

# Using Intersection of Unions to Minimize Multi-directional Linearization Error in Reachability Analysis

Arvind Adimoolam

Indian Institute of Technology - Kanpur  
Kanpur, India  
[asarvind.adimoolam@gmail.com](mailto:asarvind.adimoolam@gmail.com)

Indranil Saha

Indian Institute of Technology - Kanpur  
Kanpur, India  
[isaha@cse.iitk.ac.in](mailto:isaha@cse.iitk.ac.in)

## ABSTRACT

In piecewise linearization based reachable set computation, different linear approximations are computed around smaller pieces of the reachable set to reduce the linearization error in reachability analysis. However, this approach suffers from curse of dimensionality because the number of pieces required to restrict the linearization error below a threshold can blow up intractably for high-dimensional systems. Alternatively, we can fix the maximum number of divisions of the reachable set and optimize the division vector to minimize the linearization error. But the functions projecting the linearization error along different directions can be different, which have different optimal solutions for the division vector. Still, we may need to minimize the linearization error along multiple directions to achieve good accuracy along any one direction because the differential equations can be coupled. Therefore, we develop a new method of piecewise linearization based reachable set computation that incorporates different optimized divisions of reachable set for different projections of linearization error to improve accuracy. To do so, we use intersection of unions of sets (IoU) to approximate reachable sets such that different unions in the intersection are obtained from optimized division along different directions and forward propagation. We develop an algorithm to propagate the reachable set of the IoU in a coupled way, such that each intersecting union complements the approximation accuracy of other unions. We validate the advantage of using multiple optimal divisions instead of one optimized division. For this, we compare the performance on high dimensional examples, of the proposed algorithm with a variant of the algorithm which uses only one division vector at each time step. We also draw comparison with state-of-the-art methods and demonstrate that the accuracy of our algorithm is at par or better for the benchmarks.

### ACM Reference Format:

Arvind Adimoolam and Indranil Saha. 2022. Using Intersection of Unions to Minimize Multi-directional Linearization Error in Reachability Analysis. In the 25th ACM International Conference on Hybrid Systems: Computation and Control (HSCC '22), May 4–6, 2022, Milan, Italy. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3501710.3519524>

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

HSCC '22, May 4–6, 2022, Milan, Italy

© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-9196-2/22/05...\$15.00  
<https://doi.org/10.1145/3501710.3519524>

## 1 INTRODUCTION

Formal verification of systems modeled by ordinary differential equations typically require approximating the set of trajectories of the ODE in the presence of uncertainties. Exactly computing the set of trajectories of nonlinear ODEs can be computationally intractable, because computing accurate images of nonlinear maps is at least NP Hard [31]. Alternatively, significant research has been carried out on computing over-approximations of the reachable set in terms of data structures, called *set representations*, which can be efficiently manipulated to verify the system properties. [1, 13, 24, 26, 29, 33, 36]. But ensuring bounded accuracy can blow up the computational complexity of any nonlinear reachability analysis algorithm due to NP-hardness of the problem.

One commonly used method in computing reachable sets of nonlinear systems is piecewise linearization, [5, 6, 17, 24, 28, 33], where the nonlinear ODEs are approximated by linear ODEs in smaller subsets of the reachable set. The advantage of piecewise linearization is that reachable sets of linear ODEs can be computed far more efficiently than nonlinear ODEs. However, the number of pieces required to ensure that the linearization error is below a threshold increases exponentially in the dimension of ODEs. To our understanding, no effective solution has yet been proposed to tackle the *curse of dimensionality* in piecewise linearization based reachable set computation.

Alternatively, to avoid blowing up the number of pieces in piecewise linearization, we can fix the number of pieces. Then, we need to optimize the way the reachable set is divided into subsets to minimize the linearization error. However, the function mapping the division vector to the volume of projection of linearization error along a direction is different for different directions, i.e., we have a multi-objective function for optimization. As such, there need not be a single best way of dividing the reachable set. For different projection directions, there can be different optimal ways of splitting the reachable set, also called *Pareto optimal solutions*. But due to dependency between the state variables, the approximation accuracy along one direction can affect the accuracy along other directions. Therefore, we require a method that can incorporate different Pareto-optimal ways of dividing the reachable set in piecewise linearization to improve accuracy.

In this regard, we propose an efficient linearization method for reachability analysis that incorporates the different Pareto-optimal ways of dividing the reachable set in a complementary way to increase accuracy. At different time steps, we intersect different unions resulting from dividing the reachable set in multiple optimal ways and propagate the sets. We propagate the sets in a coupled way where each union complements the accuracy of other unions in the intersection. We use zonotopes as constituent sets of the union

because zonotopes can efficiently approximate reachable sets of linear dynamics post linearization. Thus we represent the forward reachable sets as intersection of union of zonotopes, called IoU zonotope. However, our intersection of unions approach can also be emulated with other set representations in place of zonotopes.

We develop a parallel algorithm to efficiently over-approximate forward reachable sets of an IoU zonotope. The algorithm being parallel, its speed can be boosted by using multiple processors. We evaluate our algorithm on three high-dimensional real-world examples. In the experiments, we validate the advantage of using multiple types of optimized divisions. For this, we compare the performance of the proposed algorithm with a variant of the algorithm which uses only one division vector at each time step. We also compare with other state-of-the-art methods and demonstrate improved accuracy with our approach. In summary, we make the following contributions in this paper.

1) We introduce a new approach of using intersection of unions to minimize the linearization error along multiple directions in reachability analysis.

2) We introduce a new non-convex set representation, which is the *intersection of unions of zonotopes*, called *IoU zonotope*, for efficient reachability analysis of nonlinear systems based on piecewise linearization. We develop a parallel algorithm to propagate the IoU zonotope set representation in a coupled way, where different unions of zonotopes complement each other in increasing the approximation accuracy.

3) We perform experiments on three high-dimensional real-world examples to establish the efficiency of our algorithm and validate its efficacy.

*Related work.* There are efficient algorithms to accurately approximate reachable sets of linear uncertain ODEs [7, 21–23]. This is because linear transformations of polytopic sets can be efficiently approximated for most polytopic representations. In contrast, approximating reachable sets of nonlinear ODEs requires approximating nonlinear transformation of reachable sets, which is a much harder problem [18, 30] whose complexity is at least NP-Hard [31]. Taylor models [13] and polynomial zonotopes [1, 25, 26] have been developed to approximate the nonlinear transformation of reachable sets. But the complexity of Taylor model and polynomial zonotope increases when nonlinear transformations are repeatedly applied. Although there are procedures to control this complexity, they trade approximation accuracy to reduce representation size. Another approach used in numerous research works is piecewise linear approximation of nonlinear systems [5, 6, 17, 24, 28, 33]. In this case, reachability analysis methods of affine hybrid systems are used post piecewise linear approximation. This approach faces curse of dimensionality as discussed before.

For linear systems, decomposition based techniques can be effectively used in high dimensions [8] to reduce computational complexity of reachability algorithms. Unlike this, for nonlinear dynamics, decomposition based techniques [11, 14] only work well under loose coupling between small dimensional subsystems. But our IoU method does not depend on any decomposition. Whereas, it stores the correlation between variables in terms of the relations in different zonotopic pieces of the IoU, which can improve accuracy.

Our IoU zonotope is very different from zonotope bundles [4]. While a zonotope bundle is an intersection of zonotopes and therefore a convex set, an IoU zonotope is an intersection of *unions* of zonotopes, which represents a non-convex set and is geometrically more expressive. Some methods approximate the unbounded time union of reachable sets [32, 34, 37] instead of the reachable set at each individual time point. Such approximation can verify boundedness within a safe set, but not other temporal properties like reachability within a time interval. Our algorithm computes approximation at each time stamp in a time horizon which can be helpful to verify temporal properties.

## 2 NOTATION

We denote the set of real numbers by  $\mathbb{R}$ , rationals by  $\mathbb{Q}$ , and integers by  $\mathbb{Z}$ . A box centered at a point  $c \in \mathbb{Q}^n$  whose distances along coordinate axes from the center is a vector  $r \in \mathbb{Q}_{\geq 0}^n$  is denoted as  $\mathcal{B}(c, r) = \{x \in \mathbb{R}^n \mid \forall i \in \{1, \dots, n\}, |x_i - c_i| \leq r_i\}$ . The projection of a box  $W = \mathcal{B}(c, r)$  along the  $i^{th}$  axis is denoted  $W_i = \mathcal{B}(c_i, r_i)$ . The maximum of the box  $W = \mathcal{B}(c, r)$  is  $\max(W) = c + r$  and minimum is  $\min(W) = c - r$ . The maximum displacement in the box  $W = \mathcal{B}(c, r)$  from center is  $\text{offset}(W) = r$ , which we call *offset*.

The supremum of two vectors  $u \in \mathbb{R}^n$  and  $v \in \mathbb{R}^n$  is denoted as  $u \vee v$  and infimum as  $u \wedge v$  where  $\forall i \in \{1, \dots, n\}, (u \vee v)_{ij} = \max(u_{ij}, v_{ij})$  and  $(u \wedge v)_{ij} = \min(u_{ij}, v_{ij})$ . If  $X$  and  $Y$  are two boxes, then we denote their box hull as a box  $X \vee Y$  where  $\max(X \vee Y) = \max(X) \vee \max(Y)$  and  $\min(X \vee Y) = \min(X) \wedge \min(Y)$ . The Minkowski sum of two sets  $\Psi_1$  and  $\Psi_2$  is defined as  $\Psi_1 \oplus \Psi_2 = \{x + y \mid x \in \Psi_1, y \in \Psi_2\}$ . The Minkowski sum of two boxes  $\Psi_1 = \mathcal{B}(c, r)$  and  $\Psi_2 = \mathcal{B}(c', r')$  is another box which can be computed as  $\Psi_1 \oplus \Psi_2 = \mathcal{B}(c + c', r + r')$ .

The matrix containing absolute values of the elements of a matrix  $A$  is denoted  $|A|$ , i.e.,  $|A|_{ij} = |A_{ij}|$ . An  $n \times m$  matrix whose every element is equal to a real number  $r$  is denoted  $[r]_{n \times m}$ . For a matrix  $A \subseteq \mathbb{R}^{n \times m}$  and  $\Psi \subseteq \mathbb{R}^n$ , we denote  $A\Psi = \{Ax \mid x \in \Psi\}$ . The diagonal square matrix containing elements of a vector  $r$  along its diagonal is  $\text{diag}(r)$ .

Given  $\Psi \subseteq \mathbb{R}^n$ , a function  $f : \Psi \rightarrow \mathbb{R}^m$  and  $S \subseteq \Psi$ , we denote  $f(S) = \{f(x) \mid x \in S\}$ . We define  $\mathbb{F}(\Psi)$  as the subset of  $\{f : \Psi \rightarrow \mathbb{R}\}$  whose members can be constructed recursively as,

- If  $\exists v \in \mathbb{Q}^n$  s.t.  $\forall x \in \Psi, f(x) = v^T x$ , then  $f \in \mathbb{F}(\Psi)$ .
- If  $f_1, f_2 \in \mathbb{F}(\Psi)$ , then  $\{f_1 + f_2, f_1 - f_2, f_1 f_2, \sin \circ f, \cos \circ f, \exp \circ f\} \subseteq \mathbb{F}(\Psi)$ .
- For all  $f_1, f_2 \in \mathbb{F}(\Psi)$  such that  $f_2(\Psi) \subseteq \mathbb{R}_{>0}$  (positive function), then  $\left\{\frac{f_1}{f_2}, \log \circ f_2\right\} \subseteq \mathbb{F}(\Psi)$ .

The Jacobian of a function  $f$  at a point  $x$  is denoted  $\nabla f(x)$ . If  $\Psi$  is an open set, then any  $f \in \mathbb{F}(\Psi)$  is infinitely differentiable and  $\nabla f \in \mathbb{F}(\Psi)$ , which can be computed symbolically. For any function  $f \in \mathbb{F}(\Psi)$  and a box  $\mathcal{B}(c, r) \subseteq \Psi$ , we can compute an interval over-approximation of  $f(\mathcal{B}(c, r))$  using interval arithmetic [10], which we denote as  $\bar{f}(\mathcal{B}(c, r))$ . If  $g \in \mathbb{F}(\Psi)^m$  is a tuple of functions, then we denote  $\bar{g}(\mathcal{B}(c, r)) = \bar{g}_1(\mathcal{B}(c, r)) \times \dots \times \bar{g}_m(\mathcal{B}(c, r))$ .

### 3 NONLINEAR SYSTEM AND REACHABLE SET

A nonlinear system is specified by a function  $f : \Gamma \times U \rightarrow \Gamma$ , called *vector field*, where  $\Gamma \subseteq \mathbb{R}^n$  is an open set called the state space,  $U \in \mathbb{R}^m$  is the input set, and  $f \in \mathbb{F}(\Gamma \times U)^n$ . A function  $x : [0, \infty) \rightarrow \Gamma$  is called a state trajectory of  $f$  if there exists a measurable function  $u : [0, \infty) \rightarrow U$ , called input trajectory, such that the following Lebesgue integral is satisfied  $\forall t \in [0, \infty)$ :

$$x(t) - x(0) = \int_0^t f(x(t), u(t)) dt. \quad (1)$$

The set of all possible trajectory states  $x(t)$  where  $x$  is a state trajectory of  $f$ , given the initial condition  $x(0) \in \Psi \subseteq \mathbb{R}^n$ , is called the reachable set at time  $t$  originating from  $\Psi$ , which we denote as  $\mathcal{R}_Y(\Psi, t)$ . The set of reachable states in a time interval  $[t_1, t_2]$  originating from  $\Psi$  is  $\mathcal{R}_Y(\Psi, [t_1, t_2]) = \bigcup_{t \in [t_1, t_2]} \mathcal{R}_Y(\Psi, t)$ .

We consider the following problem of computing the directional bounds on the reachable set of a nonlinear system in a sequence of successive time intervals.

**PROBLEM 3.1.** Let  $f : \Gamma \times U \rightarrow \Gamma$  be a nonlinear system where  $U \subseteq \mathbb{R}^m$  is a box. Let  $\Psi$  be a box such that  $\Psi \subseteq \Gamma$ ,  $H \in \mathbb{Q}^{p \times n}$  be a rational matrix and  $\delta^{\max} \in (0, \infty)$ . For  $N \in \mathbb{Z}_{\geq 1}$ , we have to compute a finite sequence of vectors  $\langle h^i \rangle_{i=1}^N$  such that

$$\forall i \in \{1, \dots, N\} \max_{x \in \mathcal{R}_Y(\Psi, [(i-1)\delta^{\max}, i\delta^{\max}])} Hx \leq h^i.$$

### 4 LINEARIZATION

We will propose a new piecewise linearization approach to the reachability analysis problem which combines different piecewise linearizations that minimize the linearization error along different directions. For explaining the motivation for our algorithm, we briefly revisit some concepts related to linearization in this section.

A linear system is a special case of the nonlinear system where the vector field is linear. The vector field  $g : \Gamma \times U \rightarrow \Gamma$  of an  $n$ -dimensional linear system with  $m$  inputs is

$$g(x) = Ax + Bu$$

where  $A \in \mathbb{R}^{n \times n}$  is called *state-action matrix*,  $B \in \mathbb{R}^{n \times m}$  is called *input-action matrix*. The reachable set of a linear system in high dimensions can be efficiently over-approximated by zonotopes using the algorithm in [22]. The advantage of zonotope is that its reachable set under a linear transformation is another zonotope which can be computed very efficiently and exactly. A zonotope is defined as follows.

**Definition 4.1** ([22]). For  $n, l \in \mathbb{Z}_{\geq 1}$ , let  $G \in \mathbb{R}^{n \times nl}$ , called generator matrix, and  $c \in \mathbb{R}^n$ , called center. Then

$$\text{Zon}(G, c) = \{c + G\zeta \mid \zeta \in \mathcal{B}(0, [1]_{nl \times 1})\}$$

is a zonotope of order  $l$ .

So, a box  $\mathcal{B}(c, r)$  is equivalently a zonotope  $\text{Zon}(\text{diag}(r), c)$ . Given a set of directions which are row vectors of a matrix  $H \in \mathbb{R}^{p \times n}$ , the directional bounds on a zonotope  $\text{Zon}(G, c)$  along these directions can be computed as follows (Lemma 1 in [3]).

$$\max_{x \in \text{Zon}(G, c)} Hx = Hc + |HG| [1]_{\text{cols}(G) \times 1}. \quad (2)$$

Based on above equation, we can compute the smallest box over-approximation of a zonotope  $\text{Zon}(G, c)$  as

$$\text{Zon}(G, c) \subseteq \mathcal{B}\left(Hc, |H| [1]_{\text{cols}(G) \times 1}\right).$$

We will denote the box overapproximation in the R.H.S of above equation as  $\text{Boxapp}(\text{Zon}(G, c))$

In this paper, we subsume the algorithm in [22] to compute the reachable set over-approximation of a linear system originating from a zonotope. Henceforth, given a linear system  $g : \Gamma \times U \rightarrow \Gamma$  where  $U$  is a box, a zonotope  $\Psi$  and a time interval  $[t_1, t_2] \subseteq [0, \infty)$ , we denote  $\mathcal{Z}_g(\Psi, [t_1, t_2])$  as the zonotopic over-approximation of  $\mathcal{R}_g(\Psi, [t_1, t_2])$  which is computed using the algorithm in [22], i.e.,

$$\mathcal{R}_g(\Psi, [t_1, t_2]) \subseteq \mathcal{Z}_g(\Psi, [t_1, t_2]).$$

The interested reader may refer to [22] for details of the algorithm.

As the reachable set of a linear system can be computed very efficiently, we can over-approximate the reachable set of a nonlinear system at a future time point  $t$  by using a linear approximation of the dynamics in the interval  $[0, t]$ . Let us consider a nonlinear system  $f : \Gamma \times U \rightarrow \Gamma$  where  $U = \mathcal{B}(b, s)$  is a box. Let us consider a set  $\Psi \subseteq \Gamma$  such that  $\Psi \subseteq \mathcal{B}(c, r)$ . The algorithm in [5] uses Taylor expansion and interval arithmetic to compute a linear system

$$\text{Linearize}(f, \Psi, t) = g : \Gamma \times U \times W_{\Psi, f, t} \rightarrow \Gamma$$

and  $J_{\Psi, f, t} \subseteq \mathbb{R}^n$  such that  $W \subseteq \mathbb{R}^n$  and  $\forall (x, u) \in \mathcal{R}_f(\Psi, [0, t]) \times U$ , the following is true.

$$f(x, u) \in g(x, u, W) \quad (3)$$

$$W_{\Psi, f, t} = \overline{E_f}(\Psi, U, r, s) \oplus J_{\Psi, f, t} \quad (4)$$

$$E_f(x, u, r, s) = 0.5 \begin{bmatrix} r & s \end{bmatrix} [\nabla (\nabla f_i)(x, u)] \begin{bmatrix} r \\ s \end{bmatrix} \quad (5)$$

Therefore, if  $\Psi$  is a zonotope, then we can overapproximate the reachable set of the nonlinear system  $f$  at time  $t$  by the zonotopic over-approximation of the reachable set of the linear system  $g$  as

$$\mathcal{R}_f(\Psi, t) \subseteq \mathcal{Z}_g(\Psi, t) \quad (6)$$

The error in the above reachability overapproximation increases with the size of the set  $\Psi$  as described by Equation 5. We shall call the box  $\overline{E_f}(\Psi, U, r, s)$  computed using interval arithmetic as the linearization error in this paper, which is a multi-dimensional set.

### 5 INTERSECTION OF UNIONS REACHABILITY ANALYSIS

In the previous section, we have discussed the method of approximating the reachable set of a nonlinear system using linearization. The method incurs a linearization error which is proportional to the size of the reachable set (see (5)). Therefore, we can reduce the linearization error by dividing the reachable set into smaller subsets and linearizing around the smaller subsets. But the number of divisions required to reduce the linearization error below a threshold can increase exponentially in the dimension of the state space. Therefore, restricting the linearization error below a threshold in higher dimensions can be computationally intractable. Alternatively, we can fix the number of divisions and optimize the way we divide the reachable set so as to minimize the linearization

error. However, as the Taylor error (5) is a multi-dimensional set, there can be different optimal ways of dividing the reachable set for minimizing the projection of linearization error along different directions. In other words, finding divisions to minimize the projection of linearization error along different directions is a multi-objective optimization problem which has many Pareto-optimal solutions, but no unique best solution.

*Example 5.1.* We adapted the time-discretized model of an autonomous car from [27] into a platoon of two cars with continuous-time dynamics and designed a stabilizing feedback controller. We have a 12-dimensional model where the state of first car is denoted by  $x \in \mathbb{R}^6$  and the state of the second car by  $y \in \mathbb{R}^6$ . The displacement of the cars are  $x_1, y_1$ , respectively. The steering angles are  $x_2, y_2$ , the velocities are  $x_3, y_3$ , the yaw angles are  $x_4, y_4$ , the rates of change of yaw angles are  $x_5, y_5$ , and the slip angles are  $x_6, y_6$ , respectively for both cars. There is a disturbance input  $u \in \mathbb{R}^2$ . The dynamics is given by the following vector field where the parameters are  $g = 9.81, m = 1093.3, \mu = 1.0489, l_f = 1.156, l_r = 1.422, h_{cg} = 0.2, I_z = 1791.6, C_{Sf} = 20.89, C_{Sr} = 20.89, r = 4, K_1 = 3$ . We reduced the parameter  $h_{cg}$  compared to the original model for increasing stability.

$$\begin{aligned} f_1(x, y, u) &= x_3 \cos(x_4 + x_6), \quad f_2(x, y, u) = -K_0(x_4 + x_6 + x_2), \\ f_3(x, y, u) &= K_1(5.5 - x_3) + u_1, \quad f_4(x, y, u) = x_5 \\ f_5(x, y, u) &= \frac{\mu m}{I_z(l_r + l_f)} (l_f C_{Sf}(gl_r - f_3(x, y, u)h_{cg}))x_2 + \\ (l_r C_{Sr}(gl_f + f_3(x, y, u)h_{cg}) - l_f C_{Sf}(gl_r - f_3(x, y, u)h_{cg}))x_6 \\ - (l_f^2 C_{Sf}(gl_r - f_3(x, y, u)h_{cg})l_r^2 C_{Sr}(gl_f + f_3(x, y, u)h_{cg}))\frac{x_5}{x_3} \\ f_6(x, y, u) &= \frac{\mu}{x_3(l_r + l_f)} (C_{Sf}(gl_r - f_3(x, y, u)h_{cg}))x_2 \\ + (C_{Sr}(gl_f + f_3(x, y, u)h_{cg}) - C_{Sf}(gl_r - f_3(x, y, u)h_{cg}))x_6 \\ - (l_f C_{Sf}(gl_r - f_3(x, y, u)h_{cg}) + l_r C_{Sr}(gl_f + f_3(x, y, u)h_{cg}))\frac{x_5}{x_3} - x_5 \\ f_7(x, y, u) &= y_3 \cos(y_4 + y_6) \\ f_8(x, y, u) &= -K_0(y_4 + y_6 + y_2) \\ f_{10}(x, y, u) &= K_1(x_3 - y_3) \\ f_{11}(x, y, u) &= y_5 \\ f_{12}(x, y, u) &= \frac{\mu m}{I_z(l_r + l_f)} (l_f C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}))y_2 + \\ (l_r C_{Sr}(gl_f + f_{10}(x, y, u)h_{cg}) - l_f C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}))y_6 \\ - (l_f^2 C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}) + l_r^2 C_{Sr}(gl_f + f_{10}(x, y, u)h_{cg}))\frac{y_5}{y_3} \\ y'_6 &= \frac{\mu}{y_3(l_r + l_f)} (C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}))y_2 + \\ (C_{Sr}(gl_f + f_{10}(x, y, u)h_{cg}) - C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}))y_6 \\ - (l_f C_{Sf}(gl_r - f_{10}(x, y, u)h_{cg}) + l_r C_{Sr}(gl_f + f_{10}(x, y, u)h_{cg}))\frac{y_5}{y_3} - y_5 \end{aligned}$$

In the above example,  $f_1(x, y, u)$  is a function of only  $x_3, x_4$  and  $x_6$ . So, minimizing the instantaneous linearization error along  $x_1$  coordinate requires dividing only across  $x_3, x_4$ , and  $x_6$ . On the other hand, minimizing the instantaneous linearization error along  $y_1$  coordinate requires dividing only across  $y_3, y_4$ , and  $y_6$  because  $f_7(x, y, u)$  is only a function of  $y_3, y_4$  and  $y_6$ . So, the optimal way of dividing the reachable set for minimizing linearization error along different directions can be different, i.e., it is a multi-objective optimization problem with no unique optimum.  $\blacktriangleleft$

To integrate the multiple-optimum ways of division for different directions in reachability analysis, we intersect the unions of sets resulting from different optimal divisions corresponding to different directions of minimizing the linearization error. This results in intersection of unions of sets representation of the reachable set. We will first discuss how to cast a box into IoU of boxes that minimizes linearization error along different directions. Since the forward reachable sets of these boxes post linearization are zonotopes (not boxes), we get intersection of unions of zonotopes as forward reachable sets.

## 5.1 Optimizing different division vectors for a set of projection directions

We can divide a box by partitioning across each axis and taking Cartesian product of the different intervals obtained. Let  $\mu \in \mathbb{Z}_{\geq 1}^n$  be a vector of integers, called *division vector*, where the value  $\mu_i, i \in \{1, \dots, n\}$  denotes the number of divisions across the  $i^{th}$  axis. Given a box  $\Psi = \mathcal{B}(c, r) \subseteq \mathbb{R}^n$ , a division vector  $\mu$  gives a partition of  $\Psi$  with the following collection of boxes:

$$\text{divs}(\Psi, \mu) = \left\{ \mathcal{B}(x, y) \mid \forall i \in \{1, \dots, n\} \quad y_i = \frac{r_i}{\mu_i}, \right. \\ \left. \exists j \in \{0, \dots, \mu_i - 1\} \quad x_i = x_i - r_i + 2(j - 1) \frac{r_i}{\mu_i - 1} \right\}.$$

Let us consider an  $n$ -dimensional nonlinear system  $f$  where  $U = \mathcal{B}(b, s)$  is a box. An upper bound on the offset of the linearization error for all sets in the partition  $\text{divs}(\Psi, \mu)$  can be computed from Equation (5) as follows.

$$\Delta_{f, \Psi}(\mu) = \overline{E}_f(\Psi, U, r(\text{diag}(\mu))^{-1}, s). \quad (7)$$

According to (7), the volume of linearization error reduces by increasing the division vector  $\mu$ . Next, an upper bound on the magnitude of projection of the linearization error of any of the boxes in the partition along a direction  $\alpha \in \mathbb{Q}^n$  can be computed as

$$|\alpha|^T \text{offset}(\Delta_{f, \Psi}(\mu)). \quad (8)$$

Now, given a bound  $\eta \in \mathbb{Z}_{\geq 0}$  on the logarithm number of divisions, i.e.,  $\sum_{i=1}^n \log_2 \mu_i \leq \eta$ , we want to find a division vector  $\mu$  that minimizes the upper bound on the magnitude of projection of the linearization error error in Equation (8). Solving for the most optimal division vector can be of exponential complexity in the dimension  $n$ . So, we use the greedy optimization given in Algorithm 1. In each iterative step of this algorithm, we increment the component of division vector along an optimum axis by multiplying by 2, instead of adding 1, for faster convergence. We denote the collection of subsets from the optimized partition that minimizes the linearization error along a direction  $\alpha$  as

$$\text{optdivs}(f, \Psi, \alpha, \eta) = \text{divs}(\Psi, \text{optvect}(f, \Psi, \alpha, \eta)).$$

where  $\text{optvect}(f, \Psi, \alpha, \eta)$  is computed in Algorithm 1. But for different projection directions, we get different types of partitions, which we shall use in a complementary way to minimize linearization error and increase accuracy of reachability computation.

---

**Algorithm 1:** Optimizing division vectors given a set of projection directions

---

```

Input:  $n$ -dimensional nonlinear system  $f$ , box  $\Psi$ , threshold  $\eta$ , finite set of direction vectors  $Dirs \subseteq \mathbb{Q}^n$ .
Output: Set of optimized division vectors  $optvect(f, \Psi, Dirs, \eta)$ 

1  $optvect(f, \Psi, Dirs, \eta) \leftarrow \emptyset$ .
2 for  $\alpha \in Dirs$  do
3    $optvect(f, \Psi, \alpha, \eta) \leftarrow [1]_{n \times 1}$ 
4   while  $\sum_{i=1}^n \log_2(optvect(f, \Psi, \alpha, \eta))_i \leq \eta$  do
5     for  $i \in \{1, \dots, n\}$  do
6        $\mu \leftarrow optvect(f, \Psi, \alpha, \eta)$ 
7        $\mu_i \leftarrow 2\mu_i$ .
8        $\gamma(i) = |\alpha|^T \Delta_{f, \Psi}(\mu)$ .
9     end for
10     $j \leftarrow \arg \min_{i=1}^n \gamma(i)$ .
11     $(optvect(f, \Psi, \alpha, \eta))_j \leftarrow 2(optvect(f, \Psi, \alpha, \eta))_j$ .
12  end while
13   $optvect(f, \Psi, Dirs, \eta) \leftarrow$ 
14     $optvect(f, \Psi, \alpha, \eta) \cup optvect(f, \Psi, Dirs, \eta)$ .
14 end for

```

---

## 5.2 Propagating reachable sets as IoU of zonotopes

Let us consider a finite set of direction vectors  $Dirs \subseteq \mathbb{Q}^n$  and a threshold  $\eta$  for the logarithm of number of divisions. Given a set of states  $\Psi \subseteq \mathbb{R}^n$ , we want to compute an over-approximation of the reachable set  $\mathcal{R}_f(\Psi, t)$  after a small time elapse  $t$  using linearization, such that we minimize the volume of projections of linearization error along the vectors in  $Dirs$ . If  $\Psi$  is a box, then for each direction we can compute a division which minimizes the linearization error along that direction. Then we can approximate the reachable sets of each division using linearization and obtain a zonotope. Since the type of optimized division is different for different directions, we can intersect the resulting reachable sets for different types of divisions. This gives an intersection of union (IoU) of zonotopes as the approximation of reachable set.

$$\mathcal{R}_f(\Psi, t) \subseteq \bigcap_{X \in Dirs} \left( \bigcup_{X \in optdivs(f, \Psi, \alpha, \eta)} \mathcal{Z}_f(X, t) \right) \quad (9)$$

In this context, we define the computer representation of an IoU of zonotopes below.

*Definition 5.2.* An  $n$ -dimensional IoU zonotope  $Z$  with  $k$  intersections of unions of  $l$  pieces in each intersection is an  $k \times l$  matrix whose each element  $Z_{ij} : i \in \{1, \dots, \text{rows}(Z)\}, j \in \{1, \dots, \text{cols}(Z)\}$  is an  $n$ -dimensional zonotope. The IoU zonotope denotes the set

$$[Z] = \bigcap_{i=1}^{\text{rows}(Z)} \bigcup_{j=1}^{\text{cols}(Z)} Z_{ij}.$$

We call an IoU zonotope with a single row in its matrix representation as a *union zonotope* because it represents a union of zonotopes without intersection. Given two IoU zonotopes  $Z$  and  $Z'$ , we represent the IoU zonotope resulting from their intersection

as  $Z \sqcap Z'$ , i.e.  $[Z \sqcap Z'] = Z \cap Z'$ . If  $Z$  and  $Z'$  are two union zonotopes, then we represent the union zonotope resulting from their union as  $Z \sqcup Z'$ , i.e.,  $[Z \sqcup Z'] = Z \cup Z'$ .

Since the forward reachable set of a box  $\Psi$  is approximated by an IoU zonotope in (9), we need a method to approximate the reachable set of an IoU zonotope at future time. In this process, to reduce the linearization error in propagating an IoU zonotope using linearization, we have to divide the IoU zonotope into subsets which can be conveniently manipulated for reach set computation. But accurate division of an IoU zonotope into subsets which can be conveniently manipulated during reachability computations can be intractable. Instead, we can divide a box over-approximation of the IoU zonotope, since boxes can be divided accurately into union of boxes. However, over-approximating the IoU zonotope by a box adds *wrapping error* which can reduce the accuracy of reach set approximation. Therefore, to minimize both the linearization error as well as wrapping error in propagating an IoU zonotope, we intersect it with the optimized IoU division of its box over-approximation. This computation is given in the below theorem. In this computation, we further reduce the linearization error by linearizing around a region which is an intersection of the box hull of the zonotope and the box hull of the IoU. The explanation is provided following the theorem.

**THEOREM 5.3.** Let  $Z$  be an IoU zonotope,  $Dirs \subseteq \mathbb{Q}^n$  be a finite set of directions and  $t \in [0, \infty)$ . Let us define the following quantities.

$$\Omega = \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} \text{Boxapp}(Z_{ij}) \quad (10)$$

$$Z^{new} = Z \sqcap \left( \bigcap_{\alpha \in Dirs} \bigcup_{X \in optdivs(f, \Omega, \alpha, \eta)} X \right) \quad (11)$$

$$\mathcal{L}^{ij} = \text{Linearize}\left(f, \text{Boxapp}(Z_{ij}^{new}) \wedge \Omega, t\right) \quad (12)$$

Then we get the following over-approximation of the reachable set of an IoU after time elapse  $t$ .

$$\mathcal{R}_f(Z, t) \subseteq \bigcap_{i=1}^{\text{rows}(Z^{new})} \bigcup_{j=1}^{\text{cols}(Z^{new})} \mathcal{Z}_{\mathcal{L}^{ij}}(Z_{ij}^{new}, t) \quad (13)$$

**PROOF.** In the above equations,  $\Omega$  is the box over-approximation of  $Z$  computed by intersection of joins of box overapproximations of zonotopes in different unions. Then  $\left( \bigcap_{\alpha \in Dirs} \bigcup_{X \in optdivs(f, \Omega, \alpha, \eta)} X \right)$  is the optimized division IoU obtained by dividing  $\Omega$ . So,  $[Z^{new}] = [Z]$ . Then we get

$$\begin{aligned} \mathcal{R}_f(Z, t) &\subseteq \bigcap_{i=1}^{\text{rows}(Z^{new})} \bigcup_{j=1}^{\text{cols}(Z^{new})} \mathcal{R}_f(Z_{ij}^{new}, t) \\ &= \bigcap_{i=1}^{\text{rows}(Z^{new})} \bigcup_{j=1}^{\text{cols}(Z^{new})} \mathcal{R}_f(Z_{ij}^{new} \cap \text{Boxapp}(Z), t) \end{aligned} \quad (14)$$

From Equation 6 we derive

$$\begin{aligned}\mathcal{R}_f(Z, t) &\subseteq \bigcap_{i=1}^{\text{rows}(Z^{\text{new}})} \bigcup_{j=1}^{\text{cols}(Z^{\text{new}})} \mathcal{R}_f\left(Z_{ij}^{\text{new}} \cap \Omega, t\right) \\ &\subseteq \bigcap_{i=1}^{\text{rows}(Z^{\text{new}})} \bigcup_{j=1}^{\text{cols}(Z^{\text{new}})} \mathcal{R}_{\mathcal{L}ij}\left(Z_{ij}^{\text{new}} \cap \Omega, t\right) \\ &\subseteq \bigcap_{i=1}^{\text{rows}(Z^{\text{new}})} \bigcup_{j=1}^{\text{cols}(Z^{\text{new}})} \mathcal{Z}_{\mathcal{L}ij}\left(Z_{ij}^{\text{new}}, t\right).\end{aligned}\quad \square$$

In the above theorem, while propagating a zonotope in the IoU, we linearize in the neighborhood of the intersection of zonotope and the IoU, instead of just linearizing in the neighborhood of the zonotope. This is given in (12). The reason is that along some axes, the projection volume of a constituent zonotope can happen to be larger than that of the of IoU. So, we linearize in a neighborhood of the intersection of the zonotope and the IoU, instead of just considering the zonotope alone. This computation, apart from the optimization of divisions, further reduces the linearization error. *So, each zonotope in the IoU complements other zonotopes in reducing the linearization error, which is an advantage of the above computation.*

### 5.3 Reducing representation size of IoU

In Theorem 5.3, we intersected an IoU zonotope with the optimized IoU casting of the box over-approximation of the IoU (11) to minimize the linearization error. However, this step increases the number of intersections in the IoU and therefore the representation size of the IoU in a computer. If we compute the reachable set inductively, the representation size will increase in each iteration. It will slow down computation speed. Therefore, we want to limit the IoU size within a threshold.

Given a threshold number of intersections  $\text{Maxintrs} > 1$  and an IoU zonotope  $Z$  such that  $\text{cols}(Z) > \text{Maxintrs}$ , we will construct a new IoU zonotope  $Z'$  using the zonotopes in  $Z$  such that  $\llbracket Z \rrbracket \subseteq \llbracket Z' \rrbracket$ . But we also want the size of  $Z'$  to be as small as possible for accuracy of reach set approximation. In this context, we define the over-approximation measure  $Z'$  relative to  $Z$  when  $Z \subseteq Z'$  and  $\text{Boxapp}(Z) \subseteq \text{Boxapp}(Z')$ , as follows. For an IoU zonotope  $Z$ , let us denote a union zonotope  $Z_i = \bigcup_{j=1}^{\text{cols}(Z)} Z_{ij}$ . A box overapproximation of IoU zonotope  $Z$  can be computed by taking intersection of joins of constituent zonotopes as

$$\text{Boxapp}(Z) = \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} \text{Boxapp}(Z_{ij}).$$

$$\text{Then we define } \frac{Z'}{Z} = \frac{1}{n} \sum_{i=1}^n \beta_i \quad \text{where } \forall i \in \{1, \dots, n\}$$

$$\beta_i = \begin{cases} \frac{\text{offset}(\text{Boxapp}(Z'))_i}{\text{offset}(\text{Boxapp}(Z))_i} & \text{if } \text{offset}(\text{Boxapp}(Z))_i > 0 \\ 1 & \text{if } \text{offset}(\text{Boxapp}(Z'))_i = 0 \\ \infty & \text{otherwise} \end{cases} \quad (15)$$

We use the Algorithm 2 to reduce the number of intersections. In the algorithm, we inductively intersect union zonotopes which reduce the above measure for the resulting IoU. To reduce size further, we terminate the algorithm if the final bounds of the new

---

### Algorithm 2: Reducing number of intersections in IoU

---

```

Input: IoU zonotope  $Z$ ,  $\text{Maxintrs} \in \mathbb{Z}_{>1} : \text{Maxintrs} < \text{cols}(Z)$ 
Output: IoU zonotope denoted:  $\text{ReduceInter}(Z)$ 
1 while  $\text{rows}(Z') \leq \text{Maxintrs} - 1$  and  $\text{Boxapp}(Z') \neq \text{Boxapp}(Z)$  do
2   for  $j \in \{1, \dots, n\}$  do
3      $Z^{\text{new}} = Z' \sqcap Z_j$ .
4      $\gamma(j) = \frac{Z^{\text{new}}}{Z}$ .
5   end for
6    $i \leftarrow \arg \min_{j=1}^{\text{cols}(Z)} \gamma(j).$ 
7    $Z' \leftarrow Z' \sqcap Z_i.$ 
8 end while
9  $Z' \leftarrow Z' \sqcap \left( \bigcup_{i=1}^{\text{rows}(Z)} \text{Boxapp}(Z) \right).$ 
10  $\text{ReduceInter}(Z, \text{Maxintrs}) \leftarrow Z'.$ 

```

---

IoU along coordinate directions in an iteration are the same as that of the original IoU. We denote the zonotope with reduced number of intersections as  $\text{ReduceInter}(Z, \text{Maxintrs})$ . This algorithm is only a heuristic procedure and there can be other ways to reduce the number of intersections with improved accuracy. The problem of efficiently reducing number of intersections has scope for further research.

The order of zonotope can also increase while computing forward reachable sets post linearization. Reducing the order of a zonotope is a well studied problem and there are numerous algorithms for this purpose (see survey [38]). We use a variant of [15] to reduce the order of the zonotope. Given a zonotope  $Z$  and  $l \in \mathbb{Z}_{\geq 1}$ , we denote the zonotope obtained by reducing the zonotope order to  $l$  as  $\text{ReduceOrder}(Z, l)$ .

### 5.4 Parallel algorithm for reachability analysis

Given a matrix  $H \in \mathbb{Q}^{p \times n}$ , the directional bounds for an IoU zonotope  $Z = \bigwedge_{i \in \text{rows}(Z)} \bigvee_{j \in \text{cols}(Z)} \text{Zon}(G_{ij}, c_{ij})$  can be computed based on Equation (2) as

$$\max_{x \in Z} Hx \leq \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} \mathcal{B}\left(Hc_{ij}, |HG_{ij}| [1]_{\text{cols}(G_{ij}) \times 1}\right).$$

We will denote the R.H.S of above equation as  $H \odot Z$ . Then, using Theorem 5.3, we developed a parallel Algorithm 3 to compute the directional bounds on the reachable set of a nonlinear system in a sequence of successive time intervals. The algorithm solves Problem 3.1. The algorithm being parallel, we can use multiple CPU cores to boost the computation speed.

*Choosing directions for minimizing linearization error.* The set of directions  $\text{Dirs}$  along which the linearization error is minimized can be defined by the user. But as a heuristic, in our experiments we choose the directions of the coordinate axes as well as the rows of the Jacobian of  $f$  with respect to state variables at the center of the initial set  $\Psi$ . Although we do not have a procedure to automatically find the best possible set of directions, the above heuristic choice gave good results in the experiments. The issue of choosing  $\text{Dirs}$  is similar to the issue of choosing the template directions of polytopes in template based reachability analysis [9, 16, 35].

---

**Algorithm 3:** Directional bounds on reachable sets

---

**Input:** Nonlinear system  $f : \Gamma \times U \rightarrow \Gamma$  where  $U$  is a box. A box  $\Psi \subseteq \Gamma$ ,  $\delta^{\max} \in [0, \infty)$ ,  $H \in \mathbb{Q}^{P \times n}$ ,  $N \in \mathbb{Z}_{\geq 1}$ .  
**Output:**  $\forall i \in \{1, \dots, N\}$   $h^i \in \mathbb{Q}^P$  :  
 $\max_{x \in \mathcal{R}_f(\Psi, [(i-1)\delta^{\max}, i\delta^{\max}])} Hx \leq h^i$ .

```

1 Choose finite set of direction vectors  $Dirs \subseteq \mathbb{Q}^n$ 
2 Choose small time step  $\tau \in [0, \delta) : \exists M \in \mathbb{Z}_{\geq 1} \tau = \delta^{\max}/M$ .
3 Choose threshold number of intersections  $Maxintrs \in \mathbb{Z}_{>1}$  and
   order of zonotope  $l \in \mathbb{Z}_{\geq 1}$ .
4  $Z^{init} \leftarrow \mathcal{Z}_f(\Psi, [0, \tau])$ .
5  $h^1 \leftarrow H \odot Z^{init}$ .
6  $\Omega \leftarrow \text{Boxapp}(Z^{init})$ .
7  $Z \leftarrow \bigcap_{\alpha \in Dirs} \bigcup_{X \in \text{optdivs}(f, \Omega, \alpha, \eta)} X$ 
8  $t \leftarrow \tau$ .
9 for  $i \in \{1, \dots, N\}$  do
10    $Z \leftarrow Z \cap \left( \bigcap_{\alpha \in Dirs} \bigcup_{X \in \text{optdivs}(f, \Omega, \alpha, \eta)} X \right)$ .
11   for  $j \in \{1, \dots, \text{rows}(Z)\}$  In Parallel do
12     for  $k \in \{1, \dots, \text{cols}(Z)\}$  In Parallel do
13        $Z_{ij} \leftarrow \text{ReduceOrder}(Z_{ij}, l)$ .
14        $\mathcal{L} \leftarrow \text{Linearize}(f, \text{Boxapp}(Z_{ij}) \cap \Omega)$ .
15        $Z_{ij} \leftarrow \mathcal{Z}_{\mathcal{L}}(Z_{ij}, \tau)$ 
16        $S_{ij} \leftarrow \text{Boxapp}(Z_{ij})$ .
17        $b_{ij} \leftarrow H \odot Z_{ij}$ .
18     end for
19   end for
20    $t \leftarrow t + \tau$ .
21    $\Omega \leftarrow \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} S_{ij}$ .
22   if  $t > i\delta^{\max}$  then
23      $i \leftarrow i + 1$ .
24      $h^i \leftarrow \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} b_{ij}$ .
25   else
26      $h^i \leftarrow h^i \vee \left( \bigwedge_{i=1}^{\text{rows}(Z)} \bigvee_{j=1}^{\text{cols}(Z)} b_{ij} \right)$ .
27   end if
28    $Z \leftarrow \text{ReduceInter}(Z, Maxintrs)$ .
29 end if
30 end for

```

---

*Complexity.* The number of elementary arithmetic operations required in computing reachable set of an order  $l$  zonotope after small time elapse in an  $n$ -dimensional linear system is  $O(n^3l)$ . The number of arithmetic operations required to compute bounds on a zonotope is  $O(n^2l)$ . The number of arithmetic operations required to optimize the Taylor error in Algorithm 1 is  $O(n\eta \text{len}(\nabla(\nabla f)))$  where  $\text{len}(\nabla(\nabla f))$  is the length of the symbolic formula of the Hessian of  $f$ . We consider that rational numbers are soundly over-approximated by floating point interval bounds of bounded precision using interval arithmetic. Then given  $\omega$  processors, the time complexity of our parallel algorithm is

$$O\left(N \frac{(n^3l + nm + nm + \eta n \text{len}(\nabla(\nabla f))) 2^\eta (Maxintrs + |dirs|)}{\omega}\right) \quad (16)$$

We can set the user defined parameters  $\eta$  and  $Maxintrs$  to tune the computation speed. When  $\eta$  and  $Maxintrs$  are fixed, the above complexity is polynomial time in dimension, number of inputs, order of zonotope and number of time steps.

## 6 EVALUATION

We performed experiments on three high-dimensional nonlinear models of real-world system having highly nonlinear differential equations containing trigonometric functions, inverse functions, and polynomials. The models are given below.

### 6.1 Models

*2-Car platoon.* We consider a 12-dimensional model of the platoon of 2 autonomous cars given in the previous Example 5.1. We consider the following initial set in S.I. units:

$$\begin{aligned} & [-1, 1] \times [-0.5, 0.5] \times [8, 9] \times [-0.3, 0.3] \\ & \times [-0.2, 0.2] \times [-0.3, 0.3] \times [-25, -25] \times [-0.1, 0.1] \\ & \times [5, 9] \times [-0.05, 0.05] \times [-0.1, 0.1] \times [-0.05, 0.05], \end{aligned}$$

and an input set  $[-0.01, 0.01]^2$ . The dynamics of first and second car are coupled by the function  $f_{10}(x, y, u) = K_1(x_3 - y_3)$ .

*Unicycle platoon.* Let us consider a platoon of unicycle vehicles where the  $i^{th}$  vehicle follows the  $(i-1)^{th}$  vehicle and the relative displacements between the vehicles are controlled by a state-dependent feedback. It is modeled by a tuple  $(f, U, \mathbb{R}^{3l})$  where  $l$  is the number of vehicles,  $U = \{5\}$  is the input velocity of the leader vehicle, and  $f : \mathbb{R}^{3l} \times U \rightarrow \mathbb{R}^{3l}$  is the vector field given by the following equations. Below,  $x \in \mathbb{R}^{3l}$  is the state vector and  $u \in U$  is the input. Let  $i \in \mathbb{Z}_{\geq 0}$  denote the index of a vehicle in the platoon, where  $i = 0$  denotes the leading vehicle and  $i = l-1$  denotes the last vehicle in the platoon.

if  $i = 0$ , then

$$\begin{aligned} f_{3i+1}(x, u) &= u \cos(x_3) / \left(1 + x_3^2\right) \\ f_{3i+2}(x, u) &= u \sin(x_3) / \left(1 + x_3^2\right) \\ f_{3i+3}(x, u) &= -x_3 \left(0.5 + 2x_3^2\right). \end{aligned}$$

if  $0 < i < l$ , then

$$\begin{aligned} f_{3i+1}(x, u) &= \frac{0.3(x_{3i-2} - x_{3i+1} + 10) \cos(x_{3i+3})}{1 + x_{3i+3}^2} \\ f_{3i+2}(x, u) &= \frac{0.3(x_{3i-2} - x_{3i+1} + 10) \sin(x_{3i+3})}{1 + x_{3i+3}^2} \\ f_{3i+3}(x, u) &= -x_{3i+3} \left(0.5 + 2x_{3i+3}^2\right). \end{aligned}$$

Above,  $x_{3i+1}$  is the displacement along  $X$ -axis of the  $(i+1)^{th}$  vehicle,  $x_{3i+2}$  is the displacement along  $Y$ -axis of the  $(i+1)^{th}$  vehicle and  $x_{3i+3}$  is its orientation angle with  $X$ -axis. The leading vehicle moves with a bounded time-varying speed  $u$ . The other vehicles follow the leading vehicle by applying a state-dependent feedback control, which results in the above dynamics. We consider 4 vehicles, i.e.,  $l = 4$ . We are given an initial set of states specified by the interval vector  $\Psi$  where

$$\begin{aligned} \Psi_{1:3} &= [[60, 70] \quad [-0.1, 0.1] \quad [-0.5, 0.5]]^T, \\ \Psi_{4:6} &= [[40, 50] \quad [-0.1, 0.1] \quad [-0.5, 0.5]]^T, \\ \Psi_{7:9} &= [[20, 30] \quad [-0.1, 0.1] \quad [-0.5, 0.5]]^T \end{aligned}$$

and  $\Psi_{10:12} = [[0, 10] \quad [-0.1, 0.1] \quad [-0.5, 0.5]]^T$ .

The input set is  $\{5.0\}$ . All units are S.I.

**Table 1: Computation times and hyperparameters**

2-Car Platoon		Quadrotor		Unicycle Platoon	
Method	CT (s)	Method	CT (s)	Method	CT (s)
IoU (p) $l : 50, \tau : 0.005, \eta : 2$	224	IoU (p) $l : 200, \tau : 0.025, \eta : 3$	196	IoU (p) $l : 50, \tau : 0.01, \eta : 1$	65
IoU $l : 50, \tau : 0.005, \eta : 1$	118	IoU $l : 200, \tau : 0.02, \eta : 2$	94	IoU $l : 50, \tau : 0.01, \eta : 2$	193
IoU $l : 50, \tau : 0.005, \eta : 3$	457	IoU $l : 200, \tau : 0.025, \eta : 4$	457	IoU $l : 50, \tau : 0.01, \eta : 3$	367
IoU $l : 25, \tau : 0.005, \eta : 1$	82	IoU $l : 50, \tau : 0.02, \eta : 3$	136	IoU $l : 25, \tau : 0.01, \eta : 1$	43
IoU $l : 10, \tau : 0.005, \eta : 1$	62	IoU $l : 25, \tau : 0.02, \eta : 3$	82	IoU $l : 10, \tau : 0.01, \eta : 1$	29
SO $l : 50, \tau : 0.005, \eta : 2$	225	SO $l : 100, \tau : 0.02, \eta : 3$	(IC) 10	SO $l : 50, \tau : 0.01, \eta : 1$	(IC) 2
PZ $l : 100, \tau : 0.005, DGO : 8$	123	PZ: $l : 150, \tau : 0.01, DGO : 200$	403	PZ $l : 200, \tau : 0.0025, DGO : 8$	(IC) 340
CL $l : 100, \tau : 0.001$	124	CL: $l : 1000, \tau : 0.0025$	170	CL: $l : 100, \tau : 0.001$	(IC) 28
TM EO : 4, $\tau : 0.001, RE : 0.1$	(IC) 168	TM EO : 4, $\tau : 0.005, RE = 0.1$	(IC) 223	TM: EO : 4, $\tau : 0.005, RE : 0.1$	86

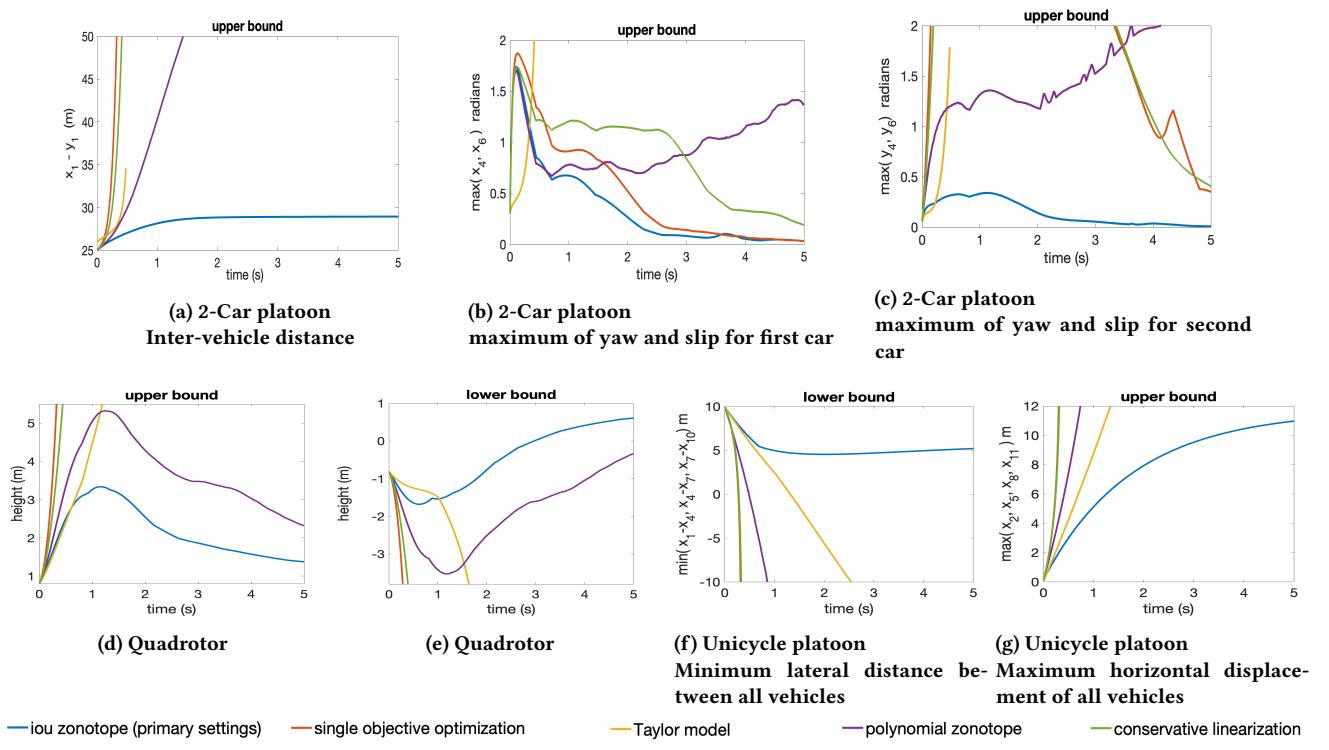
IoU: IoU algorithm using multiple solutions of multi-objective optimization,

SO: Variant of IoU algorithm using single type of division from optimization of (17)

PZ: Polynomial zonotope, CL: Conservative linearization, TM: Taylor model,  $l$ : Zonotope order, DGO: Dependent zonotope order

$\tau$ : Time step,  $\eta$ :  $\log_2$  (no. divisions per union), RE: Remainder estimation  $Maxintrs = 12$ ,  $Dimension = 12$ ,  $\epsilon = 1e - 10$ ,  $\kappa = 1000$ .

IC: Incomplete due to reach set explosion.

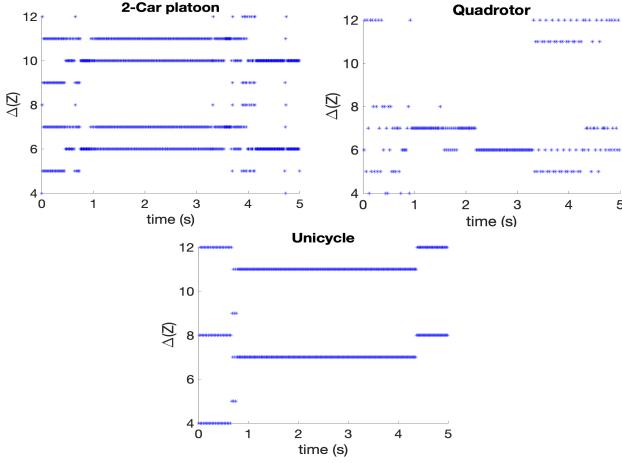


**Figure 1: Comparison of different algorithms**

*Quadrotor.* We consider the 12-dimensional model of a quadrotor with three inputs presented in the ARCH competition [19]. It has a 12-dimensional state vector  $x = (p_n, p_e, h, b, v, w, \phi, \theta, \psi, p, q, r)$ , where  $h$  is the height of the quadrotor and a 3-dimensional input vector  $u = (u_1, u_2, u_3)$ . The dynamics is given below:

$$\begin{aligned} \dot{p}_n &= b\cos(\phi)\cos(\theta) - v(\cos(\phi)\sin(\psi) - \\ &\quad \cos(\psi)\sin(\phi)\sin(\theta)) + w(\sin(\phi)\sin(\psi) + \end{aligned}$$

$$\begin{aligned} &\cos(\phi)\cos(\psi)\sin(\theta)) \\ \dot{p}_e &= v(\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\psi)\sin(\theta)) + \\ &\quad b\cos(\theta)\sin(\psi) - w(\cos(\psi)\sin(\phi) - \\ &\quad \cos(\phi)\sin(\psi)\sin(\theta)) \\ \dot{h} &= b\sin(\theta) - w\cos(\phi)\cos(\theta) - v\cos(\theta)\sin(\phi) \\ \dot{b} &= rv - qw - g\sin(\theta) \end{aligned}$$



**Figure 2: Minimum number of intersections of IoU zonotope required at each time step for greedily minimizing the relative IoU measure (see Equation 18 and Algorithm 2)**

$$\begin{aligned}
\dot{v} &= pw - rb + g \cos(\theta) \sin(\phi) \\
\dot{w} &= qb - pv + g \cos(\phi) \cos(\theta) - \frac{mg - 10(h - u_1) + 3w}{m} \\
\dot{\phi} &= p + r \cos(\phi) \sin(\theta) / \cos(\theta) + q \sin(\phi) \sin(\theta) / \cos(\theta) \\
\dot{\theta} &= q \cos(\phi) - r \sin(\phi) \\
\dot{\psi} &= r \cos(\phi) / \cos(\theta) + q \sin(\phi) / \cos(\theta) \\
\dot{p} &= (-(\phi - u_2) - p) / Jx + qr(Jy - Jz) / Jx \\
\dot{q} &= (-(\theta - u_3) - q) / Jy + pr(Jx - Jz) / Jy \\
\dot{r} &= 0 / Jz + pq(Jx - Jy) / Jz.
\end{aligned}$$

We took a larger initial set than the one given in the competition [19] so that there is significant linearization error in the flowpipe. Our initial set in S.I. units is  $[-0.8, 0.8]^6 \times [-0.5, 0.5]^2 \times [0, 0] \times [-1, 1]^2 \times [0, 0]$ . The input set is  $(u_1, u_2, u_3) \in [-0.99, 1.01] \times [-0.001, 0.001]^2$ . Notice that the angles in the initial set are larger than that given in the ARCH competition. Therefore, the plots, computation times and hyperparameters are different from the ARCH competition.

## 6.2 Experiments

We did the experiments in the following sections to demonstrate the advantage of using IoU zonotope based algorithm. The primary hyperparameters our algorithm used in the implementation are given in the first row of Table 1 labeled (p). These parameters are changed depending on the experiment. We chose the set of directions *Dirs* along which linearization error is optimized as the collection of co-ordinate axes directions and the rows of Jacobian with respect to the state variables at the midpoint of the initial set.

*Computing platform.* Our IoU algorithm and CORA tool were run on a Macbook Air with M1 chip having 8 CPU cores and 8GB RAM. We used all 8 cores for parallel processing in our algorithm. The Flowstar algorithm was run on as AWS tc2 large computer with 8GB RAM.

**6.2.1 Multiple types of division vs single type of division.** We want to show the advantage of using multiple types of optimized divisions, instead of a single type of division resulting from optimizing

a reasonable scalar objective function. So, we considered the following single objective function which is the infinity norm of the normalized upper bound on the offset of linearization error. Below,  $\kappa(\mu)$  is the objective function of the division vector  $\mu$ ,  $f$  is the nonlinear system and  $\Psi$  is the box being divided:

$$\kappa(\mu) = \max_{i=1}^n \left\| \frac{\text{offset}(\Delta_{f,\Psi}(\mu))_i}{\text{offset}(\Delta_{f,\Psi}([1]_{n \times 1}))_i + 1e-10} \right\|. \quad (17)$$

We implemented a reachability analysis algorithm where we optimized the above objective (17) in Algorithm 1 and replaced the resulting single type of division at each time step in Algorithm 3. We compared the accuracy of both algorithms under the primary settings (see Table 1 row 1). The approximation of reachable sets is given in Figure 1. The reachable set of the variant of our algorithm which optimizes the scalar objective function is plotted with the legend "single objective". There is not much difference in computation speed as shown in Table 1. But there is large increase in accuracy with our proposed approach using multiple optimized divisions compared to the one using only one optimized division at each time step. The plots are shown in Figure 1. The reason is that a single type of division may not minimize linearization error along multiple directions. But our multiple-type division approach optimizes linearization error along multiple projection directions. Since the differential equations are coupled, the linearization error may need to be minimized along multiple directions to get good approximation accuracy of reachable set along any direction.

**6.2.2 Intersecting more unions increases accuracy.** We want to show that increasing number of intersections in the IoU gives more accurate approximation of the reachable set. So, at each time step of our algorithm, we plotted the smallest size of the IoU obtained from reducing intersections using Algorithm 2, such that the over-approximation of box hull of the reduced IoU is the same as the over-approximation of box hull of the original IoU. This number is defined below for IoU zonotope  $Z$ .

$$\Delta(Z) = \min \left\{ k \in \{1, \dots, \text{cols}(Z)\} \mid \text{Boxapp}(\text{ReduceInter}(Z, k)) = \text{Boxapp}(Z) \right\} \quad (18)$$

If the above number at a time step is greater than one, it means that we need more than one zonotope to reduce the relative IoU measure given in (15). As shown in Figure 2, this number is typically greater than 5 at most of the time steps. So, increasing the number of intersections of the IoU zonotope reachable set increases the accuracy of approximating the reachable set.

**6.2.3 Changing number of divisions and zonotope order.** We performed simulations by changing the number of divisions per union and the order of zonotope. As expected, we found a significant increase in accuracy by increasing the number of divisions on all models. This is because increasing divisions reduces the linearization error. There is also increase in accuracy by increasing the zonotope order. The plots are shown in the Figure 3. But increasing divisions and zonotope order can increase computation time according the complexity measure in (16). The increase in computation time is noted in Table 1.

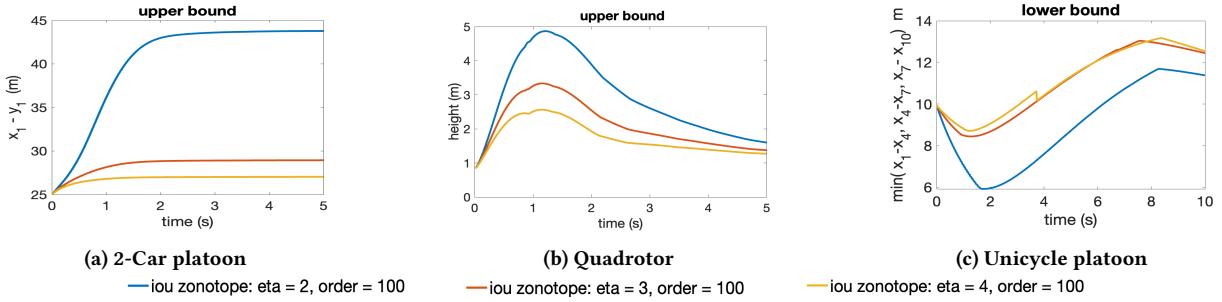


Figure 3: Changing the number of divisions per union

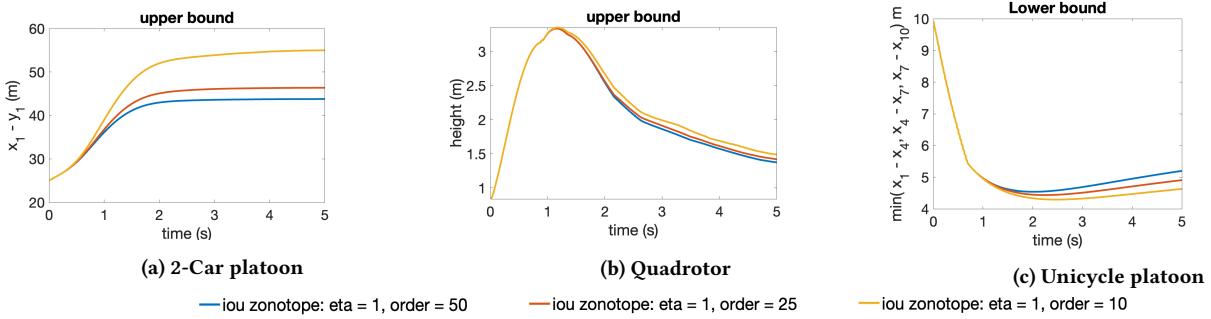


Figure 4: Changing the order of zonotope

**6.2.4 Comparison with other methods.** We compare our approach with three state of the art algorithms, i.e., (i) Conservative linearization [5] which minimizes linearization error by splitting reachable set into unions of zonotopes (not intersection of unions), (ii) Polynomial zonotopes [1], and (iii) Taylor models [13]. We implemented conservative linearization and polynomial zonotopes in CORA tool [2], while Taylor models are implemented in Flowstar tool [12]. The performance of CORA and Flowstar can be sensitive to hyperparameter setting. We note that it can take unreasonably long time to find the best performing hyperparameters of these tools using hyperparameter optimization. Therefore, we manually tuned the hyperparameters to increase accuracy while ensuring that the computation time is close to that of our IoU algorithm. Therefore, we manually tuned some of these hyperparameters, like the zonotope order, Taylor expansion order, dependent zonotope order, remainder estimation threshold and time step. We set the rest of the hyperparameters as those used in the ARCH competition [19]. Some of the tuned hyperparameter settings for Flowstar and CORA are given in Table 1.

As shown in Figures 1, the accuracy of our IoU algorithm is much higher than the other tools for comparable computation times. This establishes significant progress in the state-of-the-art of reachability analysis by our proposed method.

## 7 CONCLUSION

Dividing reachable set to restrict the linearization error below a threshold can become computationally intractable in high dimensional spaces. Alternatively, we can fix the number of pieces and then find a good division of the reach set that minimizes the

linearization error. However, the linearization error being multi-dimensional renders the optimization problem to be multi-objective, which has no single best solution. We proposed a new approach to piecewise linearization that leverages different types of divisions resulting from optimization of multiple different projections of linearization error. Our solution uses intersection of unions of zonotopes, where each intersecting union corresponds to an optimized division for each projection direction for the linearization error. We compared our algorithm with an alternative algorithm which optimizes a single objective function resulting in a single type of division. Our experiments demonstrated that intersecting multiple different optimized divisions can give better accuracy than using a single type of division. Furthermore, evaluation of this method on real world examples showed high increase in accuracy compared to state-of-the-art techniques.

An important direction for future research is extending this technique to polynomialization, where we try to minimize the error in polynomial approximation of non-linear systems along different directions using intersection of unions. In this case, we could possibly use polynomial zonotopes or Taylor models as constituent sets of the IoU representation. Furthermore, extending this approach to handle hybrid dynamics with switching conditions can require more sophisticated analysis.

## ACKNOWLEDGMENTS

The authors would like to thank Marcelo Forets for helping in using the LazySets [20] Julia package to perform some of the experiments. This work is supported by Science and Engineering Research Board, India under the NPDF Fellowship PDF/2020/000895/ES.

## REFERENCES

- [1] Matthias Althoff. 2013. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*. 173–182.
- [2] Matthias Althoff. 2016. CORA 2016 manual. *TU Munich* 85748 (2016).
- [3] Matthias Althoff and Goran Frehse. 2016. Combining zonotopes and support functions for efficient reachability analysis of linear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 7439–7446.
- [4] Matthias Althoff and Bruce H Krogh. 2011. Zonotope bundles for the efficient computation of reachable sets. In *2011 50th IEEE conference on decision and control and European control conference*. IEEE, 6814–6821.
- [5] Matthias Althoff, Olaf Stursberg, and Martin Buss. 2008. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *2008 47th IEEE Conference on Decision and Control*. IEEE, 4042–4048.
- [6] Stanley Bak, Sergiy Bogomolov, Thomas A Henzinger, Taylor T Johnson, and Pradyot Prakash. 2016. Scalable static hybridization methods for analysis of nonlinear systems. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. 155–164.
- [7] Stanley Bak and Parasara Sridhar Duggirala. 2017. Simulation-equivalent reachability of large linear systems with inputs. In *International Conference on Computer Aided Verification*. Springer, 401–420.
- [8] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Frédéric Viry, Andreas Podelski, and Christian Schilling. 2018. Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*. 41–50.
- [9] Sergiy Bogomolov, Goran Frehse, Mirco Giacobbe, and Thomas A Henzinger. 2017. Counterexample-guided refinement of template polyhedra. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 589–606.
- [10] Hervé Brönnimann, Guillaume Melquiond, and Sylvain Pion. 2006. The design of the Boost interval arithmetic library. *Theoretical Computer Science* 351, 1 (2006), 111–118.
- [11] Mo Chen, Sylvia L Herbert, Mahesh S Vashishtha, Somil Bansal, and Claire J Tomlin. 2018. Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Trans. Automat. Control* 63, 11 (2018), 3675–3688.
- [12] Xin Chen. 2013. User’s Manual of Flow v1.2.0. <https://flowstar.org/publications/>
- [13] Xin Chen, Erika Abraham, and Sriram Sankaranarayanan. 2012. Taylor model flowpipe construction for non-linear hybrid systems. In *2012 IEEE 33rd Real-Time Systems Symposium*. IEEE, 183–192.
- [14] Xin Chen and Sriram Sankaranarayanan. 2016. Decomposed reachability analysis for nonlinear systems. In *2016 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 13–24.
- [15] Christophe Combastel. 2003. A state bounding observer based on zonotopes. In *2003 European Control Conference (ECC)*. IEEE, 2589–2594.
- [16] Thao Dang and Thomas Martin Gawlitz. 2011. Template-based unbounded time verification of affine hybrid automata. In *Asian Symposium on Programming Languages and Systems*. Springer, 34–49.
- [17] Thao Dang, Oded Maler, and Romain Testylier. 2010. Accurate hybridization of nonlinear systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*. 11–20.
- [18] Thao Dang and David Salinas. 2009. Image computation for polynomial dynamical systems using the Bernstein expansion. In *International Conference on Computer Aided Verification*. Springer, 219–232.
- [19] Gidon Ernst, Paolo Arcaini, Ismail Bennani, Alexandre Donze, Georgios Fainekos, Goran Frehse, Logan Matheson, Claudio Menghi, Giulia Pedrinelli, Marc Pouzet, et al. 2020. Arch-comp 2020 category report: Falsification. *EPiC Series in Computing* (2020).
- [20] Marcelo Forets and Christian Schilling. 2021. LazySets.jl: Scalable Symbolic-Numeric Set Computations. *arXiv preprint arXiv:2110.01711* (2021).
- [21] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. 2011. SpaceEx: Scalable Verification of Hybrid Systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV) (LNCS)*. Springer.
- [22] Antoine Girard. 2005. Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 291–305.
- [23] Antoine Girard and Colas Le Guernic. 2008. Efficient reachability analysis for linear systems using support functions. *IFAC Proceedings Volumes* 41, 2 (2008), 8966–8971.
- [24] Zhi Han and Bruce H Krogh. 2006. Reachability analysis of nonlinear systems using trajectory piecewise linearized models. In *2006 American Control Conference*. IEEE, 6–pp.
- [25] Niklas Kochdumper and Matthias Althoff. 2020. Constrained polynomial zonotopes. *arXiv preprint arXiv:2005.08849* (2020).
- [26] Niklas Kochdumper and Matthias Althoff. 2020. Sparse polynomial zonotopes: A novel set representation for reachability analysis. *IEEE Trans. Automat. Control* (2020).
- [27] Abolfazl Lavaei, Fabio Somenzi, Sadegh Soudjani, Ashutosh Trivedi, and Majid Zamani. 2020. Formal controller synthesis for continuous-space MDPs via model-free reinforcement learning. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPs)*. IEEE, 98–107.
- [28] Dongxu Li, Stanley Bak, and Sergiy Bogomolov. 2020. Reachability Analysis of Nonlinear Systems Using Hybridization and Dynamics Scaling. In *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 265–282.
- [29] John Maidens and Murat Arcak. 2014. Reachability analysis of nonlinear systems using matrix measures. *IEEE Trans. Automat. Control* 60, 1 (2014), 265–270.
- [30] David Monniaux and Pierre Corbineau. 2011. On the generation of Positivstellensatz witnesses in degenerate cases. In *International Conference on Interactive Theorem Proving*. Springer, 249–264.
- [31] Katta G Murty and Santosh N Kabadi. 1985. *Some NP-complete problems in quadratic and nonlinear programming*. Technical Report.
- [32] Stephen Prajna. 2006. Barrier certificates for nonlinear model validation. *Automatica* 42, 1 (2006), 117–126.
- [33] Nacim Ramdani, Nacim Meslem, and Yves Candau. 2009. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Trans. Automat. Control* 54, 10 (2009), 2352–2364.
- [34] Enric Rodriguez-Carbonell and Ashish Tiwari. 2005. Generating polynomial invariants for hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 590–605.
- [35] Sriram Sankaranarayanan, Thao Dang, and Franjo Ivančić. 2008. Symbolic model checking of hybrid systems using template polyhedra. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 188–202.
- [36] Romain Testylier and Thao Dang. 2013. Nltoolbox: A library for reachability computation of nonlinear dynamical systems. In *Automated Technology for Verification and Analysis*. Springer, 469–473.
- [37] Ashish Tiwari. 2008. Generating box invariants. In *International Workshop on Hybrid Systems: Computation and Control*. Springer, 658–661.
- [38] Xuejiao Yang and Joseph K Scott. 2018. A comparison of zonotope order reduction techniques. *Automatica* 95 (2018), 378–384.