Sparse Polynomial Zonotopes: A Novel Set Representation for Reachability Analysis

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

Matthias Althoff
Technische Universität München
althoff@tum.de

ABSTRACT
We introduce sparse polynomial zonotopes, a new set representation for formal verification of hybrid systems. Sparse polynomial zonotopes can represent non-convex sets and are generalizations of zonotopes and Taylor models. Operations like Minkowski sum, quadratic mapping, and reduction of the representation size can be computed with polynomial complexity w.r.t. the dimension of the system. In particular, for the reachability analysis of nonlinear systems, the wrapping effect is substantially reduced using sparse polynomial zonotopes as demonstrated by numerical examples. In addition, we can significantly reduce the computation time compared to zonotopes.

Keywords
Reachability analysis, nonlinear dynamics, hybrid systems, sparse polynomial zonotopes.

1. INTRODUCTION
Efficient set representations are highly relevant for controller synthesis and formal verification of hybrid systems, since many underlying algorithms compute with sets; see e.g., [12,24,35,38]. Improvements originating from a new set representation often significantly reduce computation time and improve the accuracy of set-based computations.

State of the Art
Fig. 1 shows relevant set representations and their relations to each other. Almost all typical set representations are convex, except Taylor models, star sets, and polynomial zonotopes. Since all convex sets can be represented by support functions, which are closed under Minkowski addition, linear maps, and convex hull operations, they are a good choice for reachability analysis [17,18,20,34]. Ellipsoids and polytopes are special cases of support functions, which are often used for reachability analysis [17,18,20,34] and computations of invariant sets [1,11,28,33]. However, the disadvantage of ellipsoids is that they are not closed under intersection and Minkowski addition; the disadvantage of polytopes is that Minkowski sum is computationally expensive [37].

One important subclass of polytopes is zonotopes, which can be represented compactly by so-called generators: a zonotope with \( m \) generators in \( n \) dimensions might have up to \( \binom{m}{n} \) halfspaces. More importantly, Minkowski sum and linear maps can be computed cheaply, making them a good choice for reachability analysis [5,8,19]. Two relevant extensions to zonotopes are zonotope bundles [6], where the set is represented implicitly by the intersection of several zonotopes, and constrained zonotopes [36], where additional equality constraints on the zonotope factors are considered. Zonotope bundles, as well as constrained zonotopes, are both able to represent any polytope. Both representations make use of lazy computations and thus suffer much less from the curse of dimensionality as it is the case for polytopes [37].

A special case of zonotopes are multi-dimensional intervals, which are particularly useful for range bounding of nonlinear functions via interval arithmetic [23], but they are also used for reachability analysis [10,32]. Since intervals are not closed under linear maps, one often has to split them to reduce the wrapping effect [29].

In general, reachable sets of nonlinear systems are non-convex, so that tight enclosures can only be achieved using non-convex set representations when avoiding splitting of reachable sets. Taylor models [30], which consist of a polynomial and an interval remainder part, are an example of non-convex set representation. They are typically used for range bounding [31] and reachability analysis [13]. Another type of non-convex set representation is polynomial zonotopes, which are introduced in [3] and are a generalization of Taylor models as later shown in this work. Yet another way to represent non-convex sets is star sets, which are especially useful for simulation-based reachability analysis [10,15]. While star sets are very expressive, it is yet unclear how some operations, such as nonlinear mapping,
are computed.

Overview
In this work, we introduce a new non-convex set representation called sparse polynomial zonotopes, which is a non-trivial extension of polynomial zonotopes from [3] and exhibits the following major advantages: a) sparse polynomial zonotopes enable a very compact representation of sets, b) they are closed under all relevant operations, c) many other set representations can be converted to an sparse polynomial zonotope, and most important, d) all operations have at most polynomial complexity.

The remainder of this paper is structured as follows: In Sec. 2, the formal definition of sparse polynomial zonotopes is provided and important operations on them are derived. We show how sparse polynomial zonotopes provide substantially better results for reachability analysis in Sec. 3, which is demonstrated in Sec. 4 on two numerical examples.

Notation
In the remainder of this paper, we will use the following notations: Sets are always denoted by calligraphic letters, matrices by uppercase letters, and vectors by lowercase letters. Given a discrete set \( \mathcal{H} \in \{ \cdot \}^n \), \( |\mathcal{H}| = n \) denotes the cardinality of the set and \( \mathcal{H}(i) \) refers to the \( i \)-th entry of the set \( \mathcal{H} \). Given a vector \( b \in \mathbb{R}^n \), \( b(i) \) refers to the \( i \)-th entry. Given a matrix \( A \in \mathbb{R}^{n \times m} \), \( A(i,:) \) represents the \( i \)-th row, \( A(:,j) \) the \( j \)-th column, and \( A(i,j) \) the \( j \)-th entry of matrix row \( i \). Given a discrete set of positive integer indices \( \mathcal{H} \) with \( |\mathcal{H}| < m \), \( A(\mathcal{H},:) \) is used for \( [A_{(\mathcal{H}(1),:) \ldots A_{(\mathcal{H}(|\mathcal{H}|),:)}}] \), where \( [C \ D] \) denotes the concatenation of two matrices \( C \) and \( D \). The symbols \( 0 \) and \( 1 \) represent matrices of zeroes and ones of proper dimension. The left multiplication of a matrix \( M \in \mathbb{R}^{m \times n} \) with a set \( S \subset \mathbb{R}^n \) is defined as \( M \cdot S = \{ M \cdot s \mid s \in S \} \), and the Minkowski addition of two sets \( S_1 \subset \mathbb{R}^n \) and \( S_2 \subset \mathbb{R}^n \) is defined as \( S_1 \oplus S_2 = \{ s_1 + s_2 \mid s_1 \in S_1, s_2 \in S_2 \} \). We further introduce an \( n \)-dimensional interval as \( I := [l, u], \) \( \forall i \leq u(i), l, u \in \mathbb{R}^n \).

For the derivation of computational complexity, we consider all binary operations except concatenations, and initializations are explicitly not considered.

2. SPARSE POLYNOMIAL ZONOTOPES

Let us first define sparse polynomial zonotopes (SPZs), followed by derivations of relevant operations on them.

Definition 1. Given a generator matrix of dependent generators \( G \in \mathbb{R}^{n \times h} \), a generator matrix of independent generators \( G_I \in \mathbb{R}^{n \times q} \), and an exponent matrix \( E \in \mathbb{Z}_{\geq 0} \), any SPZ is defined as

\[
\mathcal{PZ} = \left\{ \sum_{i=1}^{h} \left( \prod_{k=1}^{p} \alpha_k^{E_{(k,i)}} \right) G_{i,:} + \sum_{j=1}^{q} \beta_j G_{I,:} \right| \alpha_i, \beta_j \in [-1,1] \}
\]

The scalars \( \alpha_i \) are called dependent factors since a change in their value affects multiplication with multiple generators. Consequently, the scalars \( \beta_j \) are called independent factors because they only affect multiplication with one generator. The number of dependent factors is \( p \), the number of independent factors is \( q \), and the number of dependent generators is \( h \). The order of an SPZ \( \rho \) is defined as \( \rho = \frac{h \cdot q}{n} \).

For the derivation of the computational complexity of set operations, we introduce

\[
h = c_h n, \quad p = c_p n, \quad q = c_q n,
\]

with \( c_h, c_p, c_q \in \mathbb{R}_{\geq 0} \). In the remainder of this paper, we call the single term \( \alpha_1^{E_{(1,i)}} \cdots \alpha_p^{E_{(p,i)}} \cdot G_{(:,i)} \) a monomial, and \( \alpha_1, \ldots, \alpha_p \) the variable part of the monomial. In order to keep track of the dependencies between the dependent factors from different SPZs, we introduce an unambiguous integer identifier for each dependent factor \( \alpha_i \) and store the identifiers for all dependent factors in a list \( \mathcal{ID} \in \{\mathbb{N}\}^p \).

Using this identifier list, we introduce the shorthand \( \mathcal{PZ} = \{ G, G_I, E, \mathcal{ID} \} \subset \mathbb{R}^n \) for the representation of SPZs. All components of a set \( \mathbb{Z} \) have the index \( i \), e.g., \( p_i, h_i \), and \( q_i \) belong to \( \mathcal{PZ} \). To make SPZs more intuitive, we introduce the following example:

Example 1. The SPZ

\[
\mathcal{PZ} = \left\{ \begin{bmatrix} 2 & 1 & 2 \\ 0 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \end{bmatrix}, \{1,2\} \right\},
\]

defines the set

\[
\mathcal{PZ} = \left\{ \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \alpha_1 + \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \alpha_2 + \begin{bmatrix} 2 & 1 \end{bmatrix} \alpha_3 \beta_1 \left| \alpha_1, \alpha_2, \beta_1 \in [-1,1] \right. \right\}.
\]

The construction of this SPZ is visualized in Fig. 2 (a) shows the set spanned by the first two dependent generators, (b) shows the addition of the dependent generator with the mixed term \( \alpha_1^2 \alpha_2, (c) \) shows the addition of the independent generator, and (d) visualizes the final set.

Our sparse representation is a non-trivial extension of polynomial zonotopes [3, Def. 1], resulting in completely different algorithms for operations on them. In [3], the generators for all possible combinations of dependent factors up to a certain polynomial degree are stored. For the
one-dimensional polynomial zonotope \( \mathcal{P}Z = \{ \alpha_1 \cdots \alpha_19 \cdot \alpha_20^p | \alpha_1, \alpha_20 \in [-1, 1] \} \) with \( p = 20 \) dependent factors and with a polynomial degree of 10, the number of dependent generators is \( h = 30045015 \). This demonstrates that the number of stored generators can become very large if the polynomial degree and the number of dependent factors are high, which makes computations on the previous set representation very inefficient. We in turn use a sparse representation, where only the generators for desired factor combinations are stored, which enables the representation of the above polynomial zonotope with only one single generator. Furthermore, our representation does not require limiting the polynomial degree of the polynomial zonotope in advance, which is advantageous for reachability analysis, as shown in Sec. 3.1.

### 2.1 Preliminary Operations

First, we introduce preliminary set operations that are required for many other operations.

#### Merging the Set of Identifiers

For all set operations that involve two or more SPZs, the operator mergeID() is necessary in order to build a common representation of exponent matrices to fully exploit the dependencies between identical dependent factors:

**Proposition 1. (Merge ID)** Given two SPZs, \( \mathcal{P}Z_1 = \{G_1, G_{11}, E_1, ID_1\} \) and \( \mathcal{P}Z_2 = \{G_2, G_{12}, E_2, ID_2\} \), mergeID() returns two adjusted SPZs with identical identifier lists that are equivalent to \( \mathcal{P}Z_1 \) and \( \mathcal{P}Z_2 \), and has a complexity of \( O(n^2) \):

\[
\text{mergeID}(\mathcal{P}Z_1, \mathcal{P}Z_2) = \{\{G_1, G_{11}, E_1, \overline{ID}\}, \{G_2, G_{12}, E_2, \overline{ID}\}\}
\]

with \( \overline{ID} = \{ID_1, ID_2(\mathcal{H}(0)), \ldots, ID_2(\mathcal{H}(n))\} \)

\[
\begin{align*}
\overline{E}_1 &= \begin{bmatrix} E_1 \end{bmatrix} \in \mathbb{R}^{m \times h_1} \\
\overline{E}_2(i) &= \begin{cases} E_2(i), & \text{if } \exists j \overline{ID}_j(i) = ID_2(i) \\
0, & \text{else} \end{cases} \\
\mathcal{H} &= \{i \mid ID_2(i) \not\subseteq ID_1\}
\end{align*}
\]

where \( a = |\mathcal{H}| + p_1 \).

**Proof.** (Merge ID) The extension of the exponent matrices with all-zero rows only changes the representation of the set, but not the set itself.

**Complexity:** The only operation with super-linear complexity is the construction of the set \( \mathcal{H} \) with \( O(p_1 \cdot p_2) = O(n^2) \) using \( \overline{E} \).

#### Transformation to a Compressed Representation

Some set operations result in an SPZ that contains multiple monomials with an identical variable part, which we combine to one single monomial:

**Proposition 2. (Compact)** Given an SPZ \( \mathcal{P}Z = \{G, G_1, E, ID\} \), the operation compact() returns a compressed representation of the set \( \mathcal{P}Z \) and has a complexity of \( O(n^2 \log(n)) \):

\[
\text{compact}(\mathcal{P}Z) = \{G, G_1, E, ID\}
\]

with \( \overline{E} = \text{uniqueColumns}(E) \in \mathbb{R}^{p \times k} \)

\[
\mathcal{M}_i = \{i \mid \overline{E}(i, j) = E(i, j) \forall j \in \{1 \ldots p\}\} \quad (5)
\]

\[
\overline{G} = \left[ \sum_{i \in \mathcal{M}_1} G_{(i, \cdot)} \ldots \sum_{i \in \mathcal{M}_k} G_{(i, \cdot)} \right],
\]

where the operation uniqueColumns() removes identical matrix columns until all columns are unique.

**Proof.** (Compact) Since the number of unique columns \( k \) of matrix \( E \) is smaller than the number of overall columns \( h \), the matrices \( \overline{E} \) and \( \overline{G} \) are smaller or of equal size compared to the matrices \( E \) and \( G \), which results in a compressed representation of the set.

**Complexity:** The operation uniqueColumns() in combination with the construction of the sets \( \mathcal{M}_i \) can be efficiently implemented by sorting the matrix columns followed by an identification of identical neighbors, which can be realized with a worst case complexity of \( O(ph \log(h)) \). Since all other operations have at most quadratic complexity, the overall complexity is \( O(n^2 \log(n)) \) using \( \overline{E} \).

The operation compact() is applied after all set operations that potentially result in SPZs containing multiple monomials with identical variable part. These operations are conversion from a Taylor model, Minkowski addition and quadratic map.

#### 2.2 Conversion from other Set Representations

This section shows how other set representations can be converted to SPZs.

**Zonotope and Interval**

We first provide the definition of a zonotope:

**Definition 2. (Zonotope)** [13, Def. 1] Given a center \( c \in \mathbb{R}^n \) and a generator matrix \( G \in \mathbb{R}^{n \times m} \), a zonotope is defined as

\[
\mathcal{Z} = \left\{ c + \sum_{i=1}^{m} \alpha_i G_{(i, \cdot)} \mid \alpha_i \in [-1, 1] \right\}.
\]

For a compact notation, we introduce the shorthand \( \mathcal{Z} = \{c, G\} \). Any zonotope can be converted to an SPZ:

**Proposition 3. (Conversion Zonotope)** A zonotope \( \mathcal{Z} = \{c, G\} \) can be represented by an SPZ \( \mathcal{P}Z \):

\[
\mathcal{P}Z = \{[c G], \mathcal{O}, \mathcal{I} \mathbf{0} I\}, \{1, \ldots, m\}\}
\]

where \( I \in \mathbb{R}^{m \times m} \) is the identity matrix. The complexity of the conversion is \( O(1) \).

**Proof.** (Conversion Zonotope) If we insert \( E = \mathcal{O} \) and \( G_1 = \mathcal{I} \) into \( \overline{E} \), we obtain a zonotope (see \( \overline{G} \)).

**Complexity:** The complexity is constant since the conversion only involves concatenations and initializations.

Since any interval can be represented as a zonotope [2] Prop. 2.1, their conversion to an SPZ is straightforward.
Taylor Model

First, we formally define multi-dimensional Taylor models:

**Definition 3. (Taylor Model)** \[\text{Def. 2.1}\] Given a vector field \( p : \mathbb{R}^n \rightarrow \mathbb{R}^n \), where each subfunction \( p(i) : \mathbb{R}^n \rightarrow \mathbb{R} \) is a polynomial function defined as

\[
p(i)(x(1), \ldots, x(n)) = \sum_{j=1}^{m_i} \sum_{k=1}^{n} b_{ij} E^{(k,j)}(x),
\]

and an interval \( I \subset \mathbb{R}^n \), a Taylor model \( T(x) \subset \mathbb{R}^n \) is defined as

\[
T(x) = \left\{ \begin{bmatrix} p(1)(x) \\
\vdots \\
p(n)(x) \end{bmatrix} \mid y \in I \right\},
\]

where \( E_i \) is a vector of polynomial coefficients.

For a concise notation, we introduce the shorthand \( \mathcal{T}(x) = \{ p, I \} \). The set defined by any Taylor model can be converted to an SPZ:

**Theorem 1. (Conversion Taylor Model)** The set defined by a Taylor model \( T(x) = \{ p, I \} \) on the domain \( x \in D \) with \( D = [a_i, b_i] \) and \( I = [r_i, s_i] \) can be equivalently represented by an SPZ \( \mathcal{PZ} = \{ \mathcal{T}(x) \} \):

\[
\mathcal{PZ} = \left\{ \begin{bmatrix} \frac{t_i + u_i}{2} \hat{G} \end{bmatrix}, G_1, [0 \hat{E}], (1, \ldots, n) \right\}
\]

with \( \hat{G} = \begin{bmatrix} \hat{G}^0 \\
\vdots \\
\hat{G}^n \end{bmatrix} \), \( \hat{E} = [E_1 \ldots E_n] \), and \( G_1 = \begin{bmatrix} u_i(1) - l_i(1) \\
\vdots \\
u_i(n) - l_i(n) \end{bmatrix} \).

The vectors \( b_i \) and the matrices \( E_i \) result from the definition

\[
p(i)(\epsilon_1, \ldots, \epsilon_n) = \sum_{j=1}^{m_i} \sum_{k=1}^{n} b_{ij} E^{(k,j)}(\epsilon),
\]

where \( \epsilon_k = 0.5(t_d(k) + u_d(k)) + 0.5 \alpha_k (u_d(k) - l_d(k)) \), \( \alpha_k \in [-1, 1], k = 1 \ldots n, \) and \( p(i)(\cdot) \) is defined as in [8]. The complexity of the conversion is \( \mathcal{O}(n^2 2^{n^2}) \).

**Proof.** (Conversion Taylor Model) The auxiliary variables \( \epsilon_k \) represent the domain \( D \) with dependent factors \( \alpha_k \in [-1, 1] \). Evaluation of the polynomial functions \( p(i)(\cdot) \) from [8] with the substitution \( x(k) = \epsilon_k \) in [8] therefore directly yields the definition of the dependent part of an SPZ (see [8]). The equation for the independent generator matrix \( G_1 \) follows from the representation of the interval \( I = [r_i, s_i] \) as a zonotope \( Z = \{0.5(t_i + s_i), 0.5 \text{diag}(u_i(1) - l_i(1), \ldots, u_i(n) - l_i(n)) \} \), where the operator \( \text{diag}(\cdot) \) constructs a diagonal matrix.

Complexity: Let \( m = \max(m_1, \ldots, m_n) \) and \( e = \max(\max(E_1), \ldots, \max(E_n)) \), where \( \max(A) \) returns the maximum entry of matrix \( A \). The exponent matrix \( \hat{E} \) has \( h = nm(2^n) \) columns in the worst case, resulting from the evaluation of the functions \( p(i)(\epsilon_1, \ldots, \epsilon_n) \). Since the complexity of \( \mathcal{O}(\hat{E}) \) is \( \mathcal{O}(nh) \) and \( \hat{E} = n \), the subsequent application of the \( \mathcal{O}(\hat{E}) \) operator has complexity \( \mathcal{O}(n^3 m(n^2 n^{2^n}) \log(nm(2^n)^n)) = \mathcal{O}(n^n m(2^n)^n) \) using [2]. This is also the overall complexity of the conversion, since all other operations have a lower complexity.

**Corollary 1.** The set defined by any Taylor model can be equivalently represented by an SPZ, but not every SPZ can be represented by a Taylor model.

The corollary follows directly from the fact that the number of dependent factors \( p \) of an SPZ can be larger than the dimension of the state space \( n \), which includes the special case of Taylor models for which \( p = n \).

### 2.3 Enclosure by other Set Representations

This subsection describes how SPZs can be enclosed by other set representations.

**Zonotope**

We first show how an SPZ can be enclosed by a zonotope:

**Proposition 4. (Zonotope)** Given an SPZ \( \mathcal{PZ} = \{ G, G_1, E, I \} \), the operation \( \text{zono}(\cdot) \) returns a zonotope that encloses \( \mathcal{PZ} \) and has a complexity of \( \mathcal{O}(n^2) \):

\[
\text{zono}(\mathcal{PZ}) = \left\{ \begin{bmatrix} \sum_{i=1}^{N} G_i(N_{i,j}) + \frac{1}{2} \sum_{i=1}^{N} G_i(N_{i,j}) \cdot \hat{E} \end{bmatrix} \right\}
\]

with \( N = \{ i \mid E_{i,j} = 0 \forall j \in \{1, \ldots, p \} \} \)

\(
\mathcal{H} = \{ i \mid \prod_{j=1}^{p} (1 - E_{i,j} \mod 2) = 1 \} \setminus N
\)

\( K = \{1, \ldots, h\} \setminus (\mathcal{H} \cup N) \).

**Proof.** (Zonotope) The over-approximation of all monomial variable parts in \( \mathcal{H} \) with additional independent factors removes the dependence between the dependent factors and yields the definition of a zonotope \( \mathcal{PZ} \). This zonotope encloses \( \mathcal{PZ} \) because removing dependence results in an over-approximation \( \mathcal{PZ} \). Since monomials with exclusively even exponents \( (i \in \mathcal{H}) \) are strictly positive, we can enclose them tighter using

\[
\forall i \in \mathcal{H} : \left( \prod_{k=1}^{p} (-1)^{E_{i,k}} \right) G_{i,k} = 0 \rightarrow \mathcal{H} = \frac{1}{2} G_{i,k} + [-1, 1] \left( \frac{1}{2} G_{i,k} \right).
\]

For all other monomials, evaluation of the monomial variable part directly results in the interval \([-1, 1]\).

Complexity: The calculation of the set \( \mathcal{H} \) has complexity \( \mathcal{O}(\hat{E}) \), and the construction of the zonotope \( \mathcal{O}(nh) \) in the worst case, resulting in an overall complexity of \( \mathcal{O}(\hat{E}) + \mathcal{O}(nh) \), which is equal to \( \mathcal{O}(n^2) \) using [2].
Support Function, Interval, and Template Polyhedra

Let us first derive the support function of an SPZ.

**Definition 4. (Support Function) Def. 1** Given a set \( S \subset \mathbb{R}^n \) and a direction \( d \in \mathbb{R}^n \), the support function \( s_S : \mathbb{R}^n \rightarrow \mathbb{R} \) of \( S \) is defined as

\[
s_S(d) = \max_{x \in S} d^T x.
\]

If \( S \) is convex, its support function is an exact representation; otherwise, an over-approximation is returned. Since SPZs are non-convex in general, one can only over-approximate them by support functions.

**Proposition 5. (Support Function) Given an SPZ \( \mathcal{PZ} = \{ G, G_1, E, ID \} \) and a direction \( d \in \mathbb{R}^n \), the support function \( \hat{s}_\mathcal{PZ}(d) \) over-approximates \( \mathcal{PZ} \):

\[
\hat{s}_\mathcal{PZ}(d) = u + \sum_{j=1}^q \| \mathbf{g}_{(j)} \|
\]

where \( d^T \odot \mathcal{PZ} = \{ \mathbf{g}, \mathbf{g}_{(j)} \} \) is the projection of \( \mathcal{PZ} \) onto the vector \( d \). The polynomial function \( w : \mathbb{R}^p \rightarrow \mathbb{R} \) is defined by the dependent generators as

\[
w(\alpha_1, \ldots, \alpha_p) = \prod_{k=1}^p \alpha_k^E_{(k, i)} \mathbf{g}_{(i)}.
\]

Furthermore, given a function \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) and an interval \( I \subset \mathbb{R}^m \), the range bounding operation

\[
B(f(x), I) = \left[ \min_{x \in I} f(x), \max_{x \in I} f(x) \right]
\]

returns an over-approximation of the exact bounds. The calculation of \( \hat{s}_\mathcal{PZ}(d) \) has complexity \( \mathcal{O}(n^2) + \mathcal{O}(B()) \), where \( \mathcal{O}(B()) \) denotes the computational complexity of the range bounding operation.

**Proof.** (Support Function) We first project the SPZ onto the vector \( d \), and then split the one-dimensional projected SPZ into one part with independent generators and one with dependent generators: The bounds for the independent part calculated by the sum of absolute values in [15] are exact [20] in Sec. 2. However, the upper bound \( u \) of the dependent part in [15] is over-approximative since the range bounding operation \( B() \) returns an over-approximation, so that \( \hat{s}_\mathcal{PZ}(d) \) over-approximates \( \mathcal{PZ} \).

Complexity: The calculation of the projection onto \( d \) has a complexity of \( \mathcal{O}(n^2) + \mathcal{O}(B()) \) (see Sec. 2), which results in an overall complexity of \( \mathcal{O}(n^2) + \mathcal{O}(B()) \) using [20] since all other operations have linear complexity.

Note that the tightness of \( \hat{s}_\mathcal{PZ}(d) \) solely depends on the tightness of the bounds of the function \( w(\cdot) \) obtained by one of the range bounding techniques, e.g., interval arithmetic [23] and verified global optimization [31].

A template polyhedron enclosing an SPZ can be easily constructed by evaluating the support function \( \hat{s}_\mathcal{PZ}(d) \) for a discrete set of directions \( D = \{ d_1, \ldots, d_m \} \). The over-approximation with an interval represents a special case where \( D = \{ I_{(1,1)}, \ldots, I_{(1,n)} - I_{(1,1)}, \ldots, -I_{(1,n)} \} \) with \( I \in \mathbb{R}^{n \times n} \) being the identity matrix.

### 2.4 Basic Set Operations

This subsection derives basic operations on SPZs.

**Multiplication with a Matrix**

The left-multiplication with a matrix is obtained as:

**Proposition 6. (Multiplication) Given an SPZ \( \mathcal{PZ} = \{ G, G_1, E, ID \} \subset \mathbb{R}^n \) and a numerical matrix \( M \in \mathbb{R}^{m \times n} \), the left-multiplication is computed as

\[
M \odot \mathcal{PZ} = \{ MG, MG_1, E, ID \},
\]

which has complexity \( \mathcal{O}(mn^2) \).

**Proof.** (Multiplication) The result follows directly from inserting the definition of SPZs in [1] into the definition of the operator \( \odot \) (see Notation in Sec. 1).

Complexity: The complexity results from the complexity of matrix multiplications and is therefore \( \mathcal{O}(mn^2) + \mathcal{O}(mnq) = \mathcal{O}(mn^2) \) using [2].

**Minkowski Addition**

Even though every zonotope can be represented as an SPZ, we provide a separate definition for the Minkowski addition of an SPZ and a zonotope for computational reasons. If two SPZs are involved, we first have to bring the exponent matrices to a common representation using mergeID().

**Proposition 7. (Addition) Given two SPZs, \( \mathcal{PZ}_1 = \{ G_1, G_{1,1}, E_1, ID_1 \} \) and \( \mathcal{PZ}_2 = \{ G_2, G_{1,2}, E_2, ID_2 \} \), as well as a zonotope \( Z = \{ c_z, G_z \} \), their Minkowski sum is defined as

\[
\mathcal{PZ}_1 \oplus \mathcal{PZ}_2 = \{ G_1, G_2 \}, \{ G_{1,1}, G_{1,2} \}, \{ [E_1, E_2], \{ ID_1, ID_2 \} \}, \]

and if \( \mathcal{PZ}_1 \oplus Z = \{ c_z, G_z \}, \{ G_{1,1}, G_z \}, \{ [E_1, E_2], ID_1 \} \), then

\[
\mathcal{PZ}_1 \oplus Z = \{ [c_z, G_1], \{ G_{1,1}, G_z \}, \{ [0, E_1], ID_1 \} \}
\]

where \( [19] \) has complexity \( \mathcal{O}(n^2 \log(n)) \) and \( [20] \) has complexity \( \mathcal{O}(1) \).

**Proof.** (Addition) Since the operation mergeID() is applied beforehand, \( ID_1 = ID_2 \). The result is obtained by inserting the definition of zonotopes [6] and SPZs [1] into the definition of the Minkowski sum (see Notation in Sec. 1).

Complexity: The construction of the resulting SPZs only involves concatenations and therefore has complexity \( \mathcal{O}(1) \). For two SPZs, the compact() operation with complexity \( \mathcal{O}(p_1(h_1 + h_2) \log(h_1 + h_2)) \) has to be additionally applied, resulting in an overall complexity of \( \mathcal{O}(n^2 \log(n)) \) using [2].

**Quadratic Map**

For reachability analysis based on the conservative polynomialization approach [3], a polynomial abstraction of the nonlinear dynamic function is calculated, requiring quadratic and higher order mappings. Here, we derive the equations for the quadratic map.

**Definition 5. (Quadratic Map) Theorem 1** Given a set \( S \subset \mathbb{R}^n \) and a discrete set of matrices \( Q_i \in \mathbb{R}^{n \times n}, i = 1 \ldots m \), the quadratic map of \( S \) is defined as

\[
sg(Q, S) = \left\{ x \mid x_{(i)} = s^T Q_i s, s \in S, i = 1 \ldots m \right\}.
\]
For SPZs, we first consider the special case without independent generators, and later present the general case.

**Proposition 8.** Given the SPZ \( \mathcal{PZ} = \{ \hat{G}, 0, \hat{E}, \mathcal{T}D \} \) and a discrete set of matrices \( Q_i \in \mathbb{R}^{n \times n}, i = 1 \ldots m \), the result of the quadratic map is

\[
\text{sq}(Q, \mathcal{PZ}) = \{ \mathcal{G}, 0, \hat{E}, \mathcal{T}D \}
\]

with

\[
\mathcal{E} = [\hat{E}^{(1)} \ldots \hat{E}^{(h)}], \quad \mathcal{G} = [\hat{G}^{(1)} \ldots \hat{G}^{(h)}],
\]

where

\[
\hat{E}^{(j)} = \hat{E} + \hat{E}_{\{j\}} \cdot 1, \quad \hat{G}^{(j)} = \begin{bmatrix} \hat{G}_{(j)}^{T}Q_1 \hat{G} \\ \vdots \\ \hat{G}^{(j)}_{n,m}Q_m \hat{G} \end{bmatrix}. \tag{23}
\]

The overall complexity is \( O(n^3m) \).

**Proof.** The equations are obtained directly by substitution of \( s \) in Def. of the SPZ from [1], where we exploit that \( \alpha^{e_1} \cdot \alpha^{e_2} = \alpha^{e_1+e_2} \) holds.

Complexity: The construction of the matrices \( \hat{E}^{(j)} \) has complexity \( O(h\hat{p}) \), and the construction of the matrices \( \hat{G}^{(j)} \) has complexity \( O(n^2hm) + O(n^2m^2) \) if the results for \( Q_1 \hat{G} \) are stored and reused. The resulting overall complexity is \( O(n^3m) \) using (2). \( \square \)

We now extend Prop. 8 to the general case including independent generators, for which we compute an over-approximation for computational reasons.

**Proposition 9.** (Quadratic Map) Given an SPZ \( \mathcal{PZ} = \{ G, G_1, E, \mathcal{T}D \} \subset \mathbb{R}^n \) and a discrete set of matrices \( Q_i \in \mathbb{R}^{n \times n}, i = 1 \ldots m \),

\[
\text{sq}(Q, \mathcal{PZ}) = \{ G, G_1, E, \mathcal{T}D \} \subset \mathbb{R}^n
\]

where

\[
\mathcal{G} = [G, G_1], \quad \mathcal{E} = [E \quad 0 \quad 0], \quad \mathcal{T}D = \{ \mathcal{T}D, M + 1, \ldots, M + q \},
\]

The complexity of the calculations is \( O(n^3\log(n)) + O(n^3m) \).

**Proof.** (Quadratic Map) We first introduce the extended generator and exponent matrices \( \hat{G} \) and \( \hat{E} \) as well as the extended list of identifiers \( \mathcal{T}D \):

\[
\hat{G} = [G, G_1], \quad \hat{E} = [E \quad 0 \quad 0], \quad \mathcal{T}D = \{ \mathcal{T}D, M + 1, \ldots, M + q \}
\]

Next, we calculate \( \mathcal{E} \) and \( \mathcal{G} \) according to (22). The resulting matrices are divided into one part that contains dependent factors only, and a second part that contains all remaining monomials using the following index sets:

\[
\mathcal{K} = \{ i \mid \exists j > p \mathcal{E}_{\{j\}} \neq 0 \}, \quad \mathcal{H} = \{ 1 \ldots h + q \} \setminus \mathcal{K}.
\]

Finally, the part that contains the independent factors is enclosed by a zonotope:

\[
\{ c_z, G_z \} = \text{zono}(\{ \mathcal{G}_{\{i\}}, 0, \mathcal{E}_{\{i\}}, \mathcal{T}D \}) \tag{27}
\]

Since the operation \( \text{zono}() \) is over-approximative, the SPZ constructed according to (24) encloses the quadratic map.

**Remark 1.** Since the independent generators add a zonotopic part to the SPZ, the set representing the quadratic map generally contains monomials with squared independent factors as well as monomials with products of independent factors, which can be deduced from [3, Theorem 1]. To represent the resulting set as a proper SPZ, we eliminate these monomials by enclosing them with a zonotope in (27), which results in an over-approximation of the quadratic map.

The extension to cubic or even higher-order maps of sets is straightforward and therefore omitted due to space limitations.

### 2.5 Auxiliary Operations

#### Order Reduction

Many set operations such as Minkowski addition or quadratic maps increase the number of generators, and consequently also the order \( p \) of the SPZ. Thus, for computational reasons, it is necessary to repeatedly reduce the zonotope order during reachability analysis. We propose a reduction operation for SPZs that is based on the order reduction of zonotopes (see e.g. [24]).

**Proposition 10.** (Reduce) Given an SPZ \( \mathcal{PZ} = \{ G, G_1, E, \mathcal{T}D \} \) and a desired zonotope order \( p_d \), the operation \( \text{reduce()} \) returns an SPZ with order smaller than \( p_d \) that encloses \( \mathcal{PZ} \):

\[
\text{reduce}(\mathcal{PZ}, p_d) = \left\{ \{ c_z, G_z \}, \{ G_{h,(\mathcal{H})}, G_z \}, \{ 0, E_{(\mathcal{K})} \}, \{ \mathcal{T}D \} \right\}
\]

\[
\{ c_z, G_z \} = \text{reduce}(\mathcal{Z}, 1), \quad \mathcal{Z} = \text{zono}(\{ G_{(\mathcal{K})}, G_{h,(\mathcal{H})}, E_{(\mathcal{K})}, \mathcal{T}D \}).
\]

For reduction, we select the smallest \( a = \lfloor h + q - n(p_d - 1) \rfloor \) generators:

\[
\mathcal{K} = \{ i \mid ||G_{(i)}||_2 \leq ||\hat{G}_{(\mathcal{N}(a))}||_2 \}, \quad \mathcal{H} = \{ i \mid ||G_{(i-1)}||_2 \leq ||\hat{G}_{(\mathcal{N}(a))}||_2 \},
\]

\[
\mathcal{K}' = \{ 1, \ldots, h \} \setminus \mathcal{K}, \quad \mathcal{H}' = \{ 1, \ldots, q \} \setminus \mathcal{H},
\]

with

\[
||\hat{G}_{(\mathcal{N}(a))}||_2 \leq \cdots \leq ||\hat{G}_{(\mathcal{N}(h+q))}||_2,
\]

where \( \hat{G} = [G, G_1] \) and \( N \in \{ Z_{\geq 0} \}^{h+q} \) is a discrete set of indices. The complexity is \( O(n^3) + O(\text{reduce}()) \), where \( O(\text{reduce}()) \) denotes the complexity of the zonotope reduction, which depends on the selected method.

**Proof.** (Reduce) The definition of \( a \) ensures that \( |\mathcal{K}| + |\mathcal{H}| + n + 1 \leq p_d \). Further, \( \text{reduce}(\mathcal{PZ}, p_d) \supseteq \mathcal{PZ} \) since \( \text{zono}() \) and the reduction of a zonotope \( \text{reduce}() \) are both over-approximative, and therefore \( \text{reduce}(\text{zono}()) \) is over-approximative, too.
Complexity: Sorting the generators has a complexity of $O(n(h+q)) + O((h+q) \log(h+q))$, and the enclosure with a zonotope has a worst case complexity of $O(ph) + O(nh)$. Using $O(n)$, the overall complexity is therefore $O(n^2) + O(\text{reduce}())$. \hfill \square

After reduction, we remove possibly generated all-zero rows in the exponent matrix.

**Restructure**

Due to the repeated order reduction and Minkowski addition during reachability analysis, the volume spanned by independent generators grows relative to the volume spanned by dependent generators. As explained later in Sec. 3 this has a negative effect on the tightness of the reachable sets.

We therefore define the operation $\text{reduce}()$, which introduces new dependent generators that over-approximate the independent ones:

**Proposition 11.** (Restructure) Given an SPZ $\mathcal{PZ} = \{G, G_1, E, TD\}$, $\text{reduce}()$ returns an SPZ that encloses $\mathcal{PZ}$ and removes all dependent generators:

$$\text{reduce}(\mathcal{PZ}) = \{c_e, G_e\} \cup \{0, E, \mathcal{TD}\}$$

where $c_e, G_e$ is $\text{reduce}()$ of $\{0, G_1\}$, $E = \begin{bmatrix} 0 & E & 0 \end{bmatrix}$, $\mathcal{TD} = \{GD, M + 1, \ldots, M + n\}$, $M = \max(\mathcal{TD})$.

$$\text{(30)}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. The overall complexity is $O(\text{reduce}())$, which is the complexity of the zonotope reduction.

**Proof.** The result of the $\text{reduce}()$ operation encloses the original set since $\text{reduce}()$ is over-approximative, and the redefinition of independent generators as new dependent generators just changes the set representation, but not the set in (30) itself.

Complexity: Since all other operations are concatenations and initializations, the complexity equals the one of $\text{reduce}()$. \hfill \square

We demonstrate the effectiveness of Prop. 11 by numerical examples in Sec. 4. To save computation time, we define an upper bound $p_d$ of factors for the SPZ after restructuring so that $p + q \leq p_d$ holds, where independent factors are removed first to maintain as much dependence as possible.

## 3. REACHABILITY ANALYSIS

In this section we demonstrate how SPZs can be used to improve reachability analysis for nonlinear systems. Our algorithm is based on the conservative polynomialization approach \cite{3} for nonlinear systems of the form

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

where $x$ is the vector of system states and $u$ is the input vector. The conservative polynomialization approach for reachability analysis is based on the abstraction of the nonlinear function $f(\cdot)$ by a Taylor expansion of order $k$:

$$\dot{x}(i) = f_{(i)}(z(t))$$

$$\leq \sum_{j=1}^k \left( (z(t) - z^*)^T \nabla \right) f_{(i)}(z) + L_{(i)}(t),$$

where $z^* \in \mathbb{R}^{n+m}$ is the expansion point for the Taylor series.

**Algorithm 1** reach($R(0), t_f, \ldots$)

**Require:** Initial set $R(0)$, input set $U$, time horizon $t_f$, time step $\tau$, factor $\lambda$.

**Ensure:** $R([0, t_f])$

1: $t_0 = 0$, $s = 0$, $R^{union} = \emptyset$, $U^\Delta = U \oplus (-\Delta u)$
2: while $t_s < t_f$ do
3: \hspace{1em} $\text{taylor} \rightarrow z^*, w, A, B, D, E$
4: \hspace{1em} $\Psi(t_s) = 0$
5: \hspace{1em} $\forall(t_s) = w \oplus Bu^* \oplus \frac{1}{2}q(D, R(t_s) \times U)$
6: \hspace{1em} $Z_s(t_s) = \text{zono}(R(t_s)) \times U$
7: \hspace{1em} $\text{repeat}$
8: \hspace{2em} $\Psi(t_s) = \text{enlarge}(\Psi(t_s), \lambda)$
9: \hspace{2em} $R^{\Delta}(t_s) = \text{post}^2(R(t_s), \Psi(t_s), A) \times U$
10: \hspace{2em} $R^{\Delta}(t_s) = \text{zono}(R^{\Delta}(t_s))$
11: \hspace{2em} $L_s(t_s) = \text{varInputs}(Z_s(t_s), R_s^\Delta(t_s), U^\Delta, B, D)$
12: \hspace{2em} $\Psi(t_s) = \text{lagrangeRemainder}(\Psi(t_s))$
13: \hspace{2em} $\Psi(t_s) = \text{post}(\Psi(t_s), \Delta z^*)$
14: \hspace{2em} $\Psi(t_s) = \text{post}(\Psi(t_s), \Delta z^*)$
15: \hspace{2em} until $\Psi(t_s) \subseteq \Psi(t_s)$
16: $R(t_{s+1}) = \text{post}(R(t_s), A, V(t_s), V^\Delta(t_s), L(t_s))$
17: $R(t_{s+1}) = \text{reduce}(R(t_{s+1}), p_d)$
18: \hspace{1em} if $\text{volRatio}(\text{Vol}(t_{s+1})) > \mu_d$ then
19: \hspace{2em} $R(t_{s+1}) = \text{reduce}(R(t_{s+1}))$
20: $\text{end if}$
21: $R^{union} = R^{union} \cup \text{post}(\Psi(t_s), A, \Psi(t_s), A) \times U$
22: $t_{s+1} = t_s + \tau, \quad s := s + 1$
23: $\text{end while}$
24: $R([0, t_f]) = R^{union}$

In order to fully exploit the advantages of SPZs, Alg. 1 is slightly modified from \cite{3}. We only specify the algorithm for the Taylor order $k = 2$ for simplicity, since the extension to higher orders is straightforward. The definitions of the operators $\text{taylor}$, $\text{enlarge}$, $\text{post}^2$, $\text{varInputs}$, and $\text{lagrangeRemainder}$ are identical to the ones in [3]. Only the definition of the $\text{post}$ operator changed, since we precompute some of the sets in our algorithm:

$$\text{post} \left( \left( R(t_s), A, \forall(t_s), V^\Delta(t_s), L(t_s) \right) \right) =$$

$$e^{Ar(t_s)} + \sum_{j=1}^{\infty} \left( \sum_{j=1}^{\infty} \text{post}(\Psi(t_s), A) \times U \right) + R^\Delta \left( \left( V^\Delta(t_s) \oplus L(t_s), r \right) \right),$$

where $R^\Delta(\cdot)$ is defined as in \cite{3} Eq. (9), and the definition for $\Gamma(r)$ can be found in \cite{3} Sec. 3.2. We proceed with a discussion of the main advantages resulting from using SPZs.

### 3.1 Advantages of using Sparse Polynomial Zonotopes

As mentioned earlier, one of the main advantages of SPZs is that they reduce the dependency problem in Alg. 1. We demonstrate this with a short example:

**Example 2.** Consider a one-dimensional problem without inputs: a reachable set $R(t_s) = \{\alpha_1 | \alpha_1 \in [-1, 1]\}$,
the parameter values $w = 0$, $A = 1$, $D = 2$, and $r = 1$. The quadratic map in line 3 of Alg. 4 evaluates to
\[ \frac{1}{2} \text{sgn}(D, R(t_1)) = \{a_1^2 | \alpha_1 \in [-1,1]\} \] for SPZs. On the other hand, if we use zonotopes, then the quadratic map has to be over-approximated with $\frac{1}{2} \text{sgn}(D, R(t_1)) = \{0.5 + 0.5a_2^2 | \alpha_2 \in [-1,1]\}$. With zonotopes, we therefore obtain for

\[ F_1 \oplus F_2 = \{2.718a_1 + 1.718(0.5 + 0.5a_2) \mid a_1, a_2 \in [-1,1]\} = [-2.718, 4.436]. \tag{34} \]

With SPZs, however, we obtain the exact set

\[ F_1 \oplus F_2 = \{2.718a_1 + 1.718a_2^2 | \alpha_1 \in [-1,1]\} = [-1.075, 4.436]. \tag{35} \]

Using zonotopes for reachability analysis therefore leads to a significant over-approximation error in each time step. A similar problem occurs with the polynomial degree in advance.

With our new SPZ representation, it is in theory even possible to approximate the exact reachable set arbitrarily close:

**Theorem 2.** Let us consider the case without uncertain inputs $U = \emptyset$, with parameter values $\mu_d = \infty$, $\mu_d = 0$, and $f(\cdot)$ being a $C^\infty$ differentiable function. In addition, $\kappa$ is chosen large enough to ensure that $L(\tau) \rightarrow 0$. We further assume that the enlargement factor $\lambda$ in line 3 of Alg. 4 is always chosen such that $\Psi(\tau) = \Psi(\tau_1)$ holds, and that the restructure operation does not result in an over-approximation, which can easily be achieved by omitting the reduction step. The reachable set computed with SPZs would then converge to the exact reachable set for $r \rightarrow 0$.

**Proof.** If $\mu_d = 0$, we execute the restructure operation in every time step. The set $\mathcal{R}(t_1)$ is therefore an SPZ without independent generators, which implies that the quadratic map does not result in an over-approximation. Consequently, the only operation that leads to an over-approximation is the Minkowski addition of the set $\mathcal{R}^{p, \Delta}(\cdot)$ during the evaluation of the post operator as shown in (32). This over-approximation error converges to zero for $r \rightarrow 0$, since per definition $\mathcal{R}^{p, \Delta}(\tau) \rightarrow 0$ for $r \rightarrow 0$ (Sec. 6, Eq. (6)), and therefore $\mathcal{R}^{p, \Delta}(\cdot) \rightarrow 0$ for $r \rightarrow 0$. \qed

3.2 Hybrid Systems

In reachability analysis for hybrid systems, the main difficulty is the calculation of the intersection between the reachable set and the guard sets. Since it is in general computationally infeasible to calculate this intersection directly for SPZs, we propose calculating the intersection with a zonotope over-approximation instead. By doing so it is possible to directly apply the well-developed techniques for the computation of guard intersections with zonotopes, like e.g., the ones from [21] or [7]. Note that even if the intersections with the guard sets are calculated with a zonotope over-approximation, the reachable set of the hybrid system calculated with SPZs is generally much tighter than the one calculated with zonotopes (see Sec. 4).

4. NUMERICAL EXAMPLES

In this section we demonstrate the improvements to reachability analysis due to using SPZs on two benchmark systems. All computations are carried out in MATLAB on a 2.9GHz quad-core i7 processor with 32GB memory. We heuristically trigger the restructure process (see Sec. 2.5) when $\text{volRatio}(P Z) > \text{vol}(\text{interval}(G, G_1))$, where $P Z = \{G, G_1, E, ID\}$ and $\text{vol}(\cdot)$ calculates the volume of a multi-dimensional interval.

The system considered first is the Van-Der-Pol oscillator taken from the 2018 ARCH competition [22]:

\[ \dot{x}_1 = x_2, \]
\[ \dot{x}_2 = (1 - x_1^2)x_2 - x_1. \tag{36} \]

For this system, we compare the results for the computation of the reachable set with Alg. 4 using zonotopes, the quadratic zonotopes from [3] and our SPZ representation. We consider the initial set $x_1 \in [1.23, 1.57]$ and $x_2 \in [2.34, 2.46]$, and we use a time step size of $r = 0.005$ seconds, a maximum zonotope order of $p_d = 50$, an enlargement factor of $\lambda = 0.1$, a maximum volume ratio of $\mu_\Delta = 0.01$, and an upper bound for the number of dependent factors of $\mu_d = 100$. The method in [19, Sec. 3.4] (Girard’s method) is applied for zonotope reduction, and we use principal-component-analysis-based order reduction in combination with the Girard’s method for the reduction during the restructure operation (see [25]). For a fair comparison, we use the same parameter values for every set representation.

The resulting reachable sets are shown in Fig. 3. It is clearly visible that the stability of the limit cycle can only be verified with SPZs when sets are not split. The computation time is 9.33 seconds for linear zonotopes, 13.38 seconds for quadratic zonotopes, and 16.52 seconds for SPZs.

An impression on how tight the reachable set can be over-approximated with SPZs is provided in Fig. 4 where the reachable set after $t = 3.15$ seconds computed with a time step size of $r = 0.001$ seconds and a maximum volume ratio of $\mu_\Delta = 0.001$ is compared to the exact reachable set of the system. The figure also demonstrates how well the SPZ approximates the shape of the exact reachable set.

For the second numerical example, we examine a drive-train [27], which is a benchmark from the ARCH 2018 competition [4], too. We consider the case with 2 rotating masses, resulting in a system dimension of $n = 11$. The
model is a hybrid system with linear dynamics. However, we apply the novel approach from [9] for calculating the intersections with guard sets, which is based on time-triggered conversion of guards and results in a significant nonlinearity due to the time scaling process. The initial set is given by \( R_0 = 0.5(X_0 - \text{center}(X_0)) + \text{center}(X_0) \), where \( X_0 \) is defined as in [4], and we consider the same extreme acceleration maneuver as in [4]. As a specification, we require that the engine torque after 1.5 seconds is at least 59 Nm, which can formally be specified as \( T_m \geq 59 Nm \forall t \geq 1.5s \).

The results for the drivetrain model are shown in Fig. 4. We explicitly considered the possibility of splitting the reachable sets, so that the specification could be verified with all set representations. However, splitting sets prolongs the computation time: with quadratic zonotopes, the verification took 93 seconds, and 221 seconds with zonotopes. Only with SPZs it was possible to verify the specification without splitting, resulting in a computation time of 15 seconds, which is 6 times faster than with quadratic zonotopes and more than 14 times faster than with zonotopes. Compared to other non-zonotopic set representations, the speed-up is even larger: in the results for this benchmark from the ARCH-18 competition [4], the tool CORA, which uses zonotopes to represent reachable sets, had for the high dimensional test cases with large initial set the smallest computation time compared to the other participating tools Flow* and Hylaa, which use Taylor models and star sets, respectively.

5. CONCLUSIONS

We have introduced sparse polynomial zonotopes, a new non-convex set representation. The sparsity results in several advantages compared to previous representations of polynomial zonotopes: sparse polynomial zonotopes enable a compact representation of sets, they are closed under all relevant set operations, and all operations have at most polynomial complexity. The fact that sparse polynomial zonotopes include several other set representations like Taylor models and zonotopes further substantiates the relevance of the new representation. One application for sparse polynomial zonotopes is reachability analysis for nonlinear systems. Our improved reachability algorithm exploits the advantages of sparse polynomial zonotopes. The numerical examples demonstrate that our approach indeed computes much tighter over-approximations of reachable sets compared to zonotopes and quadratic zonotopes. Due to the improved accuracy, splitting can be avoided by using sparse polynomial zonotopes, which results in a significant reduction of the computation time since the complexity of splitting sets grows exponentially with the system dimension.

6. REFERENCES

[1] T. Alamo, A. Cepeda, and D. Limon. Improved computation of ellipsoidal invariant sets for saturated control systems. In Decision and Control and European Control Conference. CDC-ECC’05, pages 6216–6221. IEEE, 2005.
[2] M. Althoff. Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars. Dissertation, Technische Universität München, 2010.
[3] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In Hybrid Systems: Computation and Control, pages 173–182, 2013.
[4] M. Althoff and et al. ARCH-COMP18 category report: Continuous and hybrid systems with linear continuous dynamics. In ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems, 2018.
[5] M. Althoff and G. Frehse. Combining zonotopes and support functions for efficient reachability analysis of linear systems. In Proc. of the 55th IEEE Conference on Decision and Control, pages 7439–7446, 2016.
[6] M. Althoff and B. H. Krogh. Zonotope bundles for the efficient computation of reachable sets. In Proc. of the 50th IEEE Conference on Decision and Control, pages 6814–6821, 2011.
[7] 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.
[8] M. Althoff and B. H. Krogh. Reachability analysis of nonlinear differential-algebraic systems. IEEE Transactions on Automatic Control, 59(2):371–383, 2014.
[9] S. Bak, S. Bogomolov, and M. Althoff. Time-triggered conversion of guards for reachability analysis of hybrid automata. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 133–150. Springer, 2017.

[10] S. Bak and P. S. Duggirala. HyLAA: A tool for computing simulation-equivalent reachability for linear systems. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 173–178, 2017.

[11] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.

[12] J. M. Bravo, T. Alamo, and E. F. Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42:1745–1751, 2006.

[13] 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.

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

[15] P. S. Duggirala and M. Viswanathan. Parsimonious, simulation based verification of linear systems. In *Proc. of International Conference on Computer Aided Verification*, pages 477–494, 2016.

[16] A. Eggers, N. Ramdani, N. S. Nedialkov, and M. Fränzle. Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods. *Software & Systems Modeling*, 14(1):121–148, 2012.

[17] G. Frehse, S. Bogomolov, M. Greitschus, T. Strump, and A. Podelski. Eliminating spurious transitions in reachability with support functions. In *Proc of Hybrid Systems: Computation and Control*, pages 149–158, 2015.

[18] G. Frehse and et al. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, LNCS 6806, pages 379–395. Springer, 2011.

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

[20] 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.

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

[22] F. Immler and et al. ARCH-COMP18 category report: Continuous and hybrid systems with nonlinear dynamics. In *ARCH18, 5th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2018.

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

[24] S. Kaynama, J. N. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the viability kernel using maximal reachable sets. In *Proc. of Hybrid Systems: Computation and Control*, pages 55–64, 2012.

[25] A.-K. Kopetzki, B. Schürmann, and M. Althoff. Methods for order reduction of zonotopes. In *Proc. of the 56th IEEE Conference on Decision and Control*, pages 5626–5633, 2017.

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

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

[28] B. Legat, R. M. Jungers, and P. A. Parrilo. Computing controlled invariant sets for hybrid systems with applications to model-predictive control. *arXiv preprint arXiv:1802.04522*, 2018.

[29] R. Lohner. *Perspectives on Enclosure Methods*, chapter On the Ubiquity of the Wrapping Effect in the Computation of the Error Bounds, pages 201–217. Springer, 2001.

[30] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003.

[31] K. Makino and M. Berz. Verified global optimization with Taylor model based range bounders. *Transactions on Computers*, 4(11):1611–1618, 2005.

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

[33] M. Rungger and P. Tabuada. Computing robust controlled invariant sets of linear systems. *IEEE Transactions on Automatic Control*, 62(7):3665–3670, 2017.

[34] S. Schupp, E. Abraham, I. B. Makhlouf, and S. Kowalewski. Hypro: A C++ library of state set representations for hybrid systems reachability analysis. In *NASA Formal Methods Symposium*, pages 288–294. Springer, 2017.

[35] B. Schürmann and M. Althoff. Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space. In *Proc. of the 20th World Congress of the International Federation of Automatic Control*, pages 11515–11522, 2017.

[36] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136, 2016.

[37] H. R. Tiwary. On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete and Computational Geometry*, 40:469–479, 2008.

[38] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.