Reachability Analysis of Nonlinear Systems using Conservative Polynomialization and Non-Convex Sets

Matthias Althoff
Ilmenau University of Technology
98693 Ilmenau, Germany
matthias.althoff@tu-ilmenau.de

ABSTRACT

A new technique for computing the reachable set of hybrid systems with nonlinear continuous dynamics is presented. Previous work showed that abstracting the nonlinear continuous dynamics to linear differential inclusions results in a scalable approach for reachability analysis. However, when the abstraction becomes inaccurate, linearization techniques require splitting of reachable sets, resulting in an exponential growth of required linearizations. In this work, the nonlinearity of the dynamics is more accurately abstracted to polynomial difference inclusions. As a consequence, it is no longer guaranteed that reachable sets of consecutive time steps are mapped to convex sets as typically used in previous works. Thus, a non-convex set representation is developed in order to better capture the nonlinear dynamics, requiring no or much less splitting. The new approach has polynomial complexity with respect to the number of continuous state variables when splitting can be avoided and is thus promising when a linearization technique requires splitting for the same problem. The benefits are presented by numerical examples.

Categories and Subject Descriptors

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

Keywords

Reachability Analysis, Hybrid Systems, Nonlinear Dynamics, Difference Inclusion, Polynomial Zonotopes

1. INTRODUCTION

Formal verification of hybrid systems has enormous practical relevance since in almost all engineering fields, complex systems have a mixed discrete/continuous dynamics due to the interplay of physical behavior and digital control. Those systems are difficult to analyze, especially since disturbances and other uncertainties can lead to completely different behaviors. For this reason, formal verification techniques have been developed in order to mathematically guarantee that a system model satisfies a formalized specification.

Over the past years, a variety of formal methods for hybrid system verification have been developed: Reachability analysis [6], theorem proving [33], barrier certificates [34], simulation-based verification [23], abstraction to discrete systems [14], constraint propagation [35], and many more. Related topics are the falsification of systems [27,38], i.e. finding solutions that violate a specification, and probabilistic model checking [10,15], where a probability for satisfying a specification is computed. This work is about reachability analysis so that the remainder of the literature review is on this topic. Reachability analysis is concerned with the problem of computing the set of discrete and continuous states that a system can reach, making it possible to verify if a state can avoid a set of unsafe states.

Early works on reachability analysis considered timed automata, i.e., automata with time as the only continuous variable [5,8]. This concept has been extended to linear hybrid automata, where in each discrete mode, the derivative of the continuous state vector is bounded by a hyperrectangle [11] or a polytope [19]. Linear automata can also be used to overapproximate the solutions of more complex systems, such as linear continuous systems ($\dot{x} = Ax(t) + u(t)$, $A \in \mathbb{R}^{n \times n}$, $x, u \in \mathbb{R}^n$) [24]. However, this approach causes a wrapping effect, i.e., the overapproximation increases since reachable sets are computed based on sets of previous times and the error accumulates for each iteration. A wrapping-free approach for linear systems has first been published in [22]. When using special set representations such as zonotopes or support functions, linear systems with more than 1000 continuous state variables can be verified [21,22].

For hybrid systems with nonlinear continuous dynamics, no wrapping-free algorithm exist, except when the dynamics can be rewritten as a linear system in a new coordinate system [37]. Approaches for nonlinear systems can be categorized into approaches that (1) reformulate the reachability analysis as an optimization problem, (2) use the Picard iteration in combination with Taylor models, or (3) construct mappings to propagate the set of reachable states. Reformulation as an optimization problem is performed in [13] and [31], where the first approach uses optimization to obtain the outwards translation of halfspaces confining polytopical reachable sets and the latter rewrites the entire reachable set computation in the form of Hamilton-Jacobi equations.

HSCC'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.

Typically, approaches involving optimization techniques do not scale well with the number of continuous state variables.

The Picard iteration in combination with rigorous Taylor models is first proposed by Berz and Makino and adopted by other researchers [25, 30, 32]. The main idea is to use the Picard iteration to obtain a polynomial of given degree that approximates the solution over time with respect to varying initial states and other parameters. In order to guarantee that the exact solution is captured, an uncertain multidimensional interval is added to the polynomial solution, resulting in a so-called *Taylor model*. A Taylor model is acceptable if it is contractive, i.e., running the Picard iteration with the suggested Taylor model has to result in a Taylor model enclosed by the previous one. The approach is extended to hybrid systems in [12].

Construction of mappings for the propagation of reachable sets is well developed for linear systems with a large variety of convex set representations (polytopes [13], zonotopes [22], ellipsoids [29], support functions [21], oriented hyperrectangles [39], and others). Those approaches do not require to formulate solutions over time by polynomials as done for Taylor model approaches. When the dynamics is nonlinear, most approaches linearize the dynamics. Earlier approaches define regions in which the original dynamics is linearized to which a compensating linearization error is added, resulting in linear differential inclusions [7]. When the nonlinear dynamics is a multi-affine system and the regions are hyperrectangles, it is sufficient to only consider the flow field at the vertices to determine which cells of the partition are reachable [28]. The disadvantage of fixed partitions is that the number of required regions grows exponentially with respect to the number of continuous state variables and most approaches require intersection at the borders of the regions, which is computationally expensive. More recent approaches overcome this problem by defining overlapping linearization regions that move along with the reachable set [4, 18]. The main disadvantage of those approaches is that for large linearization errors, the reachable set has to be split and one has to continue with several reachable sets in parallel, causing exponential complexity in the number of variables contributing to the linearization error.

In this work, the problem of avoiding or at least reducing splitting of map-based reachable set approaches is addressed. Instead of applying linearization, the system dynamics is abstracted to a nonlinear system with polynomial right-hand side plus additive uncertainty, resulting in *polynomial differential inclusions*. Unlike for linear systems, there exists no closed-form solution for polynomial differential inclusions, but a new tight overapproximation is presented in the form of a *polynomial difference inclusion* $x(t_{k+1}) \in f(x(t_k), u(t_k)) \oplus \mathcal{W}$, where $\mathcal{W}$ is an additive uncertainty with proper dimension and $\oplus$ denotes the Minkowski addition. Thus, reachable sets which are represented by a convex set, are now possibly mapped to non-convex sets. If one uses a convex set representation as done by almost all previous approaches, the benefit of capturing the nonlinearity by a polynomial differential inclusion is lost. For this reason, a new non-convex set representation called *polynomial zonotope* is proposed, which extends the definition of zonotopes, and is as expressive as Taylor models. We demonstrate the approach by numerical examples, compare the results to the Taylor model tool *flow** [12], and show the benefits compared to map-based linearization procedures. Since the overall complexity of the new approach is polynomial in the number of continuous state variables, it is preferable over linearization techniques for many problems which require splitting.

In order to focus on the continuous dynamics of hybrid systems, the approach is first developed for purely continuous systems. In Sec. 3, the solution of a nonlinear differential equation is overapproximated by a polynomial difference inclusion. The abstraction is used in Sec. 4 to develop the reachability algorithm. Polynomial zonotopes are introduced as the set representation in Sec. 5. The paper finishes with numerical examples in Sec. 6, where one example also briefly presents the integration of the approach into hybrid system reachability analysis.

2. BASIC PRINCIPLE

In this paper, nonlinear systems of the form

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

are considered, where x is the state vector and u is the input vector. The differential equation is required to be Lipschitz continuous and the input trajectory $u(\cdot)$ is required to be piecewise continuous so that a solution is guaranteed to exist. Let $\chi(t; x_0, u(\cdot))$ denote the solution to (1) for an initial state $x(0) = x_0$ and the input trajectory $u(\cdot)$. For a set of initial states $\mathcal{R}(0) \subset \mathbb{R}^n$ and a set of possible input values $\mathcal{U} \subset \mathbb{R}^m$, the set of reachable states is

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

The superscript e on $\mathcal{R}^e([0,r])$ denotes the exact reachable set, which cannot be computed for general nonlinear systems. Therefore, an overapproximation $\mathcal{R}([0,r]) \supseteq \mathcal{R}^e([0,r])$ is computed as accurately as possible, while at the same time ensuring that the computations are efficient and scale well with the system dimension n. As in many other works (see e.g. [13, 16, 20, 39]), the reachable set is computed for consecutive time intervals $\tau_s := [t_s, t_{s+1}]$, where $t_s = s \cdot r$, $r \in \mathbb{R}^+$ is the time increment, and $s \in \mathbb{N}$ is the time step. The reachable set for a user-defined time horizon t_f is $\mathcal{R}([0,t_f]) = \bigcup_{s=0}^{t_f/r-1} \mathcal{R}(\tau_s)$, where t_f is a multiple of r.

As later shown, the wrapping effect in this work is almost entirely determined by the accuracy of auxiliary sets $\mathcal{R}(t_s)$ at points in time t_s since those sets are computed based on the accuracy of the previous ones. In contrast to this, the sets of time intervals $\mathcal{R}(\tau_s)$ fill the "time gaps" based on the sets $\mathcal{R}(t_s)$ as shown in Fig. 1, and are not used again in the computation.

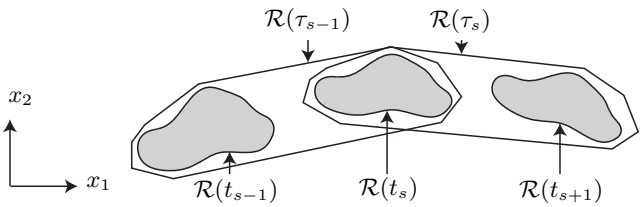

Figure 1: Stepwise computation of the reachable set.

3. ABSTRACTION OF THE NONLINEAR DYNAMICS

Nonlinear systems are hard to analyze since almost all of them do not have a closed-form solution. In this work, their solution is tightly overapproximated by first abstracting the nonlinear differential equations to a polynomial differential inclusion using a Taylor expansion. In a second step, a novel approach is used to obtain a polynomial difference equation.

3.1 Abstraction to Polynomial Differential Inclusions

For a concise notation, the combined state/input vector $\tilde{z} = \begin{bmatrix} x^T & u^T \end{bmatrix}^T \in \mathbb{R}^o$ ($o = n + m$) and the Nabla symbol $\nabla = \sum_{i=1}^{o} e^{(i)} \frac{\partial}{\partial \tilde{z}_i}$, using orthogonal unit vectors $e^{(i)}$, are introduced. Note that the superscript i is in parentheses to avoid confusion with powers, which is a notation used for other variables in this work, too. The nonlinear system in (1) is abstracted by a Taylor expansion of order κ at point z^* with Lagrange remainder $\mathcal{L}$ (see [9]):

$$\dot{x}_i = f_i(\hat{z}(t)) \in \sum_{\nu=0}^{\kappa} \left. \frac{\left((\hat{z}(t) - z^*)^T \nabla\right)^\nu f_i(\tilde{z})}{\nu!} \right|_{\tilde{z}=z^*} \oplus \mathcal{L}_i(t) \tag{2}$$

$$\mathcal{L}_i(t) = \left\{ \left. \frac{\left((\hat{z}(t) - z^*)^T \nabla\right)^{\kappa+1} f_i(\tilde{z})}{(\kappa+1)!} \right| \tilde{z} = z^* + \alpha(\hat{z}(t) - z^*),\ \alpha \in [0,1] \right\},$$

where $\mathcal{A} \oplus \mathcal{B} := \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$ is a Minkowski addition and for later derivations, set-based multiplication $\mathcal{A} \otimes \mathcal{B} := \{a\, b | a \in \mathcal{A}, b \in \mathcal{B}\}$ is introduced, too. Note that set-based multiplication has precedence over set-based multiplication, expressions are evaluated from left to right, and the symbol $\otimes$ is sometimes omitted as for classical multiplications. For subsequent derivations, the alternative notation of (2)

$$\begin{aligned} \dot{x}_i \in & w_i + \frac{1}{1!} \sum_{j=1}^{o} C_{ij} z_j(t) + \frac{1}{2!} \sum_{j=1}^{o} \sum_{k=1}^{o} D_{ijk} z_j(t) z_k(t) \\ & + \frac{1}{3!} \sum_{j=1}^{o} \sum_{k=1}^{o} \sum_{l=1}^{o} E_{ijkl} z_j(t) z_k(t) z_l(t) + \ldots \oplus \mathcal{L}_i(t) \end{aligned} \tag{3}$$

is used, where $z(t) = \hat{z}(t) - z^*$ and

$$w_i = f_i(z^*),\ C_{ij} = \left. \frac{\partial f_i(\tilde{z})}{\partial \tilde{z}_j} \right|_{\tilde{z}=z^*},\ D_{ijk} = \left. \frac{\partial^2 f_i(\tilde{z})}{\partial \tilde{z}_j \partial \tilde{z}_k} \right|_{\tilde{z}=z^*}, \ldots$$

Note that z^* is changed at times t_s so that we have $w(t_s)$, $C(t_s)$, $D(t_s)$, …, where the dependency on time is omitted in the notation since for the remainder of this section it is always assumed that $t \in \tau_s$. For reachability analysis we require a difference inclusion that encloses the solution of the differential inclusion in (3).

3.2 Abstraction to Difference Inclusions

As a first step, all higher order terms in (3) are interpreted as an input v to a linear system, where the matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are obtained from $C = \begin{bmatrix} A & B \end{bmatrix}$ and z is partially substituted by $\begin{bmatrix} x^T & u^T \end{bmatrix}^T$:

$$\begin{aligned} \dot{x} &\in Ax(t) + v(z(t), u(t)) \oplus \mathcal{L}(t) \\ v_i(z,u) &= w_i + \sum_{j=1}^{m} B_{ij} u_j + \frac{1}{2!} \sum_{j=1}^{o} \sum_{k=1}^{o} D_{ijk} z_j z_k + \ldots \end{aligned} \tag{4}$$

In order to obtain a tight overapproximation, the auxiliary variables $u^\Delta(t) = u(t) - u^c$, $z^\Delta(t) = z(t) - z(t_s)$ are introduced to split the input $v(z(t), u(t))$ for $t \in \tau_s$ into a constant part $v(z(t_s), u^c)$ fixed at the specific point in time t_s and a time-varying part $v^\Delta(z^\Delta(t), z(t_s), u^\Delta(t))$:

$$\begin{aligned} v_i(z(t), u(t)) &= w_i + \sum_{j=1}^{m} B_{ij}(u_j^c + u_j^\Delta(t)) \\ &+ \frac{1}{2!} \sum_{j=1}^{o} \sum_{k=1}^{o} D_{ijk} \underbrace{(z_j(t_s) + z_j^\Delta(t))(z_k(t_s) + z_k^\Delta(t))}_{\begin{array}{l} = z_j(t_s) z_k(t_s) + z_j(t_s) z_k^\Delta(t) \\ \quad + z_j^\Delta(t) z_k(t_s) + z_j^\Delta(t) z_k^\Delta(t) \end{array}} + \ldots \\ &= v(z(t_s), u^c) + v^\Delta(z^\Delta(t), z(t_s), u^\Delta(t)) \end{aligned}$$

where

$$\begin{aligned} v(z(t_s), u^c) &= w_i + \sum_{j=1}^{m} B_{ij} u_j^c + \\ &\quad \frac{1}{2!} \sum_{j=1}^{o} \sum_{k=1}^{o} D_{ijk} z_j(t_s) z_k(t_s) + \ldots \\ v^\Delta(z^\Delta(t), z(t_s), u^\Delta(t)) &= \sum_{j=1}^{m} B_{ij} u_j^\Delta(t) + \frac{1}{2!} \sum_{j=1}^{o} \sum_{k=1}^{o} D_{ijk} \\ &\quad \left(z_j(t_s) z_k^\Delta(t) + z_j^\Delta(t) z_k(t_s) + z_j^\Delta(t) z_k^\Delta(t) \right) + \ldots \end{aligned} \tag{5}$$

After defining $\mathcal{U}^\Delta := \mathcal{U} \oplus (-u^c)$ and assuming that the reachable set $\mathcal{R}(\tau_s)$ and

$$\begin{aligned} \mathcal{R}^\Delta(\tau_s) := \Big\{ & \chi(t; x(t_s), u(\cdot)) - x(t_s) \Big| t \in \tau_s, \\ & x(t_s) \in \mathcal{R}(t_s), \forall t \in \tau_s\, u(t) \in \mathcal{U} \Big\} \end{aligned} \tag{6}$$

are already known ($\chi()$ was defined as the solution of (1)), the set of possible values of $v^\Delta(z^\Delta(t), z(t_s), u^\Delta(t))$ is bounded by

$$\begin{aligned} \mathcal{V}^\Delta(\tau_s) := \Big\{ & v^\Delta(z^\Delta, z, u^\Delta) \Big| \\ & z^\Delta \in \mathcal{R}^\Delta(\tau_s) \times \mathcal{U}^\Delta, z \in \mathcal{R}(\tau_s) \times \mathcal{U},\ u^\Delta \in \mathcal{U}^\Delta \Big\}. \end{aligned} \tag{7}$$

Using (4) - (7), the linear differential inclusion

$$\dot{x} \in Ax(t) + v(z(t_s), u^c) \oplus \left(\mathcal{V}^\Delta(\tau_s) \oplus \mathcal{L}(\tau_s) \right)$$

is obtained for $t \in \tau_s$. Due to the superposition principle of linear systems, the solution is obtained by adding the solution of the homogeneous solution $x^h(t_{s+1})$, the input solution due to constant input $x^{p,c}(r)$, where $r = t_{s+1} - t_s$, and the input solution set due to time-varying inputs $\mathcal{R}^{p,\Delta}(\mathcal{V}^\Delta(\tau_s) \oplus \mathcal{L}(\tau_s), r)$ to

$$x(t_{s+1}) \in x^h(t_{s+1}) + x^{p,c}(t) \oplus \mathcal{R}^{p,\Delta}\left(\mathcal{V}^\Delta(\tau_s) \oplus \mathcal{L}(\tau_s), r\right). \tag{8}$$

The well-known homogeneous solution is $x^h(t_{s+1}) = e^{Ar}x(t_s)$, the input solution due to constant input is

$$x^{p,c}(r) = \Gamma(r)\, v(z(t_s), u^c), \quad \Gamma(r) := \int_0^r e^{A(r-t)}dt,$$

where $\Gamma(r) = A^{-1}(e^{Ar} - I)$ (I is the identity matrix) and when A is not invertible, the approach in [3] is used. The reachable set due to the set of uncertain time-varying inputs within $\tilde{\mathcal{V}}(\tau_s) := \mathcal{V}^{\Delta}(\tau_s) \oplus \mathcal{L}(\tau_s)$ is computed as in [3] as

$$\mathcal{R}^{p,\Delta}(\tilde{\mathcal{V}}(\tau_s), r) = \bigoplus_{i=0}^{\nu} \frac{t^{i+1}}{(i+1)!}\mathtt{CH}(A^i \otimes \tilde{\mathcal{V}}(\tau_s)) \oplus \mathcal{E}^{p,\Delta}, \quad (9)$$

where $\mathtt{CH}()$ is the operator returning the convex hull of a set, and $\mathcal{E}^{p,\Delta}$ is an interval vector which becomes arbitrarily small for $\nu \to \infty$.

The difference to previous approaches (e.g. [4, 18]) is that due to the separation in a constant and a time-varying input, nonlinear terms are saved from linearization at times t_s, while within τ_s, an abstracting linear differential inclusion is used. Inserting $v(z(t_s), u^c)$ from (5) into the overall solution (8) results in a nonlinear difference equation that encloses the exact solution:

$$\begin{aligned} x_i(t_{s+1}) \in &\sum_{j=1}^{n}(e^{Ar})_{ij}x_j(t_s) + \sum_{j=1}^{n}\Gamma_{ij}(r)\Big(w_j + \sum_{k=1}^{m}B_{jk}u_k^c \\ &+ \frac{1}{2!}\sum_{k=1}^{o}\sum_{l=1}^{o}D_{jkl}z_k(t_s)z_l(t_s) + \dots\Big) \\ &\oplus \mathcal{R}_i^{p,\Delta}(\mathcal{V}^{\Delta}(\tau_s) \oplus \mathcal{L}(\tau_s), r) \end{aligned} \quad (10)$$

The benefits of the above difference inclusion for reachability analysis do not immediately show. Note that for small time increments r, as typically used in reachability analysis, the set $\mathcal{R}_i^{p,\Delta}$ becomes small, no matter how large the set of $z(t_s)$ becomes during the reachability analysis (proof is omitted due to space limitations). Thus, for large sets of $z(t_s)$, the nonlinearity is well captured by all other terms, while the abstractions in $\mathcal{R}_i^{p,\Delta}$ are not dominant.

4. REACHABLE SET COMPUTATIONS

This section describes Alg. 1 for computing the set of reachable states when using the previously presented abstraction to difference inclusions. In order to focus on the novel aspects, the possibility to split reachable sets is not included. The algorithm consists of 2 parts as indicated in Alg. 1:

① Computing a linearization and the corresponding set of linearization errors denoted by $\Psi(\tau_s)$. The reachable set $\mathcal{R}(\tau_s)$ for the time interval τ_s is obtained as a by-product.

② Computing the reachable set at the next point in time $\mathcal{R}(t_{s+1})$ by a set-based evaluation of (10).

Since the computations of each time interval τ_s are based on the reachable set at points in time t_s, the sets $\mathcal{R}(t_s)$ are foremost contributing to the wrapping effect. The sets $\mathcal{R}(\tau_s)$ are filling the gaps between points in time t_s, which are not used for subsequent computations (see Fig. 1).

For the remainder of the paper we focus on Taylor expansions of second order with third order Lagrange remainder since higher order terms do not require a modification of the approach.

At the beginning of each time interval τ_s, the Taylor terms are re-evaluated according to a new expansion point z^*, which is indicated in line 3 of Alg. 1 by

$$\mathtt{taylor} \to z^*, w, A, B, D, E,$$

where the dependency on the time step is omitted in the notation. The expansion point z^* is chosen heuristically as $z^*(t_s) = [x^*(t_s), u^*]$, where $x^*(t_s) = x^c(t_s) + 0.5 \cdot r \cdot f(x^c(t_s), u^c) \approx x^c(t_s + 0.5 \cdot r)$, $u^* = u^c$ and $x^c(t_s)$, u^c are the volumetric centers of $\mathcal{R}(t_s)$, $\mathcal{U}$. Other linearization points within $\mathcal{R}(\tau_s)$ can be chosen, but better heuristics have not been found so far. Next, the linearization error for the reachable set within time intervals is obtained.

4.1 Overapproximating the Linearization Error

To obtain the set of linearization errors for the time interval solution $\mathcal{R}(\tau_s)$, (4) is abstracted to a linear differential inclusion

$$\begin{aligned} \dot{x} \in Ax(t) \oplus \Psi(\tau_s), \quad \Psi(\tau_s) = \mathcal{V}(\tau_s) \oplus \mathcal{L}(\tau_s), \\ \mathcal{V}(\tau_s) := \{v(z,u) | z \in \mathcal{R}(\tau_s) \times \mathcal{U},\, u \in \mathcal{U}\}. \end{aligned} \quad (11)$$

The computation of the set of linearization errors $\Psi(\tau_s)$ requires the computation of the reachable set $\mathcal{R}(\tau_s)$, which in turn requires $\Psi(\tau_s)$. This mutual dependence is initially resolved by an estimation of the set of linearization errors $\overline{\Psi}(\tau_s)$ with the goal that $\Psi(\tau_s) \subseteq \overline{\Psi}(\tau_s)$. This estimation is used to compute the set of state differences $\mathcal{R}^{\Delta}(\tau_s)$ for (11) using a slight modification of the standard techniques for linear system reachability presented in [1, Chap. 3.2]. The modification involves returning the set of state differences $\mathcal{R}^{\Delta}(\tau_s)$ instead of the set of states $\mathcal{R}(\tau_s)$. We denote this standard operation (see line 6 of Alg. 1) by

$$\mathcal{R}^{\Delta}(\tau_s) = \mathtt{post}^{\Delta}(\mathcal{R}(t_s), \overline{\Psi}(\tau_s), A).$$

Using the definition of $\mathcal{R}^{\Delta}(\tau_s)$ in (6), the overapproximation of the time interval solution is obtained as $\mathcal{R}(\tau_s) = \mathcal{R}(t_s) \oplus \mathcal{R}^{\Delta}(\tau_s)$, where $\mathcal{R}(t_s)$ is later represented by a non-convex set, $\mathcal{R}^{\Delta}(\tau_s)$ by a convex set, and the overapproximation $\mathcal{R}(\tau_s) \subseteq \mathtt{CH}\big(\mathcal{R}(t_s)\big) \oplus \mathcal{R}^{\Delta}(\tau_s)$ is used in line 7 of Alg. 1 for the efficient computation of $\mathcal{R}(\tau_s)$, since $\mathcal{R}(\tau_s)$ contributes only marginally to the wrapping effect by enlarging the overapproximation of the linearization error.

For a simple notation of subsequent computations, the operations

$$\mathrm{sq}(D, \mathcal{R}_1) := \left\{\lambda \middle| \lambda_i = \sum_{j=1}^{o}\sum_{k=1}^{o} D_{ijk}z_j z_k, z \in \mathcal{R}_1\right\},$$

$$\mathrm{mu}(D, \mathcal{R}_1, \mathcal{R}_2) := \left\{\lambda \middle| \lambda_i = \sum_{j=1}^{o}\sum_{k=1}^{o} D_{ijk}z_j \hat{z}_k, z \in \mathcal{R}_1, \hat{z} \in \mathcal{R}_2\right\}$$

are introduced. Using the definitions of the linearization error in (11) and the function $v(z,u)$ in (4), the set of linearization errors is overapproximated in line 9 of Alg. 1 by

$$\Psi(\tau_s) \subseteq w \oplus B \otimes \mathcal{U} \oplus \frac{1}{2}\mathrm{sq}(D, \mathcal{R}(\tau_s) \times \mathcal{U}) \oplus \mathcal{L}(\tau_s).$$

The computation of $B \otimes \mathcal{U}$ and $\mathrm{sq}(D, \mathcal{R}(\tau_s) \times \mathcal{U})$ is later presented for polynomial zonotopes. The Lagrangian remainder $\mathcal{L}(\tau_s)$ is small compared to the other sets and thus

less accurately overapproximated by obtaining the enclosing boxes of all sets and applying interval arithmetic [26]. This is denoted in line 8 of Alg. 1 by $\texttt{lagrangeRemainder}(\mathcal{R}(\tau_s), E(t_s), z^*(t_s))$.

In case $\Psi(\tau_s) \not\subseteq \overline{\Psi}(\tau_s)$, the result $\Psi(\tau_s)$ is uniformly enlarged in each direction by a factor $\lambda > 1$ around its volumetric center, denoted by $\overline{\Psi}(\tau_s) = \texttt{enlarge}\big(\Psi(\tau_s), \lambda\big)$ in line 5 of Alg. 1. Using the enlarged set $\overline{\Psi}(\tau_s)$, the the linearization error computation is started over.

4.2 Reachable Set at the Next Point in Time

The reachable set at the next point in time is obtained by a set-based computation of (10). Thereto, it is first required to compute $\mathcal{V}^{\Delta}(\tau_s)$ in line 12 of Alg. 1 using (5), for which the sets $\mathcal{R}_z^{\Delta}(\tau_s) = \mathcal{R}^{\Delta}(\tau_s) \times \mathcal{U}^{\Delta}$ and $\mathcal{R}_z(t_s) = \mathcal{R}(t_s) \times \mathcal{U}$ are introduced:

$$\begin{aligned}\mathcal{V}^{\Delta}(\tau_s) =& \texttt{varInputs}(\mathcal{R}_z(t_s), \mathcal{R}_z^{\Delta}(\tau_s), \mathcal{U}^{\Delta}, B, D)\\ :=& B \otimes \mathcal{U}^{\Delta} \oplus \frac{1}{2!}\Big(\mathrm{mu}(D, \mathcal{R}_z(t_s), \mathcal{R}_z^{\Delta}(\tau_s))\\ &\oplus \mathrm{mu}(D, \mathcal{R}_z^{\Delta}(\tau_s), \mathcal{R}_z(t_s)) \oplus \mathrm{sq}(D, \mathcal{R}_z^{\Delta}(\tau_s))\Big)\end{aligned}$$

The operator $\texttt{post}()$ computes $\mathcal{R}(t_{s+1})$ in line 13 of Alg. 1 by replacing exact values with sets in (10):

$$\begin{aligned}\mathcal{R}(t_{s+1}) =& \texttt{post}(\mathcal{R}(t_s), \mathcal{R}_z(t_s), w, A, B, D, \mathcal{V}^{\Delta}(\tau_s), \mathcal{L}(\tau_s))\\ :=& \underbrace{e^{Ar}\mathcal{R}(t_s)}_{=:\mathcal{PZ}_1} \oplus \mathcal{R}^{p,\Delta}(\mathcal{V}^{\Delta}(\tau_s) \oplus \mathcal{L}(\tau_s), r)\\ &\oplus \underbrace{\Gamma(r)\Big(w \oplus Bu^c \oplus \frac{1}{2!}\mathrm{sq}(D, \mathcal{R}_z(t_s))\Big)}_{=:\mathcal{PZ}_2}\end{aligned} \tag{12}$$

As previously mentioned, for small time steps r as typically used in reachability analysis, the set $\mathcal{R}^{p,\Delta}$ is small compared to the set $\mathcal{R}$ and $\mathcal{R}_z$. From this follows that the nonlinear terms capturing the original nonlinear dynamics are not marginalized by the abstractions applied to compute $\mathcal{R}^{p,\Delta}$.

The new algorithm includes nonlinear mappings so that in general convex sets are no longer mapped to convex sets as in other works, requiring a new non-convex set representation as presented in the following section.

5. POLYNOMIAL ZONOTOPES

Set representations in most previous works are convex since they are easy to represent and manipulate (see e.g. [4, 6, 7, 13, 21, 22, 29, 39]). However, the convexity property makes the efforts in capturing the nonlinear dynamics obsolete, since convex sets only work well for linear maps. A new non-convex set representation is proposed, which can be efficiently stored and manipulated. The new representation shares many similarities with Taylor models [25] (as shortly discussed later) and is a generalization of zonotopes, which have shown great performance for linear and nonlinear reachability analysis [4, 22].

Definition 1 (Polynomial Zonotope): Given a *starting point* $c \in \mathbb{R}^n$, multi-indexed *generators* $f^{([i],j,k,\ldots,m)} \in \mathbb{R}^n$, and single-indexed *generators* $g^{(i)} \in \mathbb{R}^n$, a polynomial zonotope

Algorithm 1 $\texttt{reach}(\mathcal{R}(0), t_f, \ldots)$

Require: Initial set $\mathcal{R}(0)$, input set $\mathcal{U}$, time horizon t_f, time step r, factor λ
Ensure: $\mathcal{R}([0, t_f])$

1: $t_0 = 0$, $s = 0$, $\Psi(\tau_0) = \{\mathbf{0}\}$, $\mathcal{R}^{union} = \emptyset$, $\mathcal{U}^{\Delta} = \mathcal{U} \oplus (-u^c)$
2: **while** $t_s < t_f$ **do**
3: $\quad\texttt{taylor} \rightarrow z^*, w, A, B, D, E$
4: $\quad$**repeat**
5: $\quad\quad\overline{\Psi}(\tau_s) = \texttt{enlarge}\big(\Psi(\tau_s), \lambda\big)$
6: $\quad\quad\mathcal{R}^{\Delta}(\tau_s) = \texttt{post}^{\Delta}(\mathcal{R}(t_s), \overline{\Psi}(\tau_s), A)$
7: $\quad\quad\mathcal{R}(\tau_s) = \texttt{CH}\big(\mathcal{R}(t_s)\big) \oplus \mathcal{R}^{\Delta}(\tau_s)$
8: $\quad\quad\mathcal{L}(\tau_s) = \texttt{lagrangeRemainder}(\mathcal{R}(\tau_s), E, z^*)$
9: $\quad\quad\Psi(\tau_s) = w \oplus B \otimes \mathcal{U} \oplus \frac{1}{2}\mathrm{sq}(D, \mathcal{R}(\tau_s) \times \mathcal{U}) \oplus \mathcal{L}(\tau_s)$
10: $\quad$**until** $\Psi(\tau_s) \subseteq \overline{\Psi}(\tau_s)$ } ①
11: $\quad\mathcal{R}_z(t_s) = \mathcal{R}(t_s) \times \mathcal{U}, \quad \mathcal{R}_z^{\Delta}(\tau_s) = \mathcal{R}^{\Delta}(\tau_s) \times \mathcal{U}^{\Delta}$
12: $\quad\mathcal{V}^{\Delta}(\tau_s) = \texttt{varInputs}(\mathcal{R}_z(t_s), \mathcal{R}_z^{\Delta}(\tau_s), \mathcal{U}^{\Delta}, B, D)$
13: $\quad\mathcal{R}(t_{s+1}) = \texttt{post}(\mathcal{R}(t_s), \mathcal{R}_z(t_s), w, A, B, D,$
14: $\quad\quad\quad\mathcal{V}^{\Delta}(\tau_s), \mathcal{L}(\tau_s))$
15: $\quad\mathcal{R}^{union} = \mathcal{R}^{union} \cup \mathcal{R}(\tau_s)$
16: $\quad t_{s+1} = t_s + r, \quad s := s + 1$ } ②
17: **end while**
18: $\mathcal{R}([0, t_f]) = \mathcal{R}^{union}$

is defined as

$$\begin{aligned}\mathcal{PZ} = \Big\{ & c + \sum_{j=1}^{p} \beta_j f^{([1],j)} + \sum_{j=1}^{p}\sum_{k=j}^{p} \beta_j \beta_k f^{([2],j,k)}\\ &+ \ldots + \sum_{j=1}^{p}\sum_{k=j}^{p}\cdots\sum_{m=l}^{p} \underbrace{\beta_j \beta_k \ldots \beta_m}_{\eta \text{ factors}} f^{([\eta],j,k,\ldots,m)}\\ &+ \sum_{i=1}^{q} \gamma_i g^{(i)} \Big| \beta_i, \gamma_i \in [-1, 1] \Big\}.\end{aligned} \tag{13}$$

The scalars β_i are called *dependent factors*, since changing their value does not only affect the multiplication with one generator, but other generators, too. On the other hand, the scalars γ_i only affect the multiplication with one generator, so that they are called *independent factors*. The number of dependent factors is p, the number of independent factors is q, and the polynomial order η is the maximum power of the scalar factors β_i. The order of a polynomial zonotope is defined as the number of generators ξ divided by the dimension, which is $\rho = \frac{\xi}{n}$. For a concise notation and later derivations, we introduce the matrices

$$\begin{aligned}E^{[i]} &= [\underbrace{f^{([i],1,1,\ldots,1)}}_{=:e^{([i],1)}} \cdots \underbrace{f^{([i],p,p,\ldots,p)}}_{=:e^{([i],p)}}] \text{ (equal indices)},\\ F^{[i]} &= [f^{([i],1,1,\ldots,1,2)}\ f^{([i],1,1,\ldots,1,3)} \cdots f^{([i],1,1,\ldots,1,p)}\\ &\quad\ f^{([i],1,1,\ldots,2,2)}\ f^{([i],1,1,\ldots,2,3)} \cdots f^{([i],1,1,\ldots,2,p)}\\ &\quad\ f^{([i],1,1,\ldots,3,3)} \ldots] \text{ (unequal indices)},\\ G &= [g^{(1)} \ \ldots \ g^{(q)}],\end{aligned}$$

and $E = \begin{bmatrix}E^{[1]} & \ldots & E^{[\eta]}\end{bmatrix}$, $F = \begin{bmatrix}F^{[2]} & \ldots & F^{[\eta]}\end{bmatrix}$ ($F^{[i]}$ is only defined for $i \geq 2$). Note that the indices in $F^{[i]}$ are ascending due to the nested summations in (13). In short form, a polynomial zonotope is written as $\mathcal{PZ} = (c, E, F, G)$. □

For a given polynomial order i, the total number of gener-

ators in $E^{[i]}$ and $F^{[i]}$ is derived using the number $\binom{p+i-1}{i}$ of combinations of the scalar factors β with replacement (i.e. the same factor can be used again). Adding the numbers for all polynomial orders and adding the number of independent generators q, results in $\xi = \sum_{i=1}^{\eta} \binom{p+i-1}{i} + q$ generators, which is in $\mathcal{O}(p^\eta)$ with respect to p. The non-convex shape of a polynomial zonotope with polynomial order 2 is shown in Fig. 2.

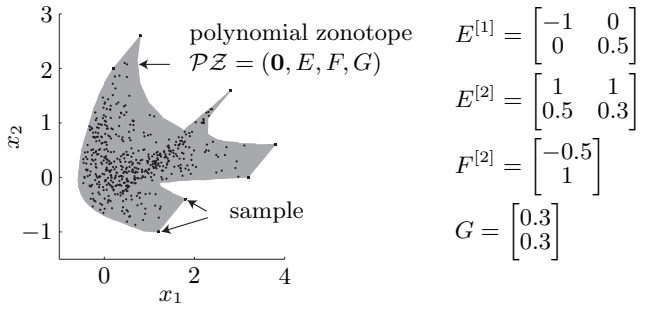

Figure 2: Overapproximative plot of a polynomial zonotope as specified in the figure. Random samples of possible values demonstrate the accuracy of the overapproximative plot.

A zonotope $\mathcal{Z}$ is a special case of a polynomial zonotope that has only generators $g^{(i)}$, which is denoted by $\mathcal{Z} = (c, G)$. Due to the absence of $E^{[i]}, F^{[i]}$ a zonotope is centrally symmetric to c so that for zonotopes, c is referred to as the *center* and not the *starting point*.

Although a Taylor model [25] is not a set, but a multidimensional polynomial plus a multidimensional interval, they can represent exactly the same sets than polynomial zonotopes when the input of the Taylor models is a multidimensional interval. The different organization of polynomial zonotopes, separating dependent from independent variables, makes it easier to overapproximate them by zonotopes or perform the order reduction techniques presented subsequently.

5.1 Operations on Polynomial Zonotopes

It is often required to overapproximate a polynomial zonotope by a zonotope:

Proposition 1 (Overapproximation by a Zonotope): A polynomial zonotope $\mathcal{PZ} = (c, E, F, G)$ can be overapproximated by a zonotope $\mathcal{Z} = \texttt{zonotope}(\mathcal{PZ}) = (\tilde{c}, \tilde{G})$ so that $\mathcal{PZ} \subseteq \mathcal{Z}$, by choosing

$$\tilde{c} = c + \frac{1}{2} \sum_{i=1}^{\lfloor \eta/2 \rfloor} \sum_{j=1}^{p} e^{([2i],j)},$$

$$\tilde{G} = \left[\tfrac{1}{2} E^{[2]} \quad \tfrac{1}{2} E^{[4]} \dots \quad E^{[1]} \quad E^{[3]} \dots \quad F^{[1]} \quad F^{[2]} \dots \quad G \right],$$

where $\lfloor \eta/2 \rfloor$ returns the lowest integer of $\eta/2$. The computational complexity for a given polynomial zonotope order ρ with respect to n is $\mathcal{O}(n^2)$. □

The proof is omitted due to limited space. A sketch of the proof is as follows: Generators with dependent factors are made independent by moving them into the generator matrix G, which always results in an overapproximation. Dependent factors β_i with even powers are within $[0, 1]$ (e.g. $\beta_1^2 \in [0, 1]$) instead of $[-1, 1]$ so that $E^{[2]}$, $E^{[4]}$, ... can be multiplied by 0.5 and their mean is added to c.

The multiplication of a matrix $M \in \mathbb{R}^{o \times n}$ with a polynomial zonotope $\mathcal{PZ} = (c, E, F, G)$ and the Minkowski addition of a zonotope $\mathcal{Z} = (\tilde{c}, \tilde{G})$ with $\mathcal{PZ}$ follow directly from the definition of polynomial zonotopes:

$$\begin{aligned} M \otimes \mathcal{PZ} &= (Mc, ME, MF, MG), \\ \mathcal{PZ} \oplus \mathcal{Z} &= (c + \tilde{c}, E, F, [G, \tilde{G}]). \end{aligned} \tag{14}$$

For a given polynomial zonotope order ρ, the computational complexity with respect to n is $\mathcal{O}(n^3)$ for the multiplication and $\mathcal{O}(n)$ for the addition. Note that the Minkowski addition of two polynomial zonotopes is never required since $\mathcal{R}^{\Delta}$ and $\mathcal{R}^{p,\Delta}$ are represented by zonotopes. The only addition between two polynomial zonotopes is between $\mathcal{PZ}_1 = (c_1, E_1, F_1, G_1)$ and $\mathcal{PZ}_2 = (c_2, E_2, F_2, G_2)$ in (12). Since both summands have the same dependent factors (proof omitted), one can apply an exact set addition, where the resulting polynomial zonotope is $(c_1 + c_2, E_1 + E_2, F_1 + F_2, [G_1, G_2])$. The generators with independent factors in G_1 and G_2 are added by concatenation as for the Minkowski addition in (14).

The reachability algorithm in Alg. 1 require set-based evaluations of higher order terms, such as the quadratic map $\text{sq}(D, \tilde{\mathcal{R}})$. When $\tilde{\mathcal{R}}$ is a zonotope, the result is exactly enclosed by a polynomial zonotope as shown by the following theorem.

Theorem 1 (Quadratic Map of a Zonotope): Given a zonotope $\mathcal{Z} = (d, g^{(1)}, \dots, g^{(h)})$ and a discrete set of matrices $Q^{(i)} \in \mathbb{R}^{n \times n}$, $i = 1 \dots n$, the set

$$\mathcal{PZ} = \text{sq}(Q, \mathcal{Z}) = \{\varphi | \varphi_i = x^T Q^{(i)} x,\ x \in \mathcal{Z}\}$$

is a polynomial zonotope (c, E, F, G), where the center is computed as $c_i = d^T Q^{(i)} d$, the generators of E and F are computed for $j = 1..h$, $k = j..h$ as

$$e_i^{([1],j)} = d^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} d, \quad e_i^{([2],j)} = {g^{(j)}}^T Q^{(i)} g^{(j)},$$
$$f_i^{([2],j,k)} = {g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)},$$

and $G = \emptyset$. The complexity of constructing this polynomial zonotope with respect to the dimension n is $\mathcal{O}(n^5)$. □

Proof: Inserting the definition of a zonotope into the set $\mathcal{PZ} = \{\varphi | \varphi_i = x^T Q^{(i)} x,\ x \in \mathcal{Z}\}$ yields

$$\left\{ \varphi \middle| \varphi_i = \left(d + \sum_{j=1}^{h} \beta_j g^{(j)} \right)^T Q^{(i)} \left(d + \sum_{j=1}^{h} \beta_j g^{(j)} \right),\ \beta_j \in [-1, 1] \right\},$$

which can be rearranged to

$$\begin{aligned} \Big\{ \varphi \Big| \varphi_i = & \underbrace{d^T Q^{(i)} d}_{c_i} + \sum_{j=1}^{h} \beta_j \underbrace{(d^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} d)}_{e_i^{([1],j)}} \\ & + \sum_{j=1}^{h-1} \sum_{k=j+1}^{h} \beta_j \beta_k \underbrace{({g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)})}_{f_i^{([2],j,k)}} \\ & + \sum_{j=1}^{h} \beta_j^2 \underbrace{{g^{(j)}}^T Q^{(i)} g^{(j)}}_{e_i^{([2],j)}}, \beta_i \in [-1, 1] \Big\}. \end{aligned}$$

Comparing the structure of the above terms with the definition of a polynomial zonotope in Def. 1 shows that the structure is identical and thus a polynomial zonotope.

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

Corollary 1 (Quadratic Map of a Polynomial Zonotope): Given is a polynomial zonotope $\mathcal{PZ} = (c, E, F, G)$ and the enclosing zonotope $\mathcal{Z} = \texttt{zonotope}(\mathcal{PZ}) = (c, h^{(1)}, \ldots, h^{(\sigma)})$ according to Prop. 1. The quadratic map is overapproximated by

$$\mathrm{sq}(Q, \mathcal{Z}) \subseteq \mathrm{sq}(Q, \mathcal{Z}_{EF}) \oplus \texttt{zonotope}(\mathrm{sq}(Q, \mathcal{Z}_G)),$$

where $\mathcal{Z}_{EF} = (c, h^{(1)}, .., h^{(p)})$ and $\mathcal{Z}_G = (\mathbf{0}, h^{(p+1)}, .., h^{(\sigma)})$, and $\mathbf{0}$ is a vector of zeros of proper dimension. The addition is performed according to (14) and the result has the same dependent factors as $\mathcal{PZ}$. The computational complexity for a given order ρ is identical to Theorem 1 ($\mathcal{O}(n^5)$). □

Proof: The zonotope $\mathcal{Z}$ is split into $\mathcal{Z} = \mathcal{Z}_{EF} \oplus \mathcal{Z}_G$, where the first one has p generators, which equals the number of generators in $E^{[1]}$. Ignoring dependencies always results in an overapproximation, such that

$$\mathrm{sq}(Q, \mathcal{Z}) \subseteq \mathrm{sq}(Q, \mathcal{Z}_{EF}) \oplus \underbrace{\mathrm{sq}(Q, \mathcal{Z}_G)}_{\subseteq \texttt{zonotope}(\mathrm{sq}(Q, \mathcal{Z}_G))}. \qquad \square$$

In order to make the above quadratic map accurate, the generators in $\mathcal{Z}_{EF}$ have to be dominant since $\mathrm{sq}(Q, \mathcal{Z}_{EF})$ is computed exactly, while $\mathrm{sq}(Q, \mathcal{Z}_G)$ is overapproximated by a zonotope. This is achieved by the order reduction technique described in the next subsection.

The operation $\mathrm{mu}(D, \tilde{\mathcal{R}})$ is similar to $\mathrm{sq}(D, \tilde{\mathcal{R}})$, which is the reason for omitting a detailed description.

5.2 Order Reduction of Polynomial Zonotopes

Many of the previously presented operations increase the order of polynomial zonotopes due to added generators. As a consequence, an order reduction technique has to be applied to limit the representation size and the computational costs. Most techniques for classical zonotopes remove generators and add new, but fewer ones that capture the spanned set of the removed generators (see e.g. [20]). This results in a reordering of the generators, which is no problem for zonotopes since the ordering of generators is irrelevant. However, generators of polynomial zonotopes can only be reordered within G, where generators are multiplied by independent factors γ_i. For this reason, a new order reduction technique is developed that does not change the ordering of generators in E and F.

The size of E and F is fixed and only G grows after performing the required operations presented in this work. Thus, it is required to remove generators from G and stretch the generators in E and F such that the ones removed from G are compensated in an overapproximative way. As for most applied order reduction techniques of zonotopes, heuristics are used rather than strict optimization techniques due to their favorable ratio of computational costs to obtained overapproximation.

Proposition 2 (Overapproximative Generator Removal): Given is $\mathcal{PZ} = (c, E, F, G)$ of which n linearly independent generators with indices $\mathrm{ind}_1, \ldots \mathrm{ind}_n$ are picked from $E^{[1]}$ and stored in $P = [e^{([1],\mathrm{ind}_1)} \ldots e^{([1],\mathrm{ind}_n)}]$ ($\det(P) \neq 0$). The overapproximating polynomial zonotope $\widehat{\mathcal{PZ}} = (c, \hat{E}, F, \hat{G})$ from which the generator $g^{(i)}$ is removed, is computed as

$$\hat{G} = \left[g^{(1)} \ldots g^{(i-1)}, g^{(i+1)}, \ldots g^{(q)} \right],$$

$$\hat{E} = \left[\hat{E}^{[1]} \; E^{[2]} \; \ldots \; E^{[\eta]} \right],$$

$$\hat{e}^{[1],j} = \begin{cases} (1 + (P^{-1} g^{(i)})_j) e^{([1],j)} \text{ for } j \in \{\mathrm{ind}_1, \ldots, \mathrm{ind}_n\}, \\ e^{([1],j)} \text{ otherwise.} \end{cases}$$

The computational complexity is $\mathcal{O}(n^3)$ due to the matrix inversion when using the Gauss-Jordan elimination. □

Proof: The generator $g^{(i)}$ can be composed from the generators in P:

$$g^{(i)} = e^{([1],\mathrm{ind}_1)} \phi_1 + \ldots + e^{([1],\mathrm{ind}_n)} \phi_n = P\phi \rightarrow \phi = P^{-1} g^{(i)}.$$

Note that P^{-1} can always be computed since $\det(P) \neq 0$. Thus, $g^{(i)}$ can be replaced by n new generators $e^{([1],\mathrm{ind}_1)} \phi_1$, …, $e^{([1],\mathrm{ind}_n)} \phi_n$, which causes an overapproximation since each generator has an independent factor γ_{q+j}:

$$\left\{ \gamma_i g^{(i)} \middle| \gamma_i \in [-1, 1] \right\} \subseteq \left\{ \sum_{j=1}^{n} \gamma_{q+j} e^{([1],\mathrm{ind}_j)} \phi_j \middle| \gamma_{q+j} \in [-1, 1] \right\}$$

The n new generators are aligned with the corresponding generators in $E^{[1]}$ and can be removed by stretching each $e^{([1],\mathrm{ind}_j)}$ by the factor

$$\frac{\|e^{([1],\mathrm{ind}_j)}\|_2 + \|e^{([1],\mathrm{ind}_j)} \phi_j\|_2}{\|e^{([1],\mathrm{ind}_j)}\|_2} = 1 + \phi_j = 1 + (P^{-1} g^{(i)})_j. \; \square$$

In this work, the longest generators $g^{(i)}$ (maximum 2-norm) are removed from G so that the generators in E are more dominant than the ones in G, which is required since only the generators in E are used for the exact computation of quadratic maps. The heuristic for choosing the set of picked generators P in Prop. 2 is as follows: The first generator is the one in $E^{[1]}$ that is best aligned with the removed generator $g^{(i)}$, and the other $n-1$ generators are the ones which have the least alignment with $g^{(i)}$ and among each other. The alignment is measured by the normalized scalar product $\frac{|g^{(i)^T} e^{([1],\mathrm{ind}_i)}|}{\|g^{(i)}\|_2 \|e^{([1],\mathrm{ind}_i)}\|_2}$, where a value of 1 occurs when the vectors are aligned and 0 for perpendicular vectors. The generators $g^{(i)}$ are removed until the order is less than a user-defined order. Without giving the proof, the complexity of the order reduction heuristic is $\mathcal{O}(n^3)$.

For a given order ρ of the polynomial zonotopes, no required operation has a complexity exceeding $\mathcal{O}(n^5)$ when a second order Taylor expansion with third order Lagrangian remainder is used. The scalability of the approach is demonstrated by the subsequent numerical examples.

6. NUMERICAL EXAMPLES

The approach is demonstrated for three examples. In all examples, the nonlinear dynamics is abstracted by a difference inclusion of second order with third order Lagrange remainder. The first example is a Van-der-Pol oscillator,

which is a standard example for nonlinear systems that have a limit cycle:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = (1 - x_1^2)x_2 - x_1$$

The reachable sets are computed with a time step of $r = 0.005$, $p = 20$ dependent factors (resulting in 230 generators with dependent factors), and 100 generators with independent factors. The initial set is a rectangle, where $x_1 \in [1.25, 1.55]$ and $x_2 \in [2.28, 2.32]$. When using polynomial zonotopes of polynomial order $\eta = 2$, a complete cycle can be computed without splitting, while with conventional zonotopes, the reachable set grows over all limits when splitting is not enforced, see Fig. 3. For a fair comparison, the maximum number of used generators are equal for the polynomial and the classical zonotope. We additionally plotted the results of the tool flow* [12] based on Taylor models, which has similar accuracy than the approach using polynomial zonotopes. The computational time in MATLAB is 23.2 seconds for polynomial zonotopes and similar for zonotopes since the total number of generators is equal. The computational time in flow* is 46.1 seconds, but an improved and faster version will soon be released. The computations have been performed on an Intel XEON X5690 processor with 3.47 Ghz. An advantage of this approach compared to Taylor models is that unlike Taylor models, arbitrarily time-varying uncertain inputs can be considered.

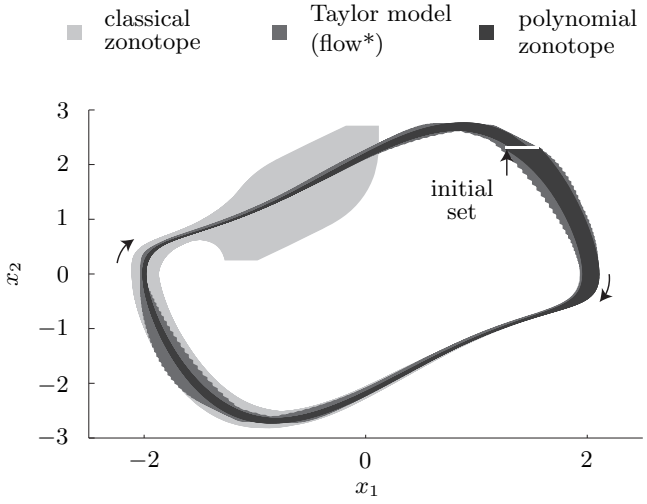

Figure 3: Reachable sets of the Van-der-Pol oscillator.

The second example is a biological model taken out of [17] with strong nonlinearity measure for studying the mitochondrial theory of aging. Due to the strong nonlinear nature, the auxiliary variables

$$S = \frac{x_9}{x_9 + ATP_c} \frac{k_1}{1 + (\frac{x_9}{ATP_c})^3} \frac{1}{x_1 + \frac{2x_2}{GDF+1} + \frac{x_3+x_4+x_5+x_6}{GDF}}$$
$$B = \frac{PAO_x}{x_8(x_1 + x_2 + x_3) + RDF\, x_8(x_4 + x_5 + x_6)}$$

are introduced. The differential equations of the model are

$$\begin{aligned}
\dot{x}_1 =& Sx_1 + \frac{2Sx_2}{GDF+1} - (\alpha + (k_M + k_D)x_8)x_1,\\
\dot{x}_2 =& \frac{-2Sx_2}{GDF+1} + \frac{2Sx_3}{GDF} + k_M x_8 x_1 - (\beta + (k_M + k_D)x_8)x_1,\\
\dot{x}_3 =& \frac{-2Sx_3}{GDF} + k_M x_8 x_2 - (\gamma + k_D x_8)x_3,\\
\dot{x}_4 =& \frac{S(x_4 + x_5)}{GDF} + k_D x_8 x_1 - (\alpha + k_M RDF x_8)x_4,\\
\dot{x}_5 =& \frac{-Sx_5}{GDF} + \frac{2Sx_6}{GDF} + k_D x_8 x_2\\
&+ k_M RDF x_8 x_1 - (\beta + k_M RDF x_8)x_5,\\
\dot{x}_6 =& \frac{-2Sx_6}{GDF} + k_D x_8 x_3 + k_M RDF x_8 x_5 - \gamma x_6,\\
\dot{x}_7 =& \frac{x_9}{x_9 + ATP_c} \frac{k_2}{1 + B} - \delta x_7,\\
\dot{x}_8 =& k_R - \frac{k_3(x_7 x_8)}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6},\\
\dot{x}_9 =& k_{ATP} x_1 + 0.5 k_{ATP} x_2 - \frac{x_9}{x_9 + ATP_c}\\
&\left(\frac{k_{EM} k_1}{1 + (\frac{x_9}{ATP_c})^3} + k_{EC} + \frac{k_{EP} k_2}{1 + B}\right),
\end{aligned}$$

where x_1-x_9 are the continuous state variables and the remaining variables are parameters whose values are as listed in [17]. The initial set has the same center as the one in [17], but the size of the initial set is 20 times larger for each coordinate, so that the volume is 20^9 times larger than in [17]. The initial set is bounded by a hyperrectangle, where $x_1 - x_3 \in [481, 521]$, $x_4 - x_6 \in [81, 121]$, $x_7 \in [181, 221]$, $x_8 \in [481, 521]$, and $x_9 \in [0, 40]$. The time increment is chosen as $r = 1.5 \cdot 10^{-5}$ and the time horizon $t_f = 0.01$ is as in [17]. For the reachable set computations, polynomial zonotopes with $p = 9$ dependent factors (resulting in 54 generators with dependent factors) and 90 generators with independent factors are used.

The results are compared for 3 approaches: Polynomialization with polynomial zonotopes, polynomialization with classical zonotopes, and linearization with classical zonotopes, which is the approach in [4]. For this example, it is not possible to compare the results with flow* since the tool does not yet support non-polynomial differential equations. When using polynomialization with polynomial zonotopes, the reachable set for the entire time horizon is computed without splitting. For polynomialization in combination with conventional zonotopes, the reachable set has to be split for the first time at $t = 0.0050$ and 6 parallel computations are required in the end. The linearization with classical zonotopes already requires the first split at time $t = 9 \cdot 10^{-5}$ and 98 parallel reachable set computations are performed in the end. Due to the possibility of splitting reachable sets, all approaches provide similar accuracy. However, the approach using polynomial zonotopes results in the tightest overapproximation.

Selected projections of the reachable set using polynomial zonotopes are shown in Fig. 4. For a fair comparison, the total number of used generators for the polynomial and the classical zonotope are chosen equal. The computation time is 1180 s using polynomial zonotopes, while almost all the time (1121 s) is spent on evaluating the third order Lagrange remainder using the interval arithmetic toolbox INTLAB

[36] for MATLAB. Since this toolbox is only efficient when matrix operations are used, the performance can be drastically improved when the interval computations are performed by precompiled code using e.g. C++. When using polynomialization in combination with classical zonotopes, the computation takes 4121 s, which is more than 3 times longer compared to polynomial zonotopes. For the linearization approach, the total computation time is even 18316 s, which is more than 15 times longer compared to polynomialization in combination with polynomial zonotopes. Note that for the specific example, the variable x_9 dominates the linearization error and thus splitting is mainly performed in one direction only. Otherwise, splits in many more directions are required, resulting in an exponential complexity with respect to the variables that have to be split.

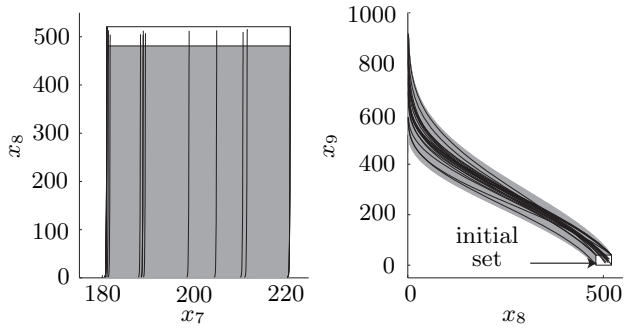

Figure 4: Reachable sets (gray area) of the biological aging model starting from a set of initial states (white area). Black lines show possible trajectories.

Finally, it is demonstrated by a third example that the proposed approach can be directly applied to hybrid systems. Thereto, the aging model is made hybrid by changing the parameter k_{ATP} from 1200 to 120 when the value $x_9 = 100$ is reached. As a consequence, the state x_9 converges to a different steady state, while the other states are only marginally affected, see Fig. 5. The extension to hybrid systems is performed as in [1], which computes the intersection with the guard set $x_9 = 100$ geometrically. Integrating the nonlinear reachability into a more efficient approach [2], which avoids computationally expensive guard intersections, is part of future work. The computation time is 1296 s, where again almost all the time (1155 s) is spent on interval arithmetic. The computation has been performed on the same machine as used for the previous experiments.

7. CONCLUSIONS

This work presents a new approach for reachability analysis of hybrid systems with nonlinear continuous dynamics. The new method successfully improves the major problem of linearization-based approaches: When the linearization error becomes too large, the linearization region has to be split, resulting in many regions that have to be considered simultaneously. Splitting results in an exponential complexity in the number of continuous state variables contributing to the linearization error.

By improving the accuracy of the approach, due to abstracting to polynomial difference inclusions instead of linear difference inclusions, splitting is avoided in the presented examples, saving computation time and improving accuracy.

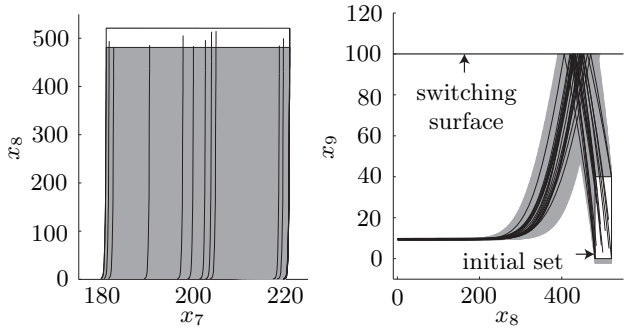

Figure 5: Reachable sets (gray area) of the hybrid biological aging model starting from a set of initial states (white area). Black lines show possible trajectories.

The new approach requires the use of a non-convex set representations, which is newly developed in this work. Since many aspects are implemented for the first time, there is a lot of potential for future improvements. For instance, one could compute with Taylor expansions of order greater than two and also use polynomial zonotopes with polynomial order $\eta > 2$, making it possible to compute with even larger sets of initial states in the future.

Acknowledgment

The author likes to thank Xin Chen for support with the tool flow* and discussions on Taylor models.

8. REFERENCES

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

[2] M. Althoff and B. H. Krogh. Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *Hybrid Systems: Computation and Control*, pages 45–54, 2012.

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

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

[5] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[6] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler. Recent progress in continuous and hybrid reachability analysis. In *Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 1582–1587, 2006.

[7] E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43:451–476, 2007.

[8] G. Behrmann, A. David, K. G. Larsen, P. Pettersson, and W. Yi. Developing UPPAAL over 15 years. *Softw., Pract. Exper.*, 41(2):133–142, 2011.

[9] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4:83–97, 1998.

[10] P. Bulychev, A. David, K. G. Larsen, A. Legay, G. Li, D. B. Poulsen, and A. Stainer. Monitor-based statistical model checking for weighted metric temporal logic. In *Proc. of 18th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 168–182, 2012.

[11] X. Chen, E. Ábráham, and G. Frehse. Efficient bounded reachability computation for rectangular automata. In *5th Workshop on Reachability Problems*, LNCS 6945, pages 139–152. Springer, 2011.

[12] 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*, 2012.

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

[14] E. Clarke, A. Fehnker, Z. Han, B. H. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald. Abstraction and counterexample-guided refinement in model checking of hybrid systems. *International Journal of Foundations of Computer Science*, 14(4):583–604, 2003.

[15] E. M. Clarke and P. Zuliani. Statistical model checking for cyber-physical systems. In *Proc. of the 9th International Symposium on Automated Technology for Verification and Analysis*, LNCS 6996, pages 1–12, 2011.

[16] T. Dang. *Vérification et synthèse des systèmes hybrides*. PhD thesis, Institut National Polytechnique de Grenoble, 2000.

[17] T. Dang, C. Le Guernic, and O. Maler. Computing reachable states for nonlinear biological models. In *Computational Methods in Systems Biology*, LNCS 5688, pages 126–141. Springer, 2009.

[18] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *Hybrid Systems: Computation and Control*, pages 11–19, 2010.

[19] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. *International Journal on Software Tools for Technology Transfer*, 10:263–279, 2008.

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

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

[22] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 257–271. Springer, 2006.

[23] A. Girard and G. J. Pappas. Verification using simulation. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 272–286. Springer, 2006.

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

[25] J. Hoefkens, M. Berz, and K. Makino. *Scientific Computing, Validated Numerics, Interval Methods*, chapter Verified High-Order Integration of DAEs and Higher-Order ODEs, pages 281–292. Springer, 2001.

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

[27] A. A. Julius, G. E. Fainekos, M. Anand, I. Lee, and G. J. Pappas. Robust test generation and coverage for hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 4416, pages 329–342. Springer, 2007.

[28] M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 348–362. Springer, 2006.

[29] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 202–214. Springer, 2000.

[30] K. Makino and M. Berz. Rigorous integration of flows and ODEs using Taylor models. In *Proc. of Symbolic-Numeric Computation*, pages 79–84, 2009.

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

[32] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45(1):236–262, 2007.

[33] A. Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, 2010. ISBN 978-3-642-14508-7.

[34] S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

[35] S. Ratschan and Z. She. Constraints for continuous reachability in the verification of hybrid systems. In *Proc. of Artificial Intelligence and Symbolic Computation*, LNCS 4120, pages 196–210. Springer, 2006.

[36] S. M. Rump. *Developments in Reliable Computing*, chapter INTLAB - INTerval LABoratory, pages 77–104. Kluwer Academic Publishers, 1999.

[37] S. Sankaranarayanan. Automatic abstraction of non-linear systems using change of bases transformations. In *Hybrid Systems: Computation and Control*, pages 143–152, 2011.

[38] S. Sankaranarayanan and G. Fainekos. Falsification of temporal properties of hybrid systems using the cross-entropy method. In *Hybrid Systems: Computation and Control*, pages 125–134, 2012.

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