce the `taylor` operator, which we will use in subsequent derivations:

Definition 7. *(Taylor Series) Given a Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$, a box $\mathcal{I} \subset \mathbb{R}^n$, and the Taylor order $\kappa \in \mathbb{N}$, the operator*

$$a, b, c, v(\cdot), l, u \leftarrow \texttt{taylor}(f(\cdot), \mathcal{I}, \kappa)$$

*returns the parameters $a, c \in \mathbb{R}^n$, $b, l, u \in \mathbb{R}$, and the function $v : \mathbb{R}^n \to \mathbb{R}$ of the Taylor series expansion with order $\kappa$ of $f(\cdot)$ around the expansion point $c =$ `center`$(\mathcal{I})$:*

$$\forall x \in \mathcal{I} : f(x) \in \underbrace{f(c)}_{b} + \underbrace{\frac{\partial f(\tilde{x})}{\partial \tilde{x}}\bigg|_{\tilde{x}=c}}_{a^T} (x - c)$$

$$+ \underbrace{\sum_{i=2}^{\kappa} \frac{\left( (x-c)^T \nabla \right)^i f(\tilde{x})}{i!}\Bigg|_{\tilde{x}=c}}_{v(x)} \oplus [l, u],$$

*where the interval $[l, u] \supseteq \mathcal{L}$ results from the over-approximative evaluation of the Lagrange remainder*

$$\mathcal{L} = \left\{ \frac{\left( (x-c)^T \nabla \right)^{\kappa+1} f(\tilde{x})}{(\kappa+1)!}\Bigg|_{\tilde{x} \in \mathcal{I}} \,\middle|\, x \in \mathcal{I} \right\}$$

*using interval arithmetics.* □

Based on Thm. 1, we now show how the intersection between a guard set $\mathcal{G} = \langle g(\cdot) \rangle_G$ defined by a non-polynomial function $g(\cdot)$ and a box $\mathcal{I}$ can be tightly enclosed with a Taylor model or polynomial zonotope. We distinguish the case where the equality constraint $g(x) = 0$ is symbolically solvable for one variable $x_{(k)}$ from the case where it is not. The constraint $g(x) = 0$ is symbolically solvable for the variable $x_{(k)}$ if the set $\{x \mid g(x) = 0\}$ can be equivalently represented as

$$\left\{ x \,\middle|\, x_{(k)} = \widehat{g}(x) \right\} \text{ with } \frac{\partial \widehat{g}(x)}{\partial x_{(k)}} = 0, \tag{5}$$

where $\partial \widehat{g}(x) / \partial x_{(k)} = 0$ implies that $\widehat{g}(x)$ does not depend on $x_{(k)}$ for all $x$. We demonstrate this with an example:

Example 2. *The guard set*

$$\mathcal{G} = \left\{ x \in \mathbb{R}^2 \,\middle|\, \underbrace{x_{(2)} x_{(1)} + \sin(x_{(1)})}_{g(x)} = 0 \right\}$$

*can be equivalently represented as*

$$\mathcal{G} = \left\{ x \in \mathbb{R}^2 \,\middle|\, x_{(2)} = \underbrace{-\frac{\sin(x_{(1)})}{x_{(1)}}}_{\widehat{g}(x)} \right\}$$

*since the constraint $g(x) = 0$ is symbolically solvable for $x_{(2)}$.*

We first consider the case where the equality constraint is symbolically solvable for one variable:

Proposition 1. *We consider a box $\mathcal{I}$, a guard set $\mathcal{G} = \langle g(\cdot) \rangle_G = \left\{ x \,\middle|\, g(x) = 0 \right\}$ which can be equivalently represented as $\mathcal{G} = \left\{ x \,\middle|\, x_{(k)} = \widehat{g}(x) \right\}$ with $\partial \widehat{g}(x) / \partial x_{(k)} = 0$, and the Taylor order $\kappa \in \mathbb{N}$. To tightly* *enclose the intersection $\mathcal{I} \cap \mathcal{G}$, we first compute the Talyor series expansion of $\widehat{g}(x)$:*

$$a, b, c, v(\cdot), l, u \leftarrow \texttt{taylor}(\widehat{g}(\cdot), \mathcal{I}, \kappa). \tag{6}$$

*Next, we compute the intersection*

$$h(x) \leftarrow \texttt{intersect}(\mathcal{I}, \mathcal{LS}), \ \mathcal{LS} = \left\{ x \, \middle| \, x_{(k)} = p_k(x) \right\} \tag{7}$$

*of $\mathcal{I}$ and $\mathcal{LS}$ according to Thm. 1, where the function*

$$p_k(x) = b + a^T(x - c) + v(x)$$

*represents the polynomial part of the Taylor series expansion. Finally, the intersection between $\mathcal{I}$ and $\mathcal{G}$ can be tightly enclosed by the Taylor model*

$$\mathcal{I} \cap \mathcal{G} \subseteq \left\{ \mathcal{T}(x) \, \middle| \, x \in \mathcal{I} \right\}$$
$$\text{with } \mathcal{T}(x) = \left\langle h(x), \underbrace{\left[ \begin{bmatrix} \mathbf{0}_{k-1} \\ l \\ \mathbf{0}_{n-k} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{k-1} \\ u \\ \mathbf{0}_{n-k} \end{bmatrix} \right]}_{\mathcal{B}}, \mathcal{I} \right\rangle_{TM}. \tag{8}$$

Proof. With the Taylor series expansion of $\widehat{g}(\cdot)$ in (6) the set $\left\{ x \in \mathcal{I} \, \middle| \, x_{(k)} = \widehat{g}(x) \right\}$ can be tightly enclosed by

$$\left\{ x \in \mathcal{I} \, \middle| \, x_{(k)} = \widehat{g}(x) \right\} \subseteq \left\{ x \in \mathcal{I} \, \middle| \, x_{(k)} \in p_k(x) \oplus [l, u] \right\}. \tag{9}$$

Using (9), the intersection between the box $\mathcal{I}$ and the guard set $\mathcal{G}$ can be formulated as

$$\begin{aligned}
&\mathcal{I} \cap \mathcal{G} \\
&= \left\{ x \in \mathcal{I} \, \middle| \, x_{(k)} = \widehat{g}(x) \right\} \overset{(9)}{\subseteq} \left\{ x \in \mathcal{I} \, \middle| \, x_{(k)} \in p_k(x) \oplus [l, u] \right\} \\
&= \left\{ \begin{bmatrix} \begin{bmatrix} x_{(1)} & \dots & x_{(k-1)} \end{bmatrix}^T \\ x_{(k)} \\ \begin{bmatrix} x_{(k+1)} & \dots & x_{(n)} \end{bmatrix}^T \end{bmatrix} \, \middle| \, x \in \mathcal{I}, \ x_{(k)} \in p_k(x) \oplus [l, u] \right\} \\
&= \left\{ \underbrace{\begin{bmatrix} \begin{bmatrix} x_{(1)} & \dots & x_{(k-1)} \end{bmatrix}^T \\ p_k(x) \\ \begin{bmatrix} x_{(k+1)} & \dots & x_{(n)} \end{bmatrix}^T \end{bmatrix}}_{\overset{\text{Thm. 1}}{=} h(x)} + \begin{bmatrix} \mathbf{0}_{k-1} \\ s \\ \mathbf{0}_{n-k} \end{bmatrix} \, \middle| \, x \in \mathcal{I}, \ s \in [l, u] \right\} \\
&\overset{(8)}{=} \left\{ \underbrace{h(x) + b \, \middle| \, b \in \mathcal{B},}_{\overset{\text{Def. 3}}{=} \mathcal{T}(x)} \ x \in \mathcal{I} \right\} = \left\{ \mathcal{T}(x) \, \middle| \, x \in \mathcal{I} \right\}.
\end{aligned}$$

□

If the equality constrained is solvable for multiple variables we choose the variable that results in the tightest enclosure of the set $\mathcal{I} \cap \mathcal{G}$. We demonstrate the computation of the intersection according to Prop. 1 with an example:

Example 3. *We consider the guard set*

$$\mathcal{G} = \left\{ x \in \mathbb{R}^2 \, \middle| \, e^{x_{(1)}} + 0.2 \, x_{(1)}^2 - x_{(2)} - 1 = 0 \right\}$$

*and the box $\mathcal{I} = [-2, -1] \times [-0.5, 0]$. Computation of a Taylor model enclosing the intersection according to Prop. 1 using a Taylor series of order $\kappa = 2$ yields*

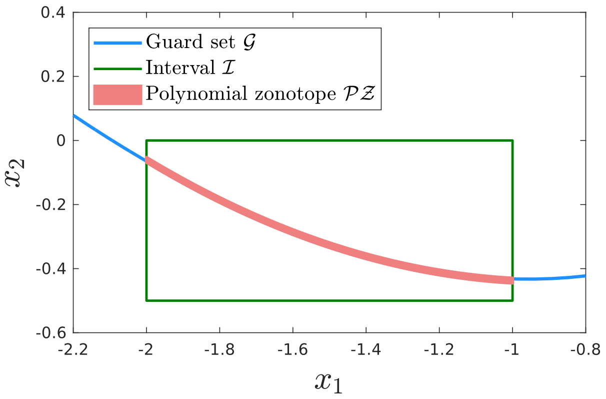

**Figure 3: Visualization of the computed intersection from Example 3.**

$$\mathcal{T}(x) = \left\langle \begin{bmatrix} x_{(1)} \\ -0.1911 + 0.5579 \, x_{(1)} + 0.3116 \, x_{(1)}^2 \end{bmatrix}, \right.$$
$$\left. \left[ \begin{bmatrix} 0 \\ -0.0077 \end{bmatrix}, \begin{bmatrix} 0 \\ 0.0077 \end{bmatrix} \right], \mathcal{I} \right\rangle_{TM},$$

*which can be equivalently represented by the polynomial zonotope*

$$\mathcal{PZ} = \left\{ \begin{bmatrix} -1.5 \\ -0.3269 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -0.1884 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0 \\ 0.0779 \end{bmatrix} \alpha_1^2 + \right.$$
$$\left. \begin{bmatrix} 0 \\ 0.0077 \end{bmatrix} \beta_1 \, \middle| \, \alpha_1, \beta_1 \in [-1, 1] \right\}.$$

*The resulting set is visualized in Fig. 3.*

Next, we consider the case where the equality constraint is not symbolically solvable for one variable:

Proposition 2. *We consider a box $\mathcal{I}$, a guard set $\mathcal{G} = \langle g(\cdot) \rangle_G$, and the Taylor order $\kappa \in \mathbb{N}$. To tightly enclose the intersection $\mathcal{I} \cap \mathcal{G}$ we first compute the Taylor series expansion of $g(\cdot)$:*

$$a, b, c, v(\cdot), l, u \leftarrow \texttt{taylor}(g(\cdot), \mathcal{I}, \kappa). \tag{10}$$

*Next, we compute the intersection*

$$h(x) \leftarrow \texttt{intersect}(\mathcal{I}, \mathcal{LS}), \ \mathcal{LS} = \left\{ x \, \middle| \, x_{(k)} = p_k(x) \right\}$$

*of $\mathcal{I}$ and $\mathcal{LS}$ according to Thm. 1, where the function $p_k(x)$ results from splitting the polynomial function*

$$\begin{aligned}
&\frac{1}{a_{(k)}} \left( -b + a^T c - \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{(i)} x_{(i)} + v(x) \right) \\
&= p_k(x) + \widehat{p}(x), \text{ with } \frac{\partial p_k(x)}{\partial x_{(k)}} = 0
\end{aligned} \tag{11}$$

*into one part $\widehat{p}(x)$ containing the variable $x_{(k)}$ and one part $p_k(x)$ that does not contain the variable $x_{(k)}$. Afterward, we calculate the over-approximation*

$$[\widehat{l}, \widehat{u}] \supseteq \left\{ \widehat{p}(x) \, \middle| \, x \in \mathcal{I} \right\} \oplus \frac{-[l, u]}{a_{(k)}}. \tag{12}$$

*using interval arithmetic. Lastly, the intersection between $\mathcal{I}$ and $\mathcal{G}$ can be tightly enclosed by the Taylor model*

$$\mathcal{I} \cap \mathcal{G} \subseteq \left\{ \mathcal{T}(x) \,\middle|\, x \in \mathcal{I} \right\}$$
$$\text{with } \mathcal{T}(x) = \left\langle h(x), \left[ \begin{bmatrix} \mathbf{0}_{k-1} \\ \widehat{l} \\ \mathbf{0}_{n-k} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{k-1} \\ \widehat{u} \\ \mathbf{0}_{n-k} \end{bmatrix} \right], \mathcal{I} \right\rangle_{TM}.$$

Proof. Using the Taylor expansion of the function $g(x)$ in (10), the equality constraint $g(x) = 0$ can be represented as

$$x_{(k)} \in \frac{1}{a_{(k)}} \Big( -b + a^T c - \sum_{\substack{i=1 \\ i \neq k}}^{n} a_{(i)} x_{(i)} + v(x) \Big) \oplus \frac{-[l,u]}{a_{(k)}}$$
$$\overset{(11)}{=} p_k(x) + \widehat{p}(x) \oplus \frac{-[l,u]}{a_{(k)}} \overset{(12)}{\subseteq} p_k(x) \oplus [\widehat{l}, \widehat{u}], \tag{13}$$

so that the intersection between the box $\mathcal{I}$ and the guard set $\mathcal{G}$ can be formulated as

$$\mathcal{I} \cap \mathcal{G} = \left\{ x \in \mathcal{I} \,\middle|\, g(x) = 0 \right\} \overset{(13)}{\subseteq} \left\{ x \in \mathcal{I} \,\middle|\, x_{(k)} \in p_k(x) \oplus [\widehat{l}, \widehat{u}] \right\}.$$

The remainder of the proof is identical to the proof of Prop. 1 and therefore omitted. □

We demonstrate the computation of the intersection for the case where the equality constraint is not solvable for one variable with an example:

Example 4. *We consider the guard set*

$$\mathcal{G} = \left\{ x \in \mathbb{R}^2 \,\middle|\, 0.2\big(\sin(x_{(1)})x_{(2)} + \cos(x_{(2)})x_{(1)}\big) - x_{(2)} - 1 = 0 \right\}$$

*and the box* $\mathcal{I} = [-3, -2] \times [-1.3, -1]$. *Computation of a Taylor model enclosing the intersection according to Prop. 2 using a Taylor series of order* $\kappa = 2$ *yields*

$$\mathcal{T}(x) = \left\langle \begin{bmatrix} x_{(1)} \\ -0.9481 - 0.0496\, x_{(1)} - 0.0437\, x_{(1)}^2 \end{bmatrix}, \left[ \begin{bmatrix} 0 \\ -0.0072 \end{bmatrix}, \begin{bmatrix} 0 \\ 0.0087 \end{bmatrix} \right], \mathcal{I} \right\rangle_{TM},$$

*which can be equivalently represented by the polynomial zonotope*

$$\mathcal{PZ} = \left\{ \begin{bmatrix} -2.5 \\ -1.0964 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.0844 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0 \\ -0.0109 \end{bmatrix} \alpha_1^2 + \begin{bmatrix} 0 \\ 0.0079 \end{bmatrix} \beta_1 \,\middle|\, \alpha_1, \beta_1 \in [-1, 1] \right\}.$$

*The resulting sets are visualized in Fig. 4.*

In rare cases the computed Taylor model or polynomial zonotope becomes very large. This is due to the obtained over-approximation. To increase the robustness of our approach, we substitute each dimension of the calculated Taylor model $\mathcal{T}(x) = \langle h(x), \mathcal{B}, \mathcal{I} \rangle_{TM}$ for which the width of the box remainder $\mathcal{B}$ of the Taylor model is larger than the width of the box $\mathcal{I}$ with the box $\mathcal{I}$.

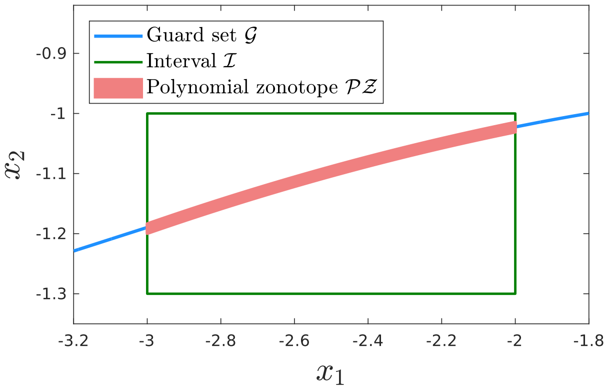

**Figure 4: Visualization of the computed intersection from Example 4.**

## 4.4 Reset Function and Unification

We first outline Step 5 of the procedure described in Sec. 3.2. In this work we consider nonlinear reset functions $r: \mathbb{R}^n \to \mathbb{R}^n$. Given the previously calculated Taylor model or polynomial zonotope $\mathcal{F}$ enclosing the intersection of the reachable set with the guard set, we first abstract the reset function with a Taylor series expansion of order $\gamma$ around the expansion point $c = \texttt{center}(\mathcal{F})$:

$$\forall x \in \mathcal{F}: \; r(x) \in \underbrace{\sum_{i=0}^{\gamma} \frac{\big((x-c)^T \nabla\big)^i \; r(\tilde{x})}{i!}\bigg|_{\tilde{x}=c}}_{p(x)} \oplus \mathcal{L},$$
$$\text{with } \mathcal{L} = \left\{ \frac{\big((x-c)^T \nabla\big)^{\gamma+1} \; r(\tilde{x})}{(\gamma+1)!}\bigg|_{\tilde{x} \in \mathcal{F}} \,\middle|\, x \in \mathcal{F} \right\}. \tag{14}$$

Next, we evaluate (14) to obtain a tight enclosure of the nonlinear map. For polynomial zonotopes the polynomial part $p(x)$ in (14) can be evaluated exactly since polynomial zonotopes are closed under quadratic and higher-order maps (see [30]). For Taylor models we compute a tight over-approximation instead. To enclose the Lagrange remainder $\mathcal{L}$ in (14), we first enclose $\mathcal{F}$ by a box, and then apply interval arithmetics to obtain an over-approximation. Methods to obtain tighter over-approximations of $\mathcal{L}$ exist (see [38]), but are computationally expensive.

Lastly, we consider the unification in Step 6 of the procedure described in Sec. 3.2. For polynomial zonotopes we unite the parallel sets with the convex hull according to [30, Prop. 13]. Since polynomial zonotopes are closed under convex hulls, the over-approximation from the unification is usually small. Taylor models are first converted to polynomial zonotopes using [30, Prop. 4], which enables the computation of the convex hull according to [30, Prop. 13]. The resulting polynomial zonotope can then be over-approximated by a Taylor model.

# 5 COMPUTATIONAL COMPLEXITY

To demonstrate the scalability of our approach, we derive the complexity with respect to the system dimension $n$. We make the assumption that the evaluation of a nonlinear function with interval arithmetics has complexity $\mathcal{O}(n)$.

The complexity for the computation of the reachable set for the continuous dynamics depends on the algorithm used. For linear as well as nonlinear continuous dynamics, algorithms with complexity equal to $\mathcal{O}(n^3)$ exist [3].

Next, we derive the complexity of the guard intersections step by step. The complexity of detecting the intersecting sets as described in Sec. 4.1 depends on the range bounding technique used. The number of nonlinear functions that have to be evaluated by range bounding does not depend on the system dimension. Therefore, the complexity for intersection detection using interval arithmetic for range bounding is $\mathcal{O}(n)$.

The complexity of the box contraction step in Sec. 4.2 depends on the contractor used. The simplest method is to use no contractor at all, which has complexity $\mathcal{O}(1)$. Furthermore, the computation of a unified box has complexity $\mathcal{O}(n)$.

If we consider an order of $\kappa = 2$, which is in most cases sufficient to achieve the required accuracy, computation of the Taylor series enclosure according to Def. 7 has complexity $\mathcal{O}(n^4)$ since it is required to evaluate $n^3$ nonlinear functions with interval arithmetic to compute an enclosure of the Lagrange remainder. The construction of the Taylor model representing $\mathcal{I} \cap \mathcal{G}$ according to Thm. 1 has complexity $\mathcal{O}(1)$. The complexity for the intersection computation step as described in Sec. 4.3 is therefore $\mathcal{O}(n^4)$.

Computing the mapping of a Taylor model or polynomial zonotope by the reset function according to (14) has complexity $\mathcal{O}(n^4)$ if an order of $\gamma = 2$ is used since the evaluation of the Lagrange remainder has complexity $\mathcal{O}(n^4)$, and the computation of the quadratic map for a polynomial zonotope has complexity $\mathcal{O}(n^4)$ (see [30, Prop. 11]). Furthermore, the unification of the parallel sets with the convex hull has complexity $\mathcal{O}(n^2)$ since this is the complexity of the convex hull operation for polynomial zonotopes (see [30, Prop. 13]).

In summary, the overall complexity for the computation of the reachable set using our presented approach is therefore $\mathcal{O}(n^4)$ with respect to the system dimension $n$ if a suitable algorithm for the computation of the continuous reachable set, a suitable range bounding technique, and a suitable contractor is used.

# 6 NUMERICAL EXAMPLES

In this section we demonstrate the performance of our novel method on several benchmark systems. All computations are carried out in MATLAB on a 2.9GHz quad-core i7 processor with 32GB memory. We use polynomial zonotopes for the representation of reachable sets. For the computation of the continuous reachable set we use the algorithm in [3]. We apply interval arithmetic for intersection detection and use our own implementation of the *HC4revise* algorithm from [34] as a contractor. Furthermore, we use a Taylor order of $\kappa = 2$ for computation of the guard intersection and a Taylor order of $\gamma = 1$ for computation of the reset mapping. Unless otherwise explicitly stated, we do not use an upper bound $\mu$ for the maximum number of boxes that are united. The implementation of our approach will be made publicly available with the next release of the CORA toolbox [2].

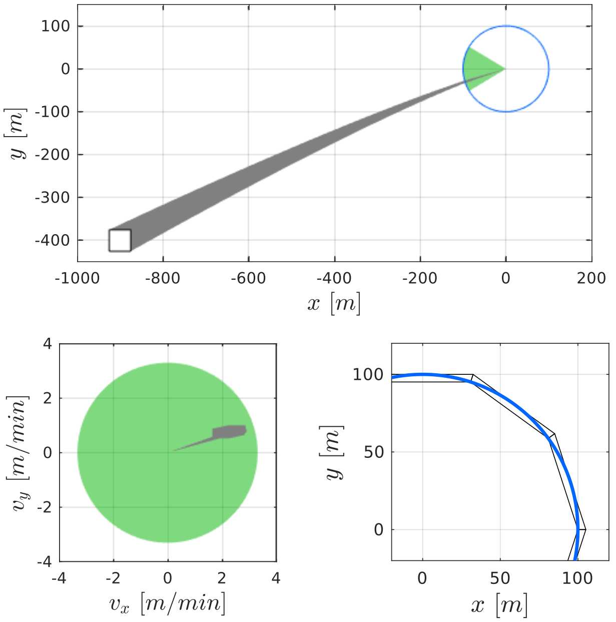

**Figure 5: Reachable set (top), reachable set for mode *rendezvous attempt* (bottom, left), and polytope enclosure of the guard set (bottom, right) for the spacecraft rendezvous benchmark.**

## 6.1 Spacecraft Rendezvous

First, we consider the spacecraft rendezvous benchmark described in [14], which is part of the ARCH 2019 competition [28]. The benchmark examines the maneuver of a spacecraft docking to a space station. The four system states are the planar positions $x, y$ and corresponding velocities $v_x, v_y$ of the spacecraft. There are two discrete modes, *approaching* and *rendezvous attempt*. The system starts in mode *approaching* from the initial set $x \in [-925, -875]$, $y \in [-425, -375]$, $v_x = 0$, and $v_y = 0$. If the spacecraft is at a distance of $100m$ from the space station, the system transitions into mode *rendezvous attempt* where a different controller is applied. This transition is modeled by the guard set

$$\mathcal{G} = \left\{ [x\ y\ v_x\ v_y]^T \in \mathbb{R}^4 \;\middle|\; x^2 + y^2 - 100^2 = 0 \right\}. \qquad (15)$$

The continuous dynamics for both discrete modes is nonlinear (see [28]). The specifications for the benchmark are that in mode *rendezvous attempt* the spacecraft stays inside the line-of-sight cone $C$ defined as

$$C = \left\{ [x\ y\ v_x\ v_y]^T \in \mathbb{R}^4 \;\middle|\; (x \geq -100) \wedge (y \geq x \tan(30^\circ)) \right. \\ \left. \wedge\, (-y \geq x\ \tan(30^\circ)) \right\}$$

and the absolute velocity stays below 3.3:

$$\sqrt{v_x^2 + v_y^2} \leq 3.3.$$

The considered time horizon is $t_f = 200$ min.

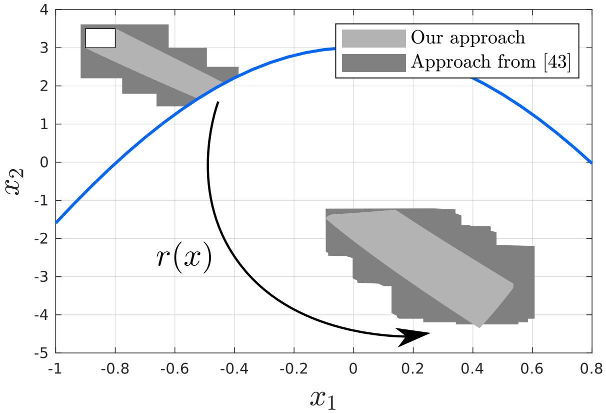

**Figure 6: Reachable set for the artificial hybrid system.**

We compare our method to the approach in [4], which is implemented in the CORA toolbox [2]. To handle discrete transitions, this approach first computes the intersection geometrically, and then encloses the union of all partial intersections with bundles of parallelotopes. Since [4] is only applicable for guard sets represented by polytopes or hyperplanes, we have to enclose the nonlinear guard set in (15) with multiple polytopes (see Fig. 5, bottom right).

Since the equality constraint that defines the guard set in (15) is not symbolically solvable for one variable we calculate the intersection with the guard set according to Prop. 2. Fig. 5 visualizes the results from our approach. The reachable sets computed with our approach and the approach in [4] both have similar precision, and both satisfy the specifications. However, computing the guard intersection with the approach in [4] takes 4.96 seconds, whereas the computation of the guard intersection with our approach takes only 0.93 seconds. Furthermore, for high-dimensional systems the computation time for enclosing nonlinear guards by polytopes might already be very large.

## 6.2 Artificial Hybrid System

Next, we consider the 2-dimensional artificial hybrid system from [43, Sec. 6.1]. The system has one nonlinear guard set

$$\mathcal{G} = \left\{ [x_1 \; x_2]^T \in \mathbb{R}^2 \;\middle|\; \cos(x_1) - 0.1 x_2 - 0.7 = 0 \right\} \tag{16}$$

and an uncertain reset function

$$r(x) = \begin{bmatrix} -x_1 \\ \nu \, x_2 \end{bmatrix}, \;\; \nu \in [-2.05, -2],$$

where $x = [x_1 \; x_2]^T$.

We compare our novel method with the approach from [43]. Since the equality constraint that defines the guard set in (16) is symbolically solvable for one variable, we calculate the intersection with the guard set according to Prop. 1. The visualization of the reachable set in Fig. 6 shows that the reachable set computed with our method is much tighter. In addition, the computation time for our method is only 0.87 seconds, while the computation time for the approach in [43] is 26 seconds on their machine.

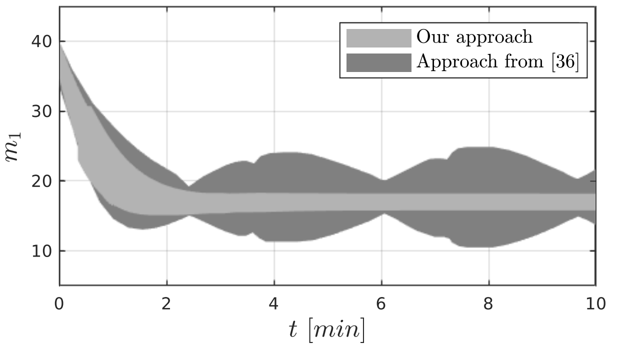

**Figure 7: Reachable set for the transcriptional regulator network with n = 12 states.**

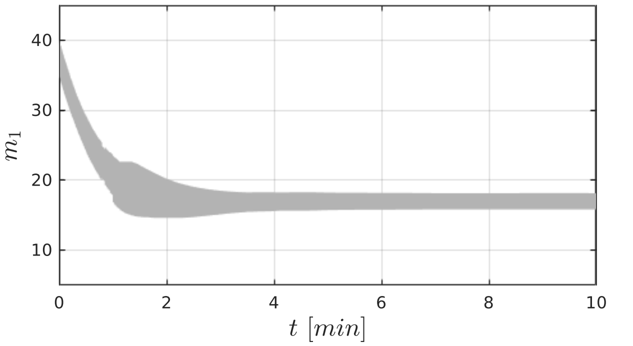

**Figure 8: Reachable set for the transcriptional regulator network with n = 24 states calculated with our approach. The computation of the reachable set using the approach in [36] resulted in a memory overflow.**

## 6.3 Transcriptional Regulator Network

To demonstrate the scalability of our approach, we consider the benchmark in [36, Sec. 8.D] describing a transcriptional regulator network with $N$ genes. For a network with $N$ genes the system has $n = 2N$ dimensions. The benchmark has one nonlinear guard set and two modes with nonlinear continuous dynamics. The parameters and equations for this benchmark are listed in [36].

We consider the cases with $N = 6$ and $N = 12$ genes, which correspond to $n = 12$ and $n = 24$ system states, respectively. For the calculation of the reachable set with our novel approach, we use an upper bound of $\mu = 5$ for the case with $N = 6$ genes as well as an upper bound $\mu = 7$ for the case with $N = 12$ genes. Since the equality constraint that defines the guard set is not symbolically solvable for one variable we calculate the intersection with the guard set according to Prop. 2.

We compare our method with the approach from [36]. The visualization of the results in Fig. 7 shows that the reachable set computed with our approach is much tighter for the case with $N = 6$ genes. In addition, the computation time for our approach is only 9.2 seconds, while the computation time for the approach in [36] is 130 seconds on their machine (see [36, Tab. 4]). For the case with $N = 12$ genes the approach from [36] is not applicable due to a memory overflow (see [36, Tab. 4]). However, with our novel approach we can calculate the reachable set in only 35.6 seconds (see Fig. 8).

## 7 CONCLUSION

In this paper, we introduced a novel method for the calculation of intersections with nonlinear guard sets in hybrid system reachability analysis. In contrast to other approaches, our novel method is applicable to high-dimensional systems because the computational complexity grows only polynomially with respect to the system dimension. Furthermore, the modular design allows us to combine our guard intersection method with different algorithms for continuous reachability analysis, which makes it applicable to a very broad class of hybrid systems. The evaluation of our novel method on numerical examples demonstrated its scalability and superior performance compared to other approaches.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the German Research Foundation (DFG) project faveAC under grant number AL 1185/5-1.

## REFERENCES

[1] M. Althoff. 2013. Reachability Analysis of Nonlinear Systems using Conservative Polynomialization and Non-Convex Sets. In *Hybrid Systems: Computation and Control.* 173–182.

[2] M. Althoff. 2015. An Introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems.* 120–151.

[3] M. Althoff and et al. 2008. Reachability Analysis of Nonlinear Systems with Uncertain Parameters using Conservative Linearization. In *Proc. of the 47th IEEE Conference on Decision and Control.* 4042–4048.

[4] M. Althoff and B. H. Krogh. 2011. Zonotope Bundles for the Efficient Computation of Reachable Sets. In *Proc. of the 50th IEEE Conference on Decision and Control.* 6814–6821.

[5] M. Althoff and B. H. Krogh. 2012. Avoiding Geometric Intersection Operations in Reachability Analysis of Hybrid Systems. In *Hybrid Systems: Computation and Control.* 45–54.

[6] M. Althoff and B. H. Krogh. 2014. Reachability Analysis of Nonlinear Differential-Algebraic Systems. *IEEE Trans. Automat. Control* 59, 2 (2014), 371–383.

[7] E. Asarin and et al. 2007. Hybridization Methods for the Analysis of Nonlinear Systems. *Acta Informatica* 43 (2007), 451–476.

[8] Stanley Bak, Sergiy Bogomolov, and Matthias Althoff. 2017. Time-triggered conversion of guards for reachability analysis of hybrid automata. In *International Conference on Formal Modeling and Analysis of Timed Systems.* Springer, 133–150.

[9] S. Bak and P. S. Duggirala. 2017. HyLAA: A Tool for Computing Simulation-Equivalent Reachability for Linear Systems. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control.* 173–178.

[10] S. Bak and P. S. Duggirala. 2017. Simulation-Equivalent Reachability of Large Linear Systems with Inputs. In *Proc. of International Conference on Computer Aided Verification.* 401–420.

[11] L. Benvenuti and et al. 2014. Assume-guarantee verification of nonlinear hybrid systems with ARIADNE. *International Journal of Robust and Nonlinear Control* 24 (2014), 699–724.

[12] Sergiy Bogomolov and et al. 2019. JuliaReach: a toolbox for set-based reachability. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control.* ACM, 39–44.

[13] G. Chabert and L. Jaulin. 2009. Contractor programming. *Artificial Intelligence* 173, 11 (2009), 1079–1100.

[14] N. Chan and S. Mitra. 2017. Verifying safety of an autonomous spacecraft rendezvous mission. In *Proc. of the 4th International Workshop on Applied Verification of Continuous and Hybrid Systems.* 20–32.

[15] X. Chen and et al. 2012. Taylor Model Flowpipe Construction for Non-linear Hybrid Systems. In *Proc. of the 33rd IEEE Real-Time Systems Symposium.*

[16] X. Chen and et al. 2013. Flow*: An Analyzer for Non-Linear Hybrid Systems. In *Proc. of Computer-Aided Verification (LNCS 8044).* Springer, 258–263.

[17] A. Chutinan and B. H. Krogh. 2003. Computational Techniques for Hybrid System Verification. *IEEE Trans. Automat. Control* 48, 1 (2003), 64–75.

[18] P. Collins, E. Bresolin, L. Geretti, and T. Villa. 2012. Computing the evolution of hybrid systems using rigorous function calculus. *IFAC Proceedings Volumes* 45, 9 (2012), 284–290.

[19] T. Dang and et al. 2010. Accurate Hybridization of Nonlinear Systems. In *Hybrid Systems: Computation and Control.* 11–19.

[20] P. S. Duggirala and M. Viswanathan. 2016. Parsimonious, Simulation Based Verification of Linear Systems. In *Proc. of International Conference on Computer Aided Verification.* 477–494.

[21] G. Frehse and et al. 2011. SpaceEx: Scalable Verification of Hybrid Systems. In *Proc. of the 23rd International Conference on Computer Aided Verification (LNCS 6806).* Springer, 379–395.

[22] G. Frehse and R. Ray. 2012. Flowpipe-Guard Intersection for Reachability Computations with Support Functions. In *Proc. of the 4th IFAC Conference on Analysis and Design of Hybrid Systems.* 94–101.

[23] A. Girard and et al. 2006. Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs. In *Hybrid Systems: Computation and Control (LNCS 3927).* Springer, 257–271.

[24] A. Girard and C. Le Guernic. 2008. Efficient Reachability Analysis for Linear Systems using Support Functions. In *Proc. of the 17th IFAC World Congress.* 8966–8971.

[25] A. Girard and C. Le Guernic. 2008. Zonotope/Hyperplane Intersection for Hybrid Systems Reachability Analysis. In *Proc. of Hybrid Systems: Computation and Control (LNCS 4981).* Springer, 215–228.

[26] P. Van Hentenryck, L. Michel, and Y. Deville. 1997. *Numerica: a modeling language for global optimization.* MIT press.

[27] F. Immler. 2015. Tool Presentation: Isabelle/HOL for Reachability Analysis of Continuous Systems. In *Proc. of the 2nd Workshop on Applied Verification for Continuous and Hybrid Systems.* 180–187.

[28] F. Immler and et al. 2019. ARCH-COMP19 Category Report: Continuous and Hybrid Systems with Nonlinear Dynamics. In *ARCH19. 6th International Workshop on Applied Verification of Continuous and Hybrid Systems.* 41–61.

[29] L. Jaulin, M. Kieffer, and O. Didrit. 2006. *Applied Interval Analysis.* Springer.

[30] N. Kochdumper and M. Althoff. 2019. Sparse Polynomial Zonotopes: A Novel Set Representation for Reachability Analysis. *arXiv preprint arXiv:1901.01780* (2019).

[31] H. Kong and et al. 2017. Safety verification of nonlinear hybrid systems based on invariant clusters. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control.* 163–172.

[32] A. B. Kurzhanski and P. Varaiya. 2000. Ellipsoidal Techniques for Reachability Analysis. In *Hybrid Systems: Computation and Control (LNCS 1790).* Springer, 202–214.

[33] G. Lafferriere, G. J. Pappas, and S. Yovine. 1999. A new Class of decidable Hybrid Systems. In *Hybrid Systems: Computation and Control (LNCS 1569).* Springer, 137–151.

[34] G. Laurent and et al. 1999. Revising hull and box consistency. In *Proc. of ICLP, The MIT Press.* 230–244.

[35] J. Liu and et al. 2011. Computing semi-algebraic invariants for polynomial dynamical systems. In *Proc. of the 9th ACM international conference on Embedded software.* 97–106.

[36] M. Maïga, N. Ramdani, L. Travé-Massuyè, and C. Combastel. 2015. A comprehensive method for reachability analysis of uncertain nonlinear hybrid systems. *IEEE Trans. Automat. Control* 61, 9 (2015), 2341–2356.

[37] M. Maïga, N. Ramdani, and L. Travé-Massuyès. 2013. A fast method for solving guard set intersection in nonlinear hybrid reachability. In *52nd IEEE Conference on Decision and Control.* IEEE, 508–513.

[38] K. Makino and M. Berz. 2003. Taylor Models and Other Validated Functional Inclusion Methods. *International Journal of Pure and Applied Mathematics* 4, 4 (2003), 379–456.

[39] N. Matringe and et al. 2010. Generating invariants for non-linear hybrid systems by linear algebraic methods. In *International Static Analysis Symposium.* Springer, 373–389.

[40] I. M. Mitchell and et al. 2005. A Time-Dependent Hamilton–Jacobi Formulation of Reachable Sets for Continuous Dynamic Games. *IEEE Trans. Automat. Control* 50, 7 (2005), 947–957.

[41] N. S. Nedialkov and M. von Mohrenschildt. 2002. Rigorous Simulation of Hybrid Dynamic Systems with Symbolic and Interval Methods. In *Proc. of the American Control Conference.* 140–147.

[42] P. Prabhakar and M. Viswanathan. 2011. A Dynamic Algorithm for Approximate Flow Computations. In *Hybrid Systems: Computation and Control.* 133–142.

[43] N. Ramdani and N. S. Nedialkov. 2009. Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems Using Interval Constraint Propagation Techniques. In *Proc. of the 3rd IFAC Conference on Analysis and Design of Hybrid Systems.* 156–161.

[44] Stefan Schupp and et al. 2017. HyPro: A C++ library of state set representations for hybrid systems reachability analysis. In *NASA Formal Methods Symposium.* Springer, 288–294.

[45] O. Stursberg and B. H. Krogh. 2003. Efficient Representation and Computation of Reachable Sets for Hybrid Systems. In *Hybrid Systems: Computation and Control (LNCS 2623).* Springer, 482–497.

[46] G. Trombettoni, Y. Papegay, G. Chabert, and O. Pourtallier. 2010. A box-consistency contractor based on extremal functions. In *International Conference on Principles and Practice of Constraint Programming.* Springer, 491–498.