Computing Non-Convex Inner-Approximations of Reachable Sets for Nonlinear Continuous Systems

Niklas Kochdumper and Matthias Althoff

***Abstract*— We present a novel approach to compute non-convex inner-approximations of reachable sets for nonlinear continuous systems. The concept of our approach is to extract inner-approximations of reachable sets from pre-computed outer-approximations, which makes our method computationally very efficient as we demonstrate with several numerical examples. Since our approach has polynomial complexity with respect to the system dimension, it is well-suited for high-dimensional systems.**

I. INTRODUCTION

Formal verification in control theory often involves reach-avoid problems, where the controlled system has to reach a goal set while avoiding a set of unsafe states. One common method to verify that a given controller satisfies the specifications of the reach-avoid problem is reachability analysis. To prove that the system does not enter the set of unsafe states requires an outer-approximation of the reachable set. On the other hand, proving that the system reaches the goal set requires an inner-approximation of the reachable set. In this paper we introduce a novel approach to efficiently extract inner-approximations of reachable sets from pre-computed outer-approximations.

A. State of the Art

The computation of outer-approximations of reachable sets is a well-studied problem for which many approaches for various system classes exist, such as linear continuous systems [4], [6], [12], [22], nonlinear continuous systems [1], [9], [21], [24] and hybrid systems [3], [5], [11].

The problem of computing inner-approximations of reachable sets has been studied far less. Efficient techniques for computing inner-approximations for linear continuous systems exist for quite some time: The approach in [12] uses zonotopes to compute inner-approximations for linear time-invariant systems with piecewise constant inputs. For time-varying linear systems the method in [22] computes inner-approximations represented by ellipsoids. The authors in [16] compute inner-approximations for piecewise-affine systems using linear matrix inequalities.

Only recently, methods for computing inner-approximations for nonlinear continuous systems have been developed: In [13] a criterion on when a box is part of the inner-approximation is provided. Based on this an inner-approximation of the reachable set represented by a union of boxes can be computed. However, this method is computationally expensive for high-dimensional systems due to the curse of dimensionality.

Several approaches compute inner-approximations represented by sub-zero level-sets based on the Hamilton-Jacobi framework [24]: The approach in [23] computes inner-approximations by under-approximating the evolution function of the system. In [29] and [30] inner-approximations for polynomial systems are obtained by solving semi-definite programs. Since the size of the semi-definite program grows rapidly with the system dimension the approaches in [29], [30] are computational expensive for high-dimensional systems.

Other approaches use the time-inverted dynamics to compute inner-approximations: The method in [31] computes inner-approximations represented by polytopes using linear programming. First, a set which encloses the boundary is computed. Afterward, a polytope outer-approximation of the reachable set is contracted until it is inclosed by the set enclosing the boundary. In [28], this approach is extended to handle delay differential equations. Since reachable sets of nonlinear systems are in general non-convex, inner-approximations represented by convex polytopes as computed in [28], [31] are often not very accurate. The approach in [10] first computes a backward-flowpipe based on Picard iteration. This flowpipe is then used to propagate the inequality constraints that define the initial set forward in time, which yields a non-convex inner-approximation represented by the intersection of polynomial inequality constraints. However, the intersection of polynomial inequality constrains can result in sets that are not connected. To determine a valid inner-approximation, the method in [10] therefore requires to prove that the resulting set is connected, which is computationally expensive.

The approach in [14] demonstrated that inner-approximations for the projection of the reachable set onto the coordinate axes can be computed very efficiently for autonomous nonlinear systems. In [15], the authors extended this approach to systems with uncertain inputs. While inner-approximations of the projection are often useful for simple verification tasks, they are in general not sufficient to answer more complex verification queries.

B. Contribution

We present a novel approach to compute inner-approximations of reachable sets for nonlinear systems. Our approach has only polynomial complexity with respect to the system dimension and is therefore well-suited for the formal analysis of high-dimensional systems. Since we extract

N. Kochdumper and M. Althoff are both with the Faculty of Computer Science, Technical University of Munich, Garching, Germany niklas.kochdumper@tum.de, althoff@tum.de

inner-approximations of reachable sets from pre-computed outer-approximations our approach is computationally very efficient compared to previous methods as we demonstrate with numerical examples in Sec. V.

C. Notation

Sets are denoted by calligraphic letters, matrices by uppercase letters, vectors by lowercase 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 $[C\ D]$, and the symbols $\mathbf{0}$ and $\mathbf{1}$ denote vectors of zeros and ones with proper dimension. Given a set $\mathcal{S} \subset \mathbb{R}^n$, $\partial\mathcal{S}$ denotes the boundary of $\mathcal{S}$. The set difference for two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^n$ is defined as $\mathcal{S}_1 \setminus \mathcal{S}_2 := \{s \mid s \in \mathcal{S}_1 \wedge s \notin \mathcal{S}_2\}$, 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 $\mathcal{S}_1 \times \mathcal{S}_2 := \left\{[s_1\ s_2]^T \;\middle|\; s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\right\}$. Given a set $\mathcal{S} \subset \mathbb{R}^n$, the operation $\texttt{project}(\mathcal{S}, d) := \{[s_{(d_{(1)})} \dots s_{(d_{(m)})}]^T \mid s \in \mathcal{S}\}$ projects the set $\mathcal{S}$ onto the dimensions given by the vector $d \in \mathbb{N}^m_{\leq n}$. 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.

II. PROBLEM STATEMENT

In this paper we consider autonomous nonlinear systems

$$\dot{x}(t) = f(x(t)), \quad x(t) \in \mathbb{R}^n, \tag{1}$$

where $x(t)$ is the state vector and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a Lipschitz continuous function. The reachable set of the system is defined as follows:

Definition 1: *(Reachable Set) Let $\xi(t, x_0)$ denote the solution to (1) at time t for the initial state $x(0) = x_0$. The reachable set for an initial set $\mathcal{X}_0 \subset \mathbb{R}^n$ is*

$$\mathcal{R}_{\mathcal{X}_0}(t) := \left\{\xi(t, x_0) \;\middle|\; x_0 \in \mathcal{X}_0\right\}.$$

The exact reachable set as defined in Def. 1 cannot be computed for general nonlinear systems. Therefore, the goal of reachability analysis is to compute tight outer-approximations $\mathcal{R}^o_{\mathcal{X}_0}(t) \supseteq \mathcal{R}_{\mathcal{X}_0}(t)$ and inner-approximations $\mathcal{R}^i_{\mathcal{X}_0}(t) \subseteq \mathcal{R}_{\mathcal{X}_0}(t)$.

Since the exact reachable set of a nonlinear system is in general non-convex, tight inner-approximations and outer-approximations can only be computed with a non-convex set representation. We use the sparse representation of polynomial zonotopes introduced in [19] to represent $\mathcal{R}^i_{\mathcal{X}_0}(t)$ and $\mathcal{R}^o_{\mathcal{X}_0}(t)$:

Definition 2: *(Polynomial Zonotope) Given a generator matrix $G \in \mathbb{R}^{n \times h}$ and an exponent matrix $E \in \mathbb{N}_0^{p \times h}$, a polynomial zonotope is defined as*

$$\mathcal{PZ} := \left\{ \sum_{i=1}^{h} \left(\prod_{k=1}^{p} \alpha_{(k)}^{E_{(k,i)}} \right) G_{(\cdot,i)} \;\middle|\; \alpha \in [-\mathbf{1}, \mathbf{1}] \right\}.$$

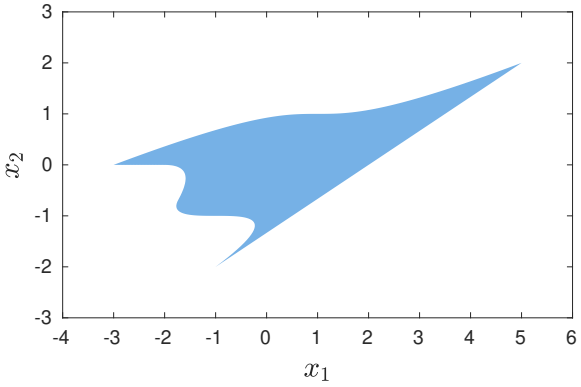

Fig. 1: Polynomial zonotope from Example 1.

The entries $\alpha_{(k)}$ of the vector $\alpha \in \mathbb{R}^p$ are called factors of the polynomial zonotope. We use the shorthand $\mathcal{PZ} = \langle G, E \rangle_{PZ}$. The following example demonstrates the concept of polynomial zonotopes:

Example 1: *The polynomial zonotope*

$$\mathcal{PZ} = \left\langle \begin{bmatrix} 2 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \end{bmatrix} \right\rangle_{PZ}$$

defines the set

$$\mathcal{PZ} = \left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix} \alpha_1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \alpha_2 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \alpha_1^3 \alpha_2 \;\middle|\; \alpha_1, \alpha_2 \in [-1, 1] \right\},$$

which is visualized in Fig. 1.

III. NON-CONVEX INNER-APPROXIMATIONS

Our method for computing an inner-approximation of the reachable set is based on the following theorem:

Theorem 1: *[10, Sec. 3] Given a set $\mathcal{B} \supseteq \partial\mathcal{R}_{\mathcal{X}_0}(t)$ that encloses the boundary of the exact reachable set, every connected set $\mathcal{C}$ that does not intersect $\mathcal{B}$ and contains some state of the exact reachable set $\mathcal{R}_{\mathcal{X}_0}(t)$, is an inner-approximation of the reachable set:*

$$\forall \mathcal{C} \subset \mathbb{R}^n : \ \left(\mathcal{C} \cap \mathcal{B} = \emptyset\right) \wedge \left(\mathcal{C} \cap \mathcal{R}_{\mathcal{X}_0}(t) \neq \emptyset\right) \Rightarrow \mathcal{C} \subseteq \mathcal{R}_{\mathcal{X}_0}(t).$$

Next, we present the basic steps of the procedure that we apply to compute a set $\mathcal{C}$ that satisfies Theorem 1.

A. Basic Procedure

To compute an inner-approximation $\mathcal{R}^i_{\mathcal{X}_0}(t)$ of the reachable set at time t, we follow the steps visualized in Fig. 2:

1. We first compute an outer-approximation $\mathcal{R}^o_{\mathcal{X}_0}(t)$ of the reachable set with the reachability algorithm from [1] using polynomial zonotopes to represent the outer-approximation of the reachable set.
2. Next, we compute a set $\mathcal{B}$ that encloses the boundary $\partial\mathcal{R}_{\mathcal{X}_0}(t)$ of the exact reachable set.
3. Afterwards, we scale the size of the set $\mathcal{R}^o_{\mathcal{X}_0}(t)$ to obtain a set $\mathcal{C}$ that does not intersect the set $\mathcal{B}$.

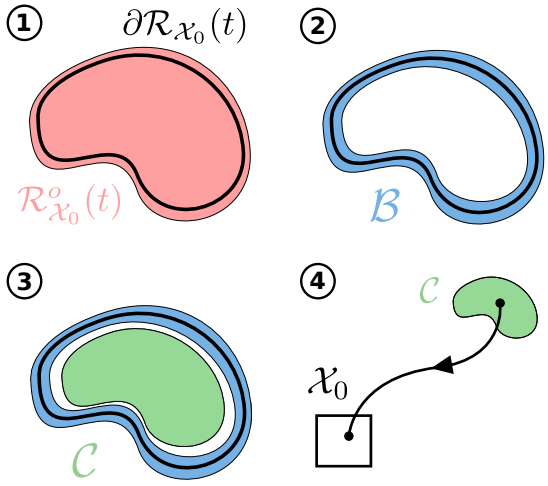

Fig. 2: Visualization of the procedure applied to calculate an inner-approximation of the reachable set.

(4) Finally, we simulate the backward flow of the system using the center of $\mathcal{C}$ as a starting point to verify that $\mathcal{C}$ satisfies the conditions from Theorem 1, which then proves that $\mathcal{C}$ is an inner-approximation $\mathcal{R}^i_{\mathcal{X}_0}(t)$ of the reachable set at time t.

This procedure for computing inner-approximations is inspired by [31]. Next, we describe the steps 2 3, and 4 in detail.

B. Enclosure of the Boundary

To compute a set which encloses the boundary of the exact reachable set, we require the following well-known theorem:

Theorem 2: *[27, Corollary 1] Given a non-empty compact initial set $\mathcal{X}_0 \subset \mathbb{R}^n$, it holds that*

$$\mathcal{R}_{\partial\mathcal{X}_0}(t) \equiv \partial\mathcal{R}_{\mathcal{X}_0}(t)$$

In Step 1 of the procedure described in Sec. III-A, we apply the approach from [1] to compute a polynomial zonotope $\mathcal{R}^o_{\mathcal{X}_0}(t) = \langle G, E\rangle_{PZ}$ that is an outer-approximation of the exact reachable set. It is shown in [20] that for all reachable sets computed with the algorithm in [1], an outer-approximation of the reachable set $\mathcal{R}^o_{\widehat{\mathcal{X}}_0}(t)$ for a subset $\widehat{\mathcal{X}}_0 \subseteq \mathcal{X}_0$ of the initial set can be extracted directly from the reachable set $\mathcal{R}^o_{\mathcal{X}_0}(t)$, which we denote by the operator `reachSubset`:

$$\mathcal{R}^o_{\widehat{\mathcal{X}}_0}(t) = \texttt{reachSubset}\big(\widehat{\mathcal{X}}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big), \tag{2}$$

where $\mathcal{R}^o_{\widehat{\mathcal{X}}_0}(t)$ is represented by a polynomial zonotope. We demonstrate the extraction of reachable subsets by an example:

Example 2: *Let us consider the nonlinear system*

$$\begin{aligned}\dot{x}_1 &= \frac{x_2}{2} + 5\\ \dot{x}_2 &= \frac{x_1}{200}\big(100 - x_1(10 + x_2)\big) + 5\end{aligned} \tag{3}$$

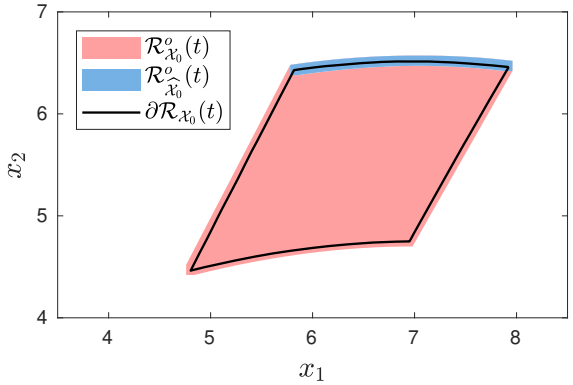

Fig. 3: Visualization of the results from Example 2.

and the initial set

$$\mathcal{X}_0 = \big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1, \alpha_2 \in [-1, 1]\big\}. \tag{4}$$

Computation of an outer-approximation for the reachable set at $t = 1s$ with the approach from [1] yields

$$\begin{aligned}\mathcal{R}^o_{\mathcal{X}_0}(t) = \Bigg\{ &\begin{bmatrix}6.39\\5.6\end{bmatrix} + \begin{bmatrix}1.06\\0.08\end{bmatrix}\alpha_1 + \begin{bmatrix}0.50\\0.92\end{bmatrix}\alpha_2 - \begin{bmatrix}0.02\\0.07\end{bmatrix}\alpha_1^2\\ &- \begin{bmatrix}0.01\\0.06\end{bmatrix}\alpha_1\alpha_2 + \begin{bmatrix}0.05\\0\end{bmatrix}\alpha_3 + \begin{bmatrix}0\\0.04\end{bmatrix}\alpha_4\\ &\Bigg| \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in [-1, 1]\Bigg\}.\end{aligned}$$

Let us now consider the set

$$\widehat{\mathcal{X}}_0 = \big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1 \in [-1, 1],\ \alpha_2 = 1\big\}, \tag{5}$$

which is a subset of the initial set $\widehat{\mathcal{X}}_0 \subset \mathcal{X}_0$ (see (4)). We apply the approach in [20] by replacing $\alpha_2 \in [-1, 1]$ in (??) by $\alpha_2 = 1$ (see (5)) to obtain

$$\mathcal{R}^o_{\widehat{\mathcal{X}}_0}(t) = \texttt{reachSubset}\big(\widehat{\mathcal{X}}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big) \overset{[21]}{=}$$

$$\begin{aligned}\Bigg\{ &\begin{bmatrix}6.39\\5.6\end{bmatrix} + \begin{bmatrix}1.06\\0.08\end{bmatrix}\alpha_1 + \begin{bmatrix}0.50\\0.92\end{bmatrix}\alpha_2 - \begin{bmatrix}0.02\\0.07\end{bmatrix}\alpha_1^2 - \begin{bmatrix}0.01\\0.06\end{bmatrix}\alpha_1\alpha_2\\ &+ \begin{bmatrix}0.05\\0\end{bmatrix}\alpha_3 + \begin{bmatrix}0\\0.04\end{bmatrix}\alpha_4 \Bigg| \alpha_1, \alpha_3, \alpha_4 \in [-1, 1],\ \alpha_2 = 1\Bigg\} =\end{aligned}$$

$$\begin{aligned}\Bigg\{ &\begin{bmatrix}6.89\\6.52\end{bmatrix} + \begin{bmatrix}1.05\\0.02\end{bmatrix}\alpha_1 - \begin{bmatrix}0.02\\0.07\end{bmatrix}\alpha_1^2 + \begin{bmatrix}0.05\\0\end{bmatrix}\alpha_3\\ &+ \begin{bmatrix}0\\0.04\end{bmatrix}\alpha_4 \Bigg| \alpha_1, \alpha_3, \alpha_4 \in [-1, 1]\Bigg\}.\end{aligned}$$

The extracted subset $\mathcal{R}^o_{\widehat{\mathcal{X}}_0}(t)$ is shown in Fig. 3.

The operator `reachSubset` as defined in (2) can be used to compute a set $\mathcal{B}$ that encloses the boundary of the exact reachable set:

Proposition 1: *Given a polynomial zonotope $\mathcal{R}^o_{\mathcal{X}_0}(t)$ that represents an outer-approximation of the reachable set, the set*

$$\mathcal{B} = \texttt{reachSubset}\big(\partial\mathcal{X}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big)$$

enclsoses the boundary of the exact reachable set $\partial\mathcal{R}_{\mathcal{X}_0}(t) \subseteq \mathcal{B}$.

Proof. According to [20, Theorem 1] it holds for every subset $\widehat{\mathcal{X}}_0 \subseteq \mathcal{X}_0$ that

$$\mathcal{R}_{\widehat{\mathcal{X}}_0}(t) \subseteq \texttt{reachSubset}\big(\widehat{\mathcal{X}}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big). \tag{6}$$

Since $\partial\mathcal{X}_0 \subseteq \mathcal{X}_0$, we therefore have

$$\begin{aligned}\partial\mathcal{R}_{\mathcal{X}_0}(t) &\overset{\text{Thm. 2}}{=} \mathcal{R}_{\partial\mathcal{X}_0}(t)\\ &\overset{(6)}{\subseteq} \texttt{reachSubset}\big(\partial\mathcal{X}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big) = \mathcal{B}\end{aligned}$$

□

For simplicity, we consider for the remainder of this paper that the initial set is given by a box $\mathcal{X}_0 = [l, u] \subset \mathbb{R}^n$. Then the boundary of the initial set is given as

$$\begin{aligned}&\partial\mathcal{X}_0 = \partial[l,u] = \bigcup_{i=1}^{n}\bigcup_{j=1}^{2}\mathcal{I}_{i,j}\\ &\text{with}\quad \mathcal{I}_{i,j} = \begin{cases}\{x \in [l,u] \mid x_{(i)} = u_{(i)}\}, & \text{for } j = 1\\ \{x \in [l,u] \mid x_{(i)} = l_{(i)}\}, & \text{otherwise}\end{cases}\end{aligned} \tag{7}$$

If we apply Prop. 1 to (7), we obtain

$$\begin{aligned}&\mathcal{B} = \texttt{reachSubset}\big(\partial\mathcal{X}_0, \mathcal{R}^o_{\mathcal{X}_0}(t)\big) =\\ &\bigcup_{i=1}^{n}\bigcup_{j=1}^{2}\underbrace{\texttt{reachSubset}\big(\mathcal{I}_{i,j}, \mathcal{R}^o_{\mathcal{X}_0}(t)\big)}_{\mathcal{PZ}_{2(i-1)+j}} = \bigcup_{k=1}^{2n}\mathcal{PZ}_k.\end{aligned} \tag{8}$$

The boundary of the exact reachable set can therefore be enclosed by the union of $2n$ polynomial zonotopes $\mathcal{PZ}_k$. We demonstrate the computation of the set $\mathcal{B}$ by an example:

Example 3: *For our running example in* (3) *with the initial set* $\mathcal{X}_0 = [-1,1]\times[-1,1]$ *(see* (4)*) the boundary of the initial set is*

$$\begin{aligned}\partial\mathcal{X}_0 =&\big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1 = 1,\ \alpha_2 \in [-1,1]\big\}\cup\\ &\big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1 = -1,\ \alpha_2 \in [-1,1]\big\}\cup\\ &\big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1 \in [-1,1],\ \alpha_2 = 1\big\}\cup\\ &\big\{[\alpha_1\ \alpha_2]^T \mid \alpha_1 \in [-1,1],\ \alpha_2 = -1\big\}.\end{aligned} \tag{9}$$

The reachable set for each of the partial sets in (9) *can be computed efficiently with the operator* `reachSubset` *as we demonstrated in Example 2. The resulting set* $\mathcal{B}$ *that encloses the boundary of the exact reachable set is shown in Fig. 4.*

C. Computation of the Inner-Approximation

To find a suitable set $\mathcal{C}$ that satisfies Theorem 1, we scale the previously computed outer-approximation $\mathcal{R}^o_{\mathcal{X}_0}(t) = \langle G, E\rangle_{PZ}$ by optimizing the lower bound $\underline{\alpha} \in \mathbb{R}^p$ and upper bound $\overline{\alpha} \in \mathbb{R}^p$ for the factors $\alpha \in \mathbb{R}^p$ of the polynomial zonotope:

$$\mathcal{C}(\underline{\alpha}, \overline{\alpha}) = \left\{\sum_{i=1}^{h}\left(\prod_{k=1}^{p}\alpha_{(k)}^{E_{(k,i)}}\right)G_{(\cdot,i)} \;\middle|\; \alpha \in [\underline{\alpha}, \overline{\alpha}]\right\},$$

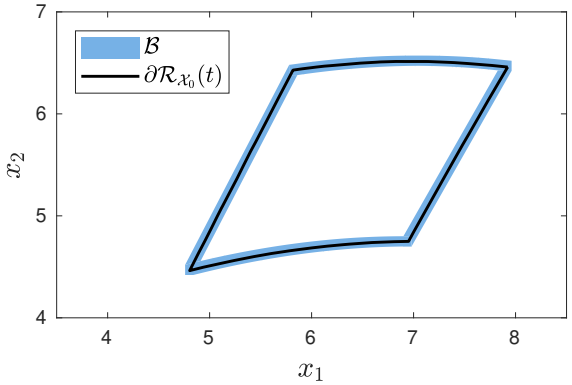

Fig. 4: Visualization of the results from Example 3.

where the bounds $\underline{\alpha}$ and $\overline{\alpha}$ are determined by solving the optimization problem

$$\begin{aligned}&\max_{\underline{\alpha},\overline{\alpha}}\ \texttt{volume}(\mathcal{C}(\underline{\alpha},\overline{\alpha}))\\ &\text{s.t. } \mathcal{C}(\underline{\alpha},\overline{\alpha}) \cap \mathcal{B} = \emptyset,\\ &\qquad [\underline{\alpha},\overline{\alpha}] \subseteq [-\mathbf{1}, \mathbf{1}],\end{aligned} \tag{10}$$

with the operator `volume` returning the volume of a set.

Solving the optimization problem in (10) exactly is computationally expensive. We therefore compute a feasible and close to optimal solution using Alg. 1 as demonstrated with numerical experiments in Sec. V.

Algorithm 1 Compute feasible solution for (10)

Require: Set $\mathcal{B} = \bigcup_{k=1}^{2n}\mathcal{PZ}_k$ enclosing the boundary, outer-approximation of the reachable set $\mathcal{R}^o_{\mathcal{X}_0}(t)$.
Ensure: Optimized lower and upper bounds $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}^p$

```
α̲ = −1,  ᾱ = 1
for k ← 1 to 2n do
    c(α) = 0 ← constraint from C(α̲, ᾱ) ∩ PZ_k (see Prop. 2)
    I = contract(c(α), [α̲, ᾱ] × [−1, 1]) (see Def. 3)
    I = project(I, [1 ... p])
    [α̲, ᾱ] = [α̲, ᾱ] \ I    (see Prop. 3)
end for
```

Alg. 1 iterates over the $2n$ polynomial zonotopes $\mathcal{PZ}_k$ of $\mathcal{B}$ (see (8)), and adapts in each iteration $\underline{\alpha}, \overline{\alpha}$ so that the intersection between $\mathcal{C}(\underline{\alpha}, \overline{\alpha})$ and $\mathcal{PZ}_k$ is empty as required by (10).

We first introduce a new approach to compute the intersection of two polynomial zonotopes:

Proposition 2: *Given two polynomial zonotopes* $\mathcal{PZ}_1 = \langle G_1, E_1\rangle_{PZ}$ *and* $\mathcal{PZ}_2 = \langle G_2, E_2\rangle_{PZ}$*, their intersection is*

computed as

$$\mathcal{PZ}_1 \cap \mathcal{PZ}_2 = \Big\{ \sum_{i=1}^{h_1} \Big(\prod_{k=1}^{p_1} \alpha_{(k)}^{E_{1(k,i)}} \Big) G_{1(\cdot,i)} \;\Big|\; \alpha \in [-\mathbf{1}, \mathbf{1}], \underbrace{\sum_{i=1}^{h_1} \Big(\prod_{k=1}^{p_1} \alpha_{(k)}^{E_{1(k,i)}} \Big) G_{1(\cdot,i)} - \sum_{i=1}^{h_2} \Big(\prod_{k=1}^{p_2} \alpha_{(p_1+k)}^{E_{2(k,i)}} \Big) G_{2(\cdot,i)} = \mathbf{0}}_{c(\alpha)} \Big\}.$$

Proof. We compute the intersection by restricting the factors α of $\mathcal{PZ}_1$ to values that belong to points that are located inside $\mathcal{PZ}_2$, which is identical to adding the equality constraint

$$\sum_{i=1}^{h_1} \Big(\prod_{k=1}^{p_1} \alpha_{(k)}^{E_{1(k,i)}} \Big) G_{1(\cdot,i)} = \sum_{i=1}^{h_2} \Big(\prod_{k=1}^{p_2} \alpha_{(p_1+k)}^{E_{2(k,i)}} \Big) G_{2(\cdot,i)}$$

to $\mathcal{PZ}_1$. □

According to Prop. 2, computation of the intersection $\mathcal{C}(\underline{\alpha}, \overline{\alpha}) \cap \mathcal{PZ}_k$ imposes a constraint $c(\alpha) = \mathbf{0}$ on the factors α. The values of α that satisfy the constraint $c(\alpha) = \mathbf{0}$ correspond to points that intersect the set $\mathcal{PZ}_k$ which encloses a part of the boundary. For computational reasons, we first compute a box enclosure $\mathcal{I}$ of all values α that satisfy $c(\alpha) = \mathbf{0}$ in Line 4 of Alg. 1. Afterwards, we subtract the box $\mathcal{I}$ from the factor domain $[\underline{\alpha}, \overline{\alpha}]$ in Line 6 of Alg. 1, so that the set $\mathcal{C}(\underline{\alpha}, \overline{\alpha})$ corresponding to the updated factor domain $[\underline{\alpha}, \overline{\alpha}]$ does not intersect the set $\mathcal{PZ}_k$ anymore. Since the set $\mathcal{B}$ enclosing the boundary of the exact reachable set is the union over all sets $\mathcal{PZ}_k$, $k = 1, \ldots, 2n$, it holds that $\mathcal{C}(\underline{\alpha}, \overline{\alpha})$ does not intersect $\mathcal{B}$ if it does not intersect any of the sets $\mathcal{PZ}_k$:

$$\mathcal{C}(\underline{\alpha}, \overline{\alpha}) \cap \underbrace{\mathcal{B}}_{=\bigcup_k \mathcal{PZ}_k} = \emptyset \;\Leftrightarrow\; \forall k: \; \mathcal{C}(\underline{\alpha}, \overline{\alpha}) \cap \mathcal{PZ}_k = \emptyset$$

It remains to show how we compute the box $\mathcal{I}$ and how we implement the set subtraction $[\underline{\alpha}, \overline{\alpha}] \setminus \mathcal{I}$. To compute $\mathcal{I}$ we apply contractor programming [17, Chapter 4]:

Definition 3: *(Contractor) Given a box $\mathcal{I} \subset \mathbb{R}^n$ and a nonlinear function $c: \mathbb{R}^n \to \mathbb{R}$ 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: For polynomial constraints the approach in [26] considers extremal functions to contract the domain. The *parallel linearization* contractor [17, Chapter 4.3.4] first encloses the nonlinear constraints by parallel hyperplanes, and then tightens the box domain by solving multiple linear programs. This contractor performs well for cases with multiple constraints. The *forward-backward* contractor [7], which is based on the forward-backward processing of the syntax tree, is advantageous if the size of the box is large. For the implementation of our approach we apply the *parallel linearization* contractor since the computation of the intersection according to Prop. 2 results in multiple constraints. To speed up the contraction we use nonlinear programming to obtain an initial guess for the contracted box.

Finally, we specify how we compute the set difference for boxes:

Proposition 3: *Given two boxes $\mathcal{I}_1 = [l_1, u_1] \subset \mathbb{R}^n$ and $\mathcal{I}_2 = [l_2, u_2] \subset \mathbb{R}^n$ with $\mathcal{I}_2 \subseteq \mathcal{I}_1$, we compute an inner-approximation of the set difference as*

$$\mathcal{I}_1 \setminus \mathcal{I}_2 \supseteq \underbrace{\big\{ x \in \mathcal{I}_1 \;\big|\; x_{(i^*)} \in \widehat{\mathcal{I}} \big\}}_{\mathcal{I}_3}, \tag{11}$$

where

$$i^* = \underset{i \in \{1,\ldots,n\}}{\operatorname{argmax}} \; \max\big(\underbrace{l_{2(i)} - l_{1(i)}}_{d_{l(i)}}, \underbrace{u_{1(i)} - u_{2(i)}}_{d_{u(i)}} \big), \tag{12}$$

$$\widehat{\mathcal{I}} = \begin{cases} [l_{1(i^*)}, l_{2(i^*)}[, & \textit{for} \;\; d_{l(i^*)} \geq d_{u(i^*)} \\ ]u_{2(i^*)}, u_{1(i^*)}], & \textit{otherwise} \end{cases}.$$

Proof. We have to show that $\mathcal{I}_3 \subseteq \mathcal{I}_1$ and $\mathcal{I}_3 \cap \mathcal{I}_1 = \emptyset$. Satisfaction of $\mathcal{I}_3 \subseteq \mathcal{I}_1$ follows directly from the definition of $\mathcal{I}_3$ in (11). Furthermore, we have

$$[l_{1(i^*)}, l_{2(i^*)}[\cap \underbrace{[l_{2(i^*)}, u_{2(i^*)}]}_{\texttt{project}(\mathcal{I}_2, i^*)} = \emptyset$$

and

$$]u_{2(i^*)}, u_{1(i^*)}] \cap \underbrace{[l_{2(i^*)}, u_{2(i^*)}]}_{\texttt{project}(\mathcal{I}_2, i^*)} = \emptyset,$$

so that $\widehat{\mathcal{I}} \cap \texttt{project}(\mathcal{I}_2, i^*) = \emptyset$ holds according to (12). Consequently, it holds according to (11) that $\mathcal{I}_3 \cap \mathcal{I}_1 = \emptyset$. □

We demonstrate the computation of the set difference for boxes by an example:

Example 4: *Given the boxes*

$$\mathcal{I}_1 = [0, 4] \times [0, 6], \quad \mathcal{I}_2 = [2, 3] \times [1, 3],$$

computation of the set difference according to Prop. 3 yields

$$\mathcal{I}_1 \setminus \mathcal{I}_2 \supseteq [0, 4] \times]3, 6].$$

The sets $\mathcal{I}_1$,$\mathcal{I}_2$, and $\mathcal{I}_1 \setminus \mathcal{I}_2$ are visualized in Fig. 5.

Finally, we demonstrate the computation of the set $\mathcal{C}$ using Alg. 1 by an example:

Example 5: *We again consider our running example with the nonlinear system in* (3)*, the initial set $\mathcal{X}_0 = [-1, 1] \times [-1, 1]$, and the set $\mathcal{B}$ that we calculated in Example 3. Execution of*

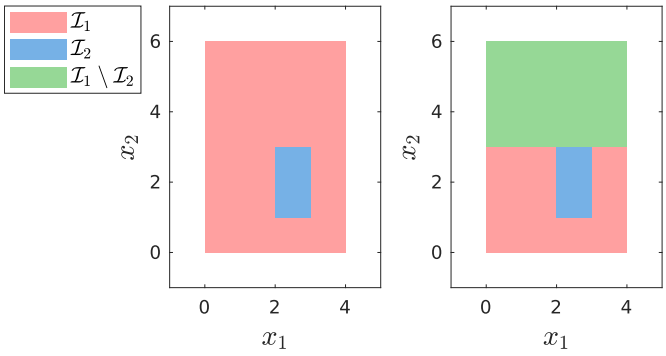

Fig. 5: Visualization of the results from Example 4.

Alg. 1 yields the following values for the four iterations of the loop in lines 2-7:

Iteration 1:

$$\mathcal{I} = \left[\begin{bmatrix} 0.84 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right], \qquad [\underline{\alpha}, \overline{\alpha}] = \left[\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.84 \\ 1 \end{bmatrix} \right]$$

Iteration 2:

$$\mathcal{I} = \left[\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -0.84 \\ 1 \end{bmatrix} \right], \qquad [\underline{\alpha}, \overline{\alpha}] = \left[\begin{bmatrix} -0.84 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.84 \\ 1 \end{bmatrix} \right]$$

Iteration 3:

$$\mathcal{I} = \left[\begin{bmatrix} -0.84 \\ 0.85 \end{bmatrix}, \begin{bmatrix} 0.84 \\ 1 \end{bmatrix} \right], \qquad [\underline{\alpha}, \overline{\alpha}] = \left[\begin{bmatrix} -0.84 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.84 \\ 0.85 \end{bmatrix} \right]$$

Iteration 4:

$$\mathcal{I} = \left[\begin{bmatrix} -0.84 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.84 \\ -0.84 \end{bmatrix} \right], \quad [\underline{\alpha}, \overline{\alpha}] = \left[\begin{bmatrix} -0.84 \\ -0.84 \end{bmatrix}, \begin{bmatrix} 0.84 \\ 0.85 \end{bmatrix} \right],$$

resulting in the set

$$\mathcal{C} = \left\{ \begin{bmatrix} 6.40 \\ 5.60 \end{bmatrix} + \begin{bmatrix} 0.89 \\ 0.07 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0.42 \\ 0.78 \end{bmatrix} \alpha_2 - \begin{bmatrix} 0.01 \\ 0.05 \end{bmatrix} \alpha_1^2 - \begin{bmatrix} 0.01 \\ 0.04 \end{bmatrix} \alpha_1 \alpha_2 \;\middle|\; \alpha_1, \alpha_2 \in [-1, 1] \right\},$$

which is shown in Fig. 6.

D. Verification of Correctness

After computing the set $\mathcal{C}$ using Alg. 1, it remains to verify that $\mathcal{C}$ is a valid inner-approximation of the reachable set. For this, we introduce the time-inverted dynamics

$$\dot{x}(t) = -f(x(t)). \tag{13}$$

of the system in (1). Using (13), we formulate the following theorem:

Theorem 3: *Let $\xi(t, x_0)$ denote the solution to* (13) *at time t for the initial point $x(0) = x_0$. The set $\mathcal{C}$ computed with Alg. 1 is an inner-approximation $\mathcal{C} \subseteq \mathcal{R}_{\mathcal{X}_0}(t)$ of the reachable set $\mathcal{R}_{\mathcal{X}_0}(t)$ if*

$$\xi(t, \texttt{center}(\mathcal{C})) \in \mathcal{X}_0, \tag{14}$$

where the operator `center` *returns the center of a set.*

Proof. According to Theorem 1, a set $\mathcal{C}$ is an inner-approximation of the reachable set if $\mathcal{C}$ is connected, $\mathcal{C} \cap \mathcal{B} = \emptyset$, and $\mathcal{C} \cap \mathcal{R}_{\mathcal{X}_0}(t) \neq \emptyset$. Since $\mathcal{C}$ is a polynomial zonotope and all polynomial zonotopes are connected, it holds that $\mathcal{C}$ is connected. Furthermore, since Alg. 1 computes a set which corresponds to a feasible solution for the optimization problem (10), $\mathcal{C} \cap \mathcal{B} = \emptyset$ holds. Finally, if condition (14) is satisfied, $\texttt{center}(\mathcal{C}) \in \mathcal{R}_{\mathcal{X}_0}(t)$ holds, which proves that $\mathcal{C} \cap \mathcal{R}_{\mathcal{X}_0}(t) \neq \emptyset$. □

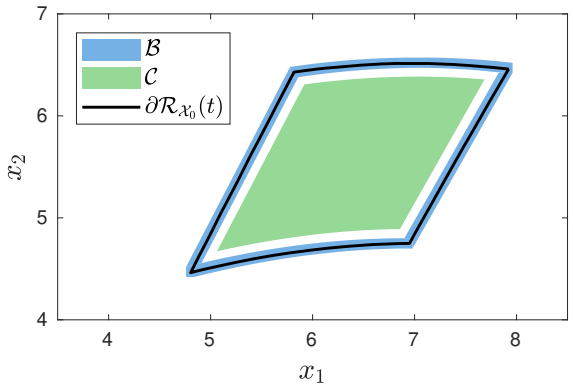

Fig. 6: Visualization of the results from Example 5.

For formal correctness, validated integration methods [25] have to be used for the simulation of the time-inverted dynamics.

IV. COMPUTATIONAL COMPLEXITY

In this section we derive the computational complexity of our approach with respect to the system dimension n. We make the assumption that the evaluation of a nonlinear function in p variables with interval arithmetics [17] has complexity $\mathcal{O}(p)$. Furthermore, it holds for the number of polynomial zonotope factors p that

$$p = c\, n \quad \text{with } c \in \mathbb{R}_{\geq 0}. \tag{15}$$

A. Outer-approximation of the Reachable Set

The computation of an outer-approximation $\mathcal{R}^o_{\mathcal{X}_0}(t)$ with the reachability algorithm from [1] has complexity $\mathcal{O}(n^5)$.

B. Enclosure of the Boundary

The extraction of one reachable subset has according to [20, Sec. 3.3] complexity $\mathcal{O}(n^2)$. Since the computation of the set $\mathcal{B}$ enclosing the boundary requires to extract $2n$ reachable subsets (see (8)), the complexity is $2n \cdot \mathcal{O}(n^2) = \mathcal{O}(n^3)$.

C. Contractor Programming

Computation of the intersection according to Prop. 2 results in n constraints in $2p$ variables. We apply the *parallel linearization* approach to contract the box. According to [17, Chapter 4.3.4], *parallel linearization* requires to evaluate the $n \times 2p$ entries of the Jacobian matrix with interval arithmetics, which has complexity $\mathcal{O}(np^2)$. In addition, the *parallel linearization* contractor requires to solve $4p$ linear programs with $2p$ variables (see [17, Chapter 4.3.4]), which has according to [18] a worst-case complexity of

Tab. 1: Benchmark description, where n is the system dimension, and we compute an inner-approximation and outer-approximation at time t.

Benchmark	n	t	Reference
Brusselator	2	3	[8, Example 3.4.1]
jet engine	2	4	[8, Example 3.3.9]
Rössler	3	1.5	[8, Example 3.4.3]
Lotka-Volterra	4	1	[8, Example 5.2.3]
biological system	7	0.2	[8, Example 5.2.4]

Tab. 2: Comparison of our approach with the approach from [10]. The results for the approach in [10] are taken from [10, Tab. 1].

Benchmark	Our Approach		Approach in [10]	
	time [s]	γ_{min}	time [s]	γ_{min}
Brusselator	64	**0.87**	**55**	0.7
jet engine	**48**	**0.82**	56	0.8
Rössler	**32**	**0.73**	165	0.5
Lotka-Volterra	**238**	0.32	297	**0.4**
biological system	**82**	**0.89**	632	0.25

$4p \cdot \mathcal{O}((2p)^{3.5}) = \mathcal{O}(p^{4.5})$. Since contractor programming is executed $2n$ times in Alg. 1, the overall complexity of the contraction is

$$2n\big(\mathcal{O}(np^2) + \mathcal{O}(p^{4.5})\big) \overset{(15)}{=} \mathcal{O}(n^{5.5}).$$

The warm-start initialization for the contraction using nonlinear programming is not considered for the derivation of the computational complexity since this step is optional.

D. Overall Algorithm

Since all other operations used for our approach have lower complexity, the overall complexity is the sum of the complexities from Sec. IV-A, Sec. IV-B, and Sec. IV-C, which yields

$$\mathcal{O}(n^5) + \mathcal{O}(n^3) + \mathcal{O}(n^{5.5}) = \mathcal{O}(n^{5.5})$$

with respect to the system dimension n.

V. NUMERICAL EXAMPLES

In this section we demonstrate the performance of our novel method on several benchmarks. The computations for our approach are carried out in MATLAB on a 2.9GHz quad-core i7 processor with 32GB memory. Our implementation will be made publicly available with the next release of the CORA toolbox [2].

A. Comparision to [10]

First, we compare our novel approach with the method in [10] using the benchmarks from [10, Sec. VI] (see Tab. 1). To evaluate the precision of the computed inner-approximation, we use the minimum width ratio γ_{min} from [10, Sec. VI] defined as

$$\gamma_{min} = \min_{v \in \mathcal{V}} \frac{|\gamma_i(v)|}{|\gamma_o(v)|}$$

$$\text{with} \quad \gamma_i(v) = \max_{x \in \mathcal{R}^i_{\mathcal{X}_0}(t)} v^T x + \max_{x \in \mathcal{R}^i_{\mathcal{X}_0}(t)} -v^T x \tag{16}$$

$$\gamma_o(v) = \max_{x \in \mathcal{R}^o_{\mathcal{X}_0}(t)} v^T x + \max_{x \in \mathcal{R}^o_{\mathcal{X}_0}(t)} -v^T x,$$

where we select the n axis-aligned unit-vectors as the set of vectors $\mathcal{V} \subset \mathbb{R}^n$ (see [10, Sec. VI]). For a ratio of $\gamma_{min} = 1$, the inner-approximation along the vectors $v \in \mathcal{V}$ is identical to the outer-approximation, whereas for a ratio of $\gamma_{min} = 0$, the inner-approximation is empty. Since γ_{min} cannot be computed exactly for polynomial zonotopes in general, we compute a tight under-approximation instead. In this paper, we focus on initial sets given as boxes, whereas in [10], initial sets given as simplices or ellipsoids are considered. We use box-enclosures of the simplices given in [10, Sec. VI] as initial sets and compare our approach with the results from the method in [10] for initial sets given as simplices (see [10, Tab. 1]). The computations for the method in [10] are carried out in C++ on the machine from the authors of [10].

The results in Tab. 2 demonstrate that for most benchmarks, our novel approach is both faster and more precise than the method from [10], even though we use larger initial sets. Especially for high-dimensional systems our approach exhibits superior performance. For the Brusselator benchmark the inner-approximation and outer-approximation computed with our approach are visualized in Fig. 7.

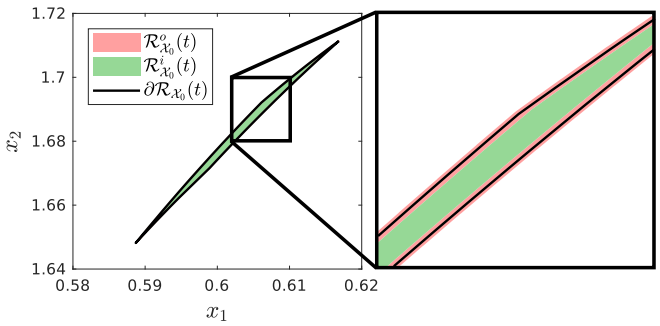

Fig. 7: Inner-approximation and outer-approximation of the reachable set for the Brusselator benchmark at time $t = 3$.

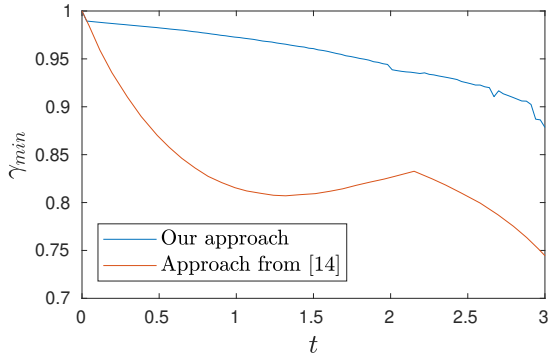

Fig. 8: Minimum width ratio for the Brusselator benchmark over time. The results for the approach in [14] are taken from [14, Fig. 2].

B. Comparision to [14]

In addition, we compare our novel approach with the method in [14], which computes inner-approximations for the projection of the reachable set onto the coordinate axes. For the comparison, the Brusselator benchmark (see Tab. 1) with the initial set $\mathcal{X}_0 = [0.9, 1] \times [0, 0.1]$ (see [14, Sec. 4.2]) is used. The resulting minimum width ratio γ_{min} (see (16)) over time is shown in Fig. 8. It is clearly visible that the results for our approach are much tighter, even though we compute a full inner-approximation and not just an inner-approximation of the projection.

VI. CONCLUSIONS

We introduced a novel approach to simultaneously compute inner-approximations as well as outer-approximations of reachable sets for nonlinear continuous systems. One advantage of our approach is that it has only polynomial complexity with respect to the system dimension so that it is applicable to high-dimensional systems. Since we use a non-convex set representation to represent the inner-approximation, we obtain very accurate results. In addition, since we extract the inner-approximations directly from pre-computed outer-approximations of reachable sets, our approach is computationally very efficient compared to previous methods as we demonstrated on several numerical examples.

VII. 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. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Proc. of the 16th International Conference on Hybrid Systems: Computation and Control*, pages 173–182, 2013.

[2] M. Althoff. An introduction to CORA 2015. In *Proc. of the 1st and 2nd International Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[3] M. Althoff and B. H. Krogh. Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *Proc. of the 15th International Conference on Hybrid Systems: Computation and Control*, pages 45–54, 2012.

[4] M. Althoff, C. Le Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Proc. of the 14th International Conference on Hybrid Systems: Computation and Control*, pages 93–102, 2011.

[5] S. Bak, S. Bogomolov, and M. Althoff. Time-triggered conversion of guards for reachability analysis of hybrid automata. In *Proc. of the 15th International Conference on Formal Modeling and Analysis of Timed Systems*, pages 133–150, 2017.

[6] S. Bak and P. S. Duggirala. Simulation-equivalent reachability of large linear systems with inputs. In *Proc. of the 29th International Conference on Computer Aided Verification*, pages 401–420, 2017.

[7] F. Benhamou and et al. Revising hull and box consistency. In *Proc. of the 16th International Conference on Logic Programming*, pages 230–244, 1999.

[8] X. Chen. *Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models*. PhD thesis, RWTH Aachen University, 2015.

[9] X. Chen, S. Sankaranarayanan, and E. Ábrahám. Taylor model flowpipe construction for non-linear hybrid systems. In *Proc. of the 33rd IEEE Real-Time Systems Symposium*, pages 183–192, 2012.

[10] X. Chen, S. Sankaranarayanan, and E. Ábrahám. Under-approximate flowpipes for non-linear continuous systems. In *Proc. of the 14th International Conference on Formal Methods in Computer-Aided Design*, pages 59–66, 2014.

[11] G. Frehse and et al. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, pages 379–395, 2011.

[12] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Proc. of the 9th International Conference on Hybrid Systems: Computation and Control*, pages 257–271, 2006.

[13] E. Goubault and et al. Inner approximated reachability analysis. In *Proc. of the 17th International Conference on Hybrid Systems: Computation and Control*, pages 163–172, 2014.

[14] E. Goubault and S. Putot. Forward inner-approximated reachability of non-linear continuous systems. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 1–10, 2017.

[15] E. Goubault and S. Putot. Inner and outer reachability for the verification of control systems. In *Proc. of the 22nd International Conference on Hybrid Systems: Computation and Control*, pages 11–22, 2019.

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

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

[18] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proc. of the 16th ACM Symposium on Theory of Computing*, pages 302–311, 1984.

[19] N. Kochdumper and M. Althoff. Sparse polynomial zonotopes: A novel set representation for reachability analysis. *arXiv preprint arXiv:1901.01780*, 2019.

[20] N. Kochdumper, B. Schürmann, and M. Althoff. Utilizing dependencies to obtain subsets of reachable sets. In *Proc. of the 23rd International Conference on Hybrid Systems: Computation and Control*, 2020.

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

[22] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *Proc. of the 3rd International Conference on Hybrid Systems: Computation and Control*, pages 202–214, 2000.

[23] M. Li and et al. Safe over-and under-approximation of reachable sets for autonomous dynamical systems. In *Proc. of the 16th International Conference on Formal Modeling and Analysis of Timed Systems*, pages 252–270, 2018.

[24] I. Mitchell, , A. Bayen, and C. Tomlin. A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

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

[26] G. Trombettoni and et al. A box-consistency contractor based on extremal functions. In *Proc. of the 16th International Conference on Principles and Practice of Constraint Programming*, pages 491–498, 2010.

[27] B. Xue and et al. Reach-avoid verification for nonlinear systems based on boundary analysis. *IEEE Transactions on Automatic Control*, 62(7):3518–3523, 2016.

[28] B. Xue and et al. Safe over-and under-approximation of reachable sets for delay differential equations. In *Proc. of the 15th International Conference on Formal Modeling and Analysis of Timed Systems*, pages 281–299, 2017.

[29] B. Xue, M. Fränzle, and N. Zhan. Under-approximating reach sets for polynomial continuous systems. In *Proc. of the 21st International Conference on Hybrid Systems: Computation and Control*, pages 51–60, 2018.

[30] B. Xue, M. Fränzle, and N. Zhan. Inner-approximating reachable sets for polynomial systems with time-varying uncertainties. *IEEE Transactions on Automatic Control*, 65(4):1468–1483, 2019.

[31] B. Xue, Z. She, and A. Easwaran. Under-approximating backward reachable sets by polytopes. In *Proc. of the 28th International Conference on Computer Aided Verification*, pages 457–476, 2016.