# Methods for Order Reduction of Zonotopes

Anna-Kathrin Kopetzki, Bastian Schürmann, and Matthias Althoff

***Abstract*— Zonotopes are a special subclass of polytopes, which have several favorable properties: They can be represented in a compact way and they are closed under the Minkowski sum as well as under linear transformations. Zonotopes are a popular set representation used e.g. for reachability analysis of dynamic systems, set-based observers and robust control. The complexity of algorithms that work on zonotopes strongly depends on their order (i.e. their number of generators and dimensions), which is often increased by operations like the Minkowski sum. Thus, to keep computations efficient, zonotopes of high orders are often over-approximated as tight as possible by zonotopes of smaller order. This paper has two main contributions: First, we propose new methods based on principle component analysis (PCA), clustering and constrained optimization for tight over-approximation of zonotopes. Second, we provide an overview of the most important known methods for order reduction and compare the performance of new and known methods in low- and high-dimensional spaces.**

## I. INTRODUCTION

Zonotopes are point-symmetric sets in $n$-dimensional space that can be represented in a very compact way: A zonotope $Z$ is defined by a center $c \in \mathbb{R}^n$ and $p$ generators $g^{(i)} \in \mathbb{R}^n, i \in \{1,\dots,p\}$ stored as columns of a matrix $G \in \mathbb{R}^{n\times p}$ as

$$Z = (c \mid G) := \left\{ c + \sum_{i=1}^{p} \beta_i g^{(i)} : \beta_i \in [-1,1] \right\}. \quad (1)$$

Figure 1 illustrates the construction of a zonotope, consisting of three 2-dimensional generators.

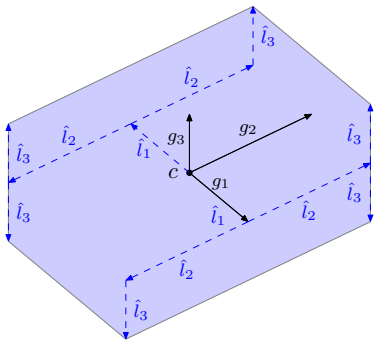


Fig. 1. Construction of zonotope $Z = (c \mid [g_1, g_2, g_3])$ according to (1) by adding line segments: $c \oplus \hat{l}_1 \oplus \hat{l}_2 \oplus \hat{l}_3$, where $\hat{l}_i = [-1,1] \cdot g_i$.

Zonotopes have the favorable property that they are closed under the Minkowski sum. Thus, the sum of two zonotopes $Z_1 = (c_1 \mid G_1)$ and $Z_2 = (c_2 \mid G_2)$ is a zonotope and can be computed as $Z_1 \oplus Z_2 = (c_1 + c_2 \mid [G_1\ G_2])$. Furthermore, zonotopes are closed under linear transformations $AZ = (Ac \mid AG)$ [2], [23].

Due to these properties, zonotopes have become increasingly popular in several areas. They are used to compute the reachable states of dynamic systems for safety verification [2], which is important in autonomous driving [5], human-robot collaboration [26] and smart grids [3]. Other applications are rigorous state estimation of linear and nonlinear systems [1], [12], [13], systems with uncertain parameters [25], and state estimation for model predictive control [10], [24]. Furthermore, zonotopes are used as a set representation in automated theorem provers [22], as abstract domains in abstract interpretation for static program analysis [19], [17], and as bounding volumes for fast collision detection algorithms [21].

In addition, zonotopes are studied in the field of computational geometry. They are associated with higher Bruhat orders [14]. Zonotopes with large 2D-cuts are investigated in [27]. A technique to approximate a zonoid by a zonotope in fixed directions is presented in [11] and the problem of approximating a ball by a Minkowski sum of segments with equal length is addressed in [9]. An algorithm that solves the fixed rank convex quadratic maximization in binary variables by constructing parallel zonotopes is proposed in [15]. In [6] coherence and enumeration of tilings of 3-zonotopes is analyzed, while in [16], the problem of listing all extreme points of a zonotope is studied.

The complexity of algorithms that work on zonotopes depends and increases with the order of the zonotopes. The order of a $n$-dimensional zonotope $Z$ that is described by $p$ generators is $\mathrm{ord}(Z) = \frac{p}{n}$. Operations like the Minkowski sum that increase the order of a zonotope are repeatedly used during reachability analysis, for example. Thus, to keep computations efficient, $Z$ is often over-approximated by another zonotope $Z_{\mathrm{red}}$ of smaller order: $Z \subseteq Z_{\mathrm{red}}$. It is important that this over-approximation is as tight as possible to ensure that reachability analysis still provides good results [2], [28].

Currently, there are only a few methods known for order reduction. We provide a short overview of these methods. Furthermore, we propose new fast techniques for order reduction, yielding in tight over-approximations especially in spaces with more than two dimensions.

Subsequently, we focus on methods that over-approximate a zonotope $Z$ by another zonotope $Z_{\mathrm{red}}$ of order 1. The reason for this is that each method can be easily extended to reduce to a higher order. This extension approach works

A. Kopetzki, B. Schürmann, M. Althoff are with the Faculty of Computer Science, Robotics and Embedded Systems, Technical University Munich, 85748 Garching, Germany anna.k.kopetzki@tum.de, bastian.schuermann@in.tum.de, althoff@in.tum.de

the same for all presented methods and is described for one technique in section II-A [18]. Thus, reducing to order 1 most obviously illustrates the different performance of the presented methods.

## II. Known Order Reduction Methods

### A. Box Method

The box method is a simple and fast technique for over-approximating a zonotope $Z = (c \mid G) = \left(c \mid g^{(1)}, g^{(2)}, \dots, g^{(p)}\right)$ by $Z_{\text{red}}$. This reduction method sorts the generators according to

$$||g^{(1)}||_1 - ||g^{(1)}||_\infty \leq \dots \leq ||g^{(p)}||_1 - ||g^{(p)}||_\infty \quad (2)$$

where $||g^{(i)}||_1 = \sum_{k=1}^{n} |g_k^{(i)}|$ is the $L_1$-norm and $||g^{(i)}||_\infty = \max_k |g_k^{(i)}|$ is the infinity norm [18]. The value $g_k^{(i)}, i \in \{1, \dots, p\}, k \in \{1, \dots, n\}$ refers to the $k$-th element of the $i$-th generator of G ($G_{ik}$). If the order of the desired over-approximating zonotope $Z_{\text{red}}$ is greater than 1, $Z$ is split into two zonotopes $Z_1 = \left(0_n \mid g^{(1)}, \dots, g^{(q)}\right)$ and $Z_2 = \left(c \mid g^{(q+1)}, \dots, g^{(p)}\right)$, such that $Z_1$ contains the smallest $q$ and $Z_2$ contains the largest $(\text{ord}(Z) - 1)n$ generators (with respect to (2), where $0_n$ is a vector of $n$ zeros). $Z_1$ is over-approximated by any of the methods presented in the following. One possibility to over-approximate $Z_1$ is by computing its interval hull as proposed by Kühn in [23]:

$$\text{IH}(Z_1) := [c - \delta g, c + \delta g], \quad \delta g = \sum_{i=1}^{q} |g^{(i)}|. \quad (3)$$

The reduced zonotope $Z_{\text{red}}$ is the Minkowski sum of $Z_1$ and $Z_2$:

$$Z_{\text{red}} = Z_1 \oplus Z_2 = \left(c \mid \left[\text{diag}\left(\delta g\right), g^{(q+1)}, \dots, g^{(p)}\right]\right), \quad (4)$$

where $\text{diag}\left(\delta g\right)$ returns a diagonal matrix with $\delta g$ as diagonal elements.

A variant heuristic to sort the generators before splitting is by using the L2-norm $||g^{(i)}||_2 = \sqrt{\sum_{k=1}^{n} \left(g_k^{(i)}\right)^2}$ as proposed in [12].

### B. The Transformation Method

Another order reduction method which over-approximates a zonotope $Z$ by a parallelotope, i.e. a zonotope of order 1, is proposed by Althoff in [2]. The approach is based on the idea of linearly transforming a zonotope $Z$ by a transformation matrix $A$ such that its shape becomes similar to a box (i.e. an axes-aligned parallelotope). The transformed zonotope is over-approximated by its box-shaped interval hull. Then, the interval hull is transformed back into the original space to obtain an over-approximating parallelotope $Z_{\text{red}} = A \cdot \text{IH}(A^{-1}Z)$ as illustrated in Figure 2. The most challenging part of this approach is to find a linear transformation matrix $A$ that transforms $Z$ into a box-like zonotope.

The transformation method as proposed in [2] uses a subset of $n$ generators of the original zonotope $Z$ as transformation matrix. This choice of $A$ ensures that the reduced zonotope touches the facets of the original zonotope. The suggested methods in [2] for choosing this subset of generators are:

- The method $\text{ExSe}_y$ uses a parameter $y \in \{n, \dots, p\}$ and performs an exhaustive search among the $y$ longest generators (according to the L2-norm) of $Z$ to find the combination of generators that results in $Z_{\text{red}}$ with the smallest volume. The number of possible combinations is $\binom{y}{n}$ and thus in $\mathcal{O}(y^n)$ (see Algorithm 1) [2].
- The method $\text{NSe}_{y,z}$ uses two parameters $y \in \{n, \dots, p\}$ and $z \in \{n, \dots, y\}$. $\text{NSe}_{y,z}$ normalizes the generator matrix $G$: For each dimension $i \in \{1, \dots, n\}$, the length $\text{len}_i$ of the interval in which all values $G_{ik}$ ($k \in \{1, \dots, p\}$) are located is computed. Then each value $G_{ik}$ is divided by the corresponding $\text{len}_i$. Next, it determines the $y$ longest generators according to their L2-norm. Among these $y$ generators, the $z$ combinations that maximize $|\det\left(g^{(i1)}, \dots, g^{(in)}\right)|$ are chosen. Thereafter, an exhaustive search for the best combination of generators on the selected $z$ combinations is performed. Each generator of the final combination is normalized (divided by its L2-norm) [4].

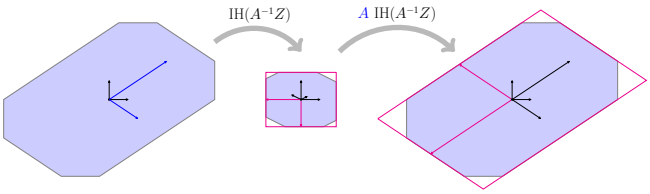


Fig. 2. Over-approximation of zonotope $Z$ (blue) of order 2 by $Z_{\text{red}}$ (magenta) of order 1 using the transformation method [2].

**Algorithm 1:** Over-approximation by $\text{ExSe}_y$, see [2]

**Input:** Zonotope $Z = (c \mid G)$, $y \in \mathbb{N}$
**Output:** Zonotope $Z_{\text{red}}$

```
Select the longest n + y generators of Z: G_y
for all possile matrices A containing n columns of G_y do
    Z_cand = A · IH(A^{-1} Z_cand)
    if volume(Z_cand) < volume(Z_red) then
        Z_red = Z_cand
    end
end
```

## III. New Order Reduction Methods

In this section, we describe new methods to over-approximate a zonotope $Z$ of order greater than 1 by a zonotope $Z_{\text{red}}$ of order 1. These methods can be extended to reduce $Z$ to a zonotope of higher order than 1 as described in section II-A for the box method.

The first and second technique (principle component analysis and line clustering) are based on the transformation

method and propose new and efficient approaches to determine a transformation matrix $A$. Finding such a matrix $A$ that transforms $Z$ into a box-like zonotope is the most challenging part of this order reduction approach. The third and fourth method (direct constrained optimization and SVD-based constrained optimization) directly compute the generator matrix of the reduced zonotope. The evaluation and performance of all methods are discussed in section IV.

### A. Principle Component Analysis (PCA)

A $n$-dimensional zonotope $Z = (c \mid G)$ that is described by $p > n$ generators can be imagined as the mapping of a $p$-dimensional hypercube to $n$-dimensions. Over-approximating $Z$ by $Z_{\text{red}}$ can be considered as a reduction problem, i.e. describing the mapped high-dimensional hypercube by $n$-generators such that the covariance between the new generators is zero (as detailed below).

A well known solution for such reduction problems is principle component analysis (PCA), which we employ to find a transformation matrix as shown in Algorithm 2.

**Algorithm 2:** Over-approximation by PCA

**Input:** Zonotope $Z = (c \mid G)$
**Output:** Zonotope $Z_{\text{red}}$
1 $X = [G \ -G]^T$
2 Covariance: $C_o = X^T X$
3 Decomposition: $USV^T = C_o$
4 $Z_{\text{red}} = U \cdot \text{IH}\left(U^T Z\right)$

First, we computed a generator matrix $X$ (line 1) such that the mean vector over $X$ is $\bar{x} = \sum_{i=1}^{2p} x^{(i)} = 0_n$. Then we compute the covariance $C_o$ between the generators of $X$ (line 2) and decompose $C_o$ using singular value decomposition (SVD, line 3). The $n$ most important, linear independent directions are the eigenvectors that correspond to the $n$ largest eigenvalues of the covariance matrix, i.e. the columns of $U$. Thus, we use $U$ as the transformation matrix. As $U$ is orthogonal ($U^{-1} = U^T$), the computation of $Z_{\text{red}}$ does not require a possibly time-consuming or numerically instable matrix inversion anymore (line 4). This makes the PCA-based reduction technique fast, numerically stable and applicable to high-dimensional zonotopes of any order.

### B. Line Clustering (LineCl)

Another quite intuitive way to determine a transformation matrix $A$ is clustering. Clustering groups the $p$ generators of zonotope $Z = (c \mid G)$ into $n$ groups (clusters) and computes a representative vector for each cluster (see Figure 3). These representative vectors are used as columns of the transformation matrix.

We consider two generators as similar if they point in the same direction or in the opposite direction, because a zonotope $Z = (c \mid G)$ is point-symmetric with respect to $c$. Thus, each generator $g \in G$ can be replaced by $-g$ without changing $Z$.

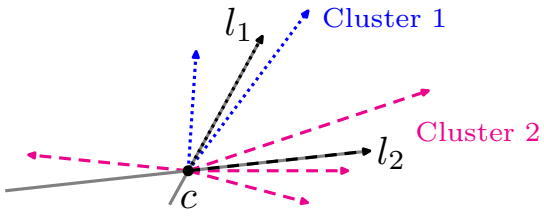


Fig. 3. Clustering of 6 generators into 2 clusters (dotted blue and dashed magenta), each represented by a vector: $l_1$ and $l_2$ (dotted and dashed gray-black)

Considering these similarities, we propose to represent a cluster $k$ by a line that runs through the center of the zonotope. Such a line can be described by an aligned vector $l_k$ (see Figure 3). The distance between the line represented by $l_k$ and a generator $g$ is computed as the orthogonal distance between the endpoint of $g$ and the line represented by $l_k$ (as illustrated in Figure 4):

$$d(l_k, g) = \left\| g - \frac{l_k^T g}{\|l_k\|_2} l_k \right\|_2 \tag{5}$$

This distance depends on the angle between $l_k$ and $g$ and on the length of $g$.

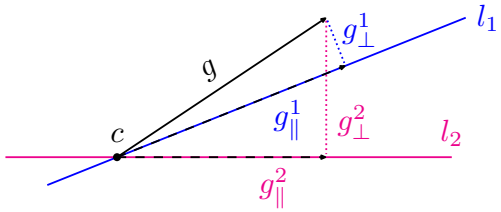


Fig. 4. Minimizing the distance of a vector from a line is equivalent to maximizing the length of its projection on the line.

For each cluster $k$ we want to compute a representative line described by a vector $l_k$. This vector is computed such that it minimizes the sum of squared distances between the line it represents and the generators that are assigned to cluster $k$.

To this end, we show that minimizing the distance between a generator $g$ and a line segment, as defined in (5), is equivalent to maximizing the length of the projection of $g$ on the line: Let $g = g_{||} + g_\perp$, where $g_\parallel$ is the projection of $g$ on the line and $g_\perp$ is orthogonal to the line (see Figure 4). Due to the Pythagorean theorem $|g|^2 = |g_{||}|^2 + |g_\perp|^2$ and the fact that the length of $g$ is constant, it follows that minimizing the length of $g_\perp$ is equivalent to maximizing the length of $g_{||}$.

This maximization problem is well known and can be solved by using singular value decomposition (SVD): The vector $l_k$ describing the desired line for cluster $k$ is the eigenvector that corresponds to the largest eigenvalue of $G_k$ (where the columns of $G_k$ are the generators of cluster $k$) and $l_k$ is the first column of $U_k$ [8, p. 38ff, chapter 3]:

$$\begin{aligned} G_k &= U_k S_k V_k^T \\ l_k &= U_k \cdot e_1^T \end{aligned} \tag{6}$$

where $e_1$ is the unit vector with a 1 at position 1.

To over-approximate a zonotope $Z = (c \mid G)$ by clustering we use Algorithm 3 (LineCl), which is based on algorithms for K-means [7, p. 423ff, chapter 9].

**Algorithm 3:** Over-approximation by LineCl

**Input:** Zonotope $Z = (c \mid G)$, $\xi > 0$, no. of runs r
**Output:** Zonotope $Z_{\text{red}}$

1 **for** $t = 1 : r$ **do**
2 Initialize $L \in \mathbb{R}^{n \times n}$ using $n$ randomly selected columns of $G \in \mathbb{R}^{n \times p}$
3 **while** $||L - L_{\text{old}}||_2 > \xi$ **do**
4 **Assignment:**
5 **for** *Generator* $g$ *of* $Z$ **do**
6 $g$ is in cluster $\arg\min_k d(l_k, g)$, see eq. (5)
7 **end**
8 **Update cluster representatives:**
9 Columns $l_k$ of $L$, see eq. (6)
10 $L_{\text{old}}$ = L
11 **end**
12 Length assignment: $l_k = l_k \frac{1}{\text{no generators in cluster k}} \sum_{g \in \text{cluster } k} |g|$
13 $Z_{\text{cand}} = (c \mid L \cdot \text{IH}(L^{-1} Z))$
14 **if** $V(Z_{\text{cand}}) < V(Z_{\min})$, *see eq.* (7) **then**
15 $L_{\min} = L$
16 $Z_{\min} = Z_{\text{cand}}$
17 **end**
18 **end**
19 $Z_{\text{red}} = L_{\min} \cdot \text{IH}\left(L_{\min}^{-1} Z\right)$

We initialize the cluster representatives using $n$ randomly selected generators of $G$ and store them as columns of matrix $L$ (line 2). The random initialization of cluster representatives can lead to non-reproducible results, thus we do the same as the classical Kmeans algorithm: We run algorithm 3 several times and keep the best result. Then, we repeat the assignment and update step of Algorithm 3 until achieving convergence, i.e. until the cluster representatives do not change any more: First, each generator $g$ is assigned to the cluster from which representative it has minimal distance (line 6). Second, the cluster representatives are recomputed according to (6) (line 10). When Algorithm 3 has converged, we assign each vector $l_k$ a length (line 12). After each run of Algorithm 3, the volume [20] of the current reduced candidate zonotope $Z_{\text{cand}} = (c \mid G_{\text{cand}}) = L \cdot \text{IH}(c \mid L^{-1} Z)$ is computed (line 14):

$$V(Z_{\text{cand}}) = 2^n |\det(G_{\text{cand}})| \tag{7}$$

Finally, the transformation matrix $L_{\min}$ that leads to $Z_{\min}$ with minimal volume is used to compute the over-approximating zonotope $Z_{\text{red}}$ (line 19).

### C. HybridPC

In addition, we have implemented a hybrid technique consisting of LineCl (line clustering) and PCA, called HybridPC. HybridPC executes both methods, evaluates the volume of the obtained reduced zonotope $Z_{\text{red}}$ and returns $Z_{\text{red}}$ with minimal volume.

We combined these two techniques because their strengths complement each other: In most cases LineCl yields tight over-approximations. However, there are some rare cases in which it returns unnecessarily large over-approximations. PCA-based over-approximation is robust, i.e. shows a constantly good performance but does not return the tightest possible over-approximation.

### D. Constrained Optimization

Another way to over-approximate a zonotope $Z$ by a zonotope $Z_{\text{red}}$ of smaller order is to formulate this problem as a constrained optimization problem.

*1) Direct Optimization (CoOptdir):* We want to obtain a tight over-approximation of a zonotope $Z = (c \mid G)$, $G \in \mathcal{R}^{n \times p}$ by $Z_{\text{red}} = (c \mid C)$, $C \in \mathbb{R}^{n \times n}$. To this end, we need to compute the generator matrix $C$. To ensure that $Z_{\text{red}}$ is as tight as possible, we compute $C$ by minimizing over the volume [20] of $Z_{\text{red}}$:

$$\arg\min_C V(Z_{\text{red}}) = \arg\min_C 2^n |\det(C)| = \arg\min_C |\det(C)|. \tag{8}$$

$Z_{\text{red}}$ must be an over-approximation of $Z$, i.e. $Z \subseteq Z_{\text{red}}$. If we transform both zonotopes by $C^{-1}$ and use that the identity matrix $I_n$ is the generator matrix of an axis-aligned box:

$$\begin{aligned} C^{-1} Z_{\text{red}} &= \left(C^{-1} c \mid C^{-1} C\right) = \left(C^{-1} c \mid I_n\right) \\ &= \text{IH}\left(C^{-1} c \mid I_n\right) = \text{IH}\left(C^{-1} Z_{\text{red}}\right) \end{aligned} \tag{9}$$

As $C^{-1} Z \subseteq \text{IH}\left(C^{-1} Z\right)$ and zonotopes are closed under linear transformations it holds that if $\text{IH}\left(C^{-1} Z\right) \subseteq \text{IH}\left(C^{-1} Z_{\text{red}}\right)$ then $Z \subseteq Z_{\text{red}}$. Thus, the constraint $Z \subseteq Z_{\text{red}}$ can be encoded by:

$$\begin{aligned} &\text{IH}\left(C^{-1} c \mid C^{-1} G\right) \subseteq \text{IH}\left(C^{-1} c \mid I_n\right) \\ &\Rightarrow \forall i \in \{1, \ldots, n\} : \sum_{j=1}^{p} |C^{-1} G|_{ij} \leq 1 \end{aligned} \tag{10}$$

where we assume that both zonotopes have the same origin. The problem of computing an over-approximating zonotope $Z_{\text{red}} = (c \mid C)$ of $Z = (c \mid G)$ by minimizing $\arg\min_C V(Z_{\text{red}})$ (equation 8) subject to $Z \subseteq Z_{\text{red}}$ (equation 10) is a constrained non-linear optimization problem. We solve this problem using the MATLAB fmincon solver and the interior-point algorithm. As an initial solution of the optimization, we use the zonotope obtained by over-approximating $Z$ by the PCA-based method described in section III-A.

*2) SVD-Based Optimization (CoOptSVD):* A disadvantage of CoOptdir is that we need to compute the inverse of the generator matrix for initialization and to obtain the generator matrix of the over-approximation zonotope $Z_{\text{red}}$. This might become numerically instable in some cases. Thus, we propose a SVD-based optimization method that we call CoOptSVD, which does not need to invert matrices.

CoOptSVD initializes the generator matrix $C \in \mathbb{R}^{n\times n}$ of $Z_{\text{red}}$ using PCA. Then it decomposes it through SVD: $USV^T = C$ where $U$ and $V$ are orthogonal and $S$ is a diagonal matrix. We still compute $C$ by minimizing over the volume, as shown below:

$$\begin{aligned}\arg\min_C V(Z_{\text{red}}) &= \arg\min_C 2^n |\det(C)| \\ &= \arg\min_C |\det(USV^T)| \\ &= \arg\min_S \sum_{i=1}^{n} \ln(|S_{ii}|)\end{aligned} \tag{11}$$

We exploit the facts that $\det(AB) = \det(A)\det(B)$ and $U$, $V$ are orthogonal, which means that their determinants are $\pm 1$. Furthermore, minimizing a function is equivalent to minimizing the logarithm of this function. The inverse of $C$ can be computed using the fact that $U$ and $V$ are orthogonal ($U^{-1} = U^T$ and $V^{-1} = V^T$) and $S$ is diagonal ($S^{-1} = \text{diag}(s_{11}, s_{22}, \dots s_{nn})^{-1} = \text{diag}(\frac{1}{s_{11}}, \frac{1}{s_{22}}, \dots, \frac{1}{s_{nn}})$):

$$\begin{aligned}C^{-1} &= (USV^T)^{-1} \\ &= \left(V^T\right)^{-1} \text{diag}(s_{11}, s_{22}, \dots s_{nn})^{-1} \left(U^T\right)^{-1} \\ &= V \text{diag}\left(\frac{1}{s_{11}}, \frac{1}{s_{22}}, \dots, \frac{1}{s_{nn}}\right) U^T\end{aligned} \tag{12}$$

Thus, we do not need to invert matrices which makes this optimization method numerically stable. The constraints of this optimization problem are

$$\begin{aligned}&U^TU = I_n \quad \text{and} \quad V^TV = I_n, \\ &\text{IH}\left(0_n \mid V\text{diag}\left(\frac{1}{s_{11}}, \frac{1}{s_{22}}, \dots, \frac{1}{s_{nn}}\right) U^TG\right) \subseteq \text{IH}\left(0_n \mid I_n\right) \\ &\Rightarrow \forall i \in \{1, \dots, n\} : \\ &\quad \sum_{j=1}^{p} \left| V\text{diag}\left(\frac{1}{s_{11}}, \dots, \frac{1}{s_{nn}}\right) U^TG \right|_{ij} \leq 1.\end{aligned} \tag{13}$$

We solve this constrained non-linear optimization problem as shown in Algorithm 4, using the MATLAB fmincon solver and the interior-point algorithm.

**Algorithm 4:** Over-approximation by CoOptSVD

**Input:** Zonotope $Z = (c \mid G)$, $G \in \mathbb{R}^{n\times p}$
**Output:** Zonotope $Z_{\text{red}} = (c \mid C)$, $C \in \mathbb{R}^{n\times n}$
**Initialize** C using Algorithm 2
Decomposition: $C = USV^T$
**while** *not converged* **do**
  **Minimize**: $\arg\min_C V(Z_{\text{red}})$ (11) **Subject to**: (13)
**end**
Compute $USV^T = C$
$Z_{\text{red}} = (c \mid C)$

## IV. Comparison of Order Reduction Methods

To evaluate the performance of the presented methods, we have implemented all techniques described above in MATLAB as part of the CORA toolbox [4].

### A. Random Zonotope Generation and Performance Measure

A random zonotope with $p$ $n$-dimensional generators is obtained by sampling $p$ independent generators. To obtain a generator $g$, its endpoint is sampled from points that are uniformly distributed on a unit hypersphere. To this end $\frac{x}{||x||_2}$ is computed, where $x \in \mathbb{R}^n$ is a random variable with normal distribution. The direction $v$ of the generator $g$ is obtained by the vector pointing from the origin of the hypersphere to the sampled endpoint. The length $l$ of the generator is sampled from a uniform, Gaussian or Gamma distribution in the interval $[0, 100]$. The generator is given by $g = lv$.

To measure tightness of the over-approximation, we use the volume ratio of the original zonotope $Z$ and its over-approximation $Z_{\text{red}}$: $R = \left(\frac{V(Z_{\text{red}})}{V(Z)}\right)^{\frac{1}{n}}$ as proposed in [2]. The smaller this ratio, the tighter is the over-approximation. For medium and high dimensions or orders of $Z$, $V(Z)$ is not computable anymore, and we use $R_G = \left(\frac{V(Z_{\text{red}})}{V(Z_G)}\right)^{\frac{1}{n}}$ as a tightness measure, where $Z_G$ (of order 1) is obtained by over-approximating $Z$ using the box method. Thus, $R_G$ can be smaller than 1.

### B. Performance of Order Reduction Methods

All experiments are run in MATLAB (version R2016b, with a memory limit of 10 GB) on a laptop with an Intel i7 CPU with 2.5 GHz and 16 GB of RAM running Ubuntu 16.04.

For each combination of dimension and order, we evaluate the performance of all techniques described above on 100 randomly sampled zonotopes, as long as the technique finishes in reasonable time and does not require more than 10 GB of memory. Sampling zonotopes from a uniform, Gaussian or Gamma distribution has a small influence on the performance of the methods. The box method performs best on zonotopes sampled from a uniform distribution. The other methods result in slightly tighter over-approximations if the zonotope was sampled from a Gaussian or Gamma distribution than from a uniform distribution. From now on, only the results for zonotopes sampled from a uniform distribution are shown. The most important results in low- ($n < 10$), medium- ($10 \leq n \leq 20$) and high-dimensional spaces ($n \geq 20$) are summarized in Table I and Figure IV-B.

*1) Box Method:* The box method is fast, robust and on average results in over-approximations of 1 up to 2 times the original volume in low dimensions.

*2) Transformation Method:* The exhaustive search transformation methods ($\text{ExSe}_y$ and $\text{NSe}_{y,z}$) have been run on all generators ($y = \text{all}$, $\text{ExSe}_{\text{All}}$), on the $y = n + 8$ longest generators and $z = n + 3$ combinations ($\text{ExSe}_8$, $\text{NSe}_{8,3}$ where $n$ is the number of dimensions). $\text{ExSe}_{\text{All}}$ leads to tight over-approximations, but the time needed to determine the transformation matrix is in $\mathcal{O}((n \cdot \text{ord}(Z))^n)$ (see Figure IV-B, first line, second column). Thus, this method can only be used to over-approximate zonotopes of small orders in low dimensions. In the low dimensional space, $\text{ExSe}_8$ and $\text{NSe}_{8,3}$ lead to similarly tight over-approximations as $\text{ExSe}_{\text{All}}$ does, but need less time to determine the transformation matrix. On

TABLE I

PERFORMANCE OF ORDER REDUCTION METHODS. THE MEAN, MEDIAN AND MAX VALUE REFER TO $R = \left(\frac{V(Z_{\text{red}})}{V(Z)}\right)^{\frac{1}{n}}$ OR $R_G = \left(\frac{V(Z_{\text{red}})}{V(Z_G)}\right)^{\frac{1}{n}}$ AND RUNTIME IS GIVEN IN SECONDS AND IS SET TO $\epsilon$ IF $< 0.0005$. BEST VALUES ARE SHOWN IN BLUE.

| $R$ | Distr.: uniform, Dim.: 3, Order: 2 | | | | | Distr.: uniform, Dim.: 3, Order: 4 | | | | | Distr.: uniform, Dim.: 3, Order: 6 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.647 | 1.576 | 2.507 | 0.248 | $\epsilon$ | 1.387 | 1.376 | 1.827 | 0.088 | $\epsilon$ | 1.329 | 1.331 | 1.509 | 0.053 | $\epsilon$ |
| ExSeAll | **1.099** | **1.092** | 1.208 | 0.044 | 0.007 | **1.158** | 1.162 | 1.241 | 0.027 | 0.062 | **1.178** | 1.181 | 1.222 | 0.022 | 0.215 |
| ExSe8 | **1.099** | **1.092** | 1.208 | 0.044 | 0.006 | 1.159 | 1.162 | 1.241 | 0.028 | 0.046 | 1.181 | 1.182 | 1.231 | 0.023 | 0.044 |
| NexSe8 | **1.099** | **1.092** | 1.208 | 0.044 | 0.002 | 1.166 | 1.166 | 1.251 | 0.033 | 0.003 | 1.192 | 1.194 | 1.291 | 0.03 | 0.003 |
| PCA | 1.374 | 1.375 | 1.552 | 0.079 | $\epsilon$ | 1.302 | 1.301 | 1.427 | 0.048 | $\epsilon$ | 1.283 | 1.282 | 1.362 | 0.033 | $\epsilon$ |
| LineCl | 1.302 | 1.209 | 4.412 | 0.371 | 0.081 | 1.276 | 1.228 | 2.226 | 0.169 | 0.084 | 1.315 | 1.238 | 3.177 | 0.299 | 0.087 |
| HybridPC | 1.206 | 1.193 | 1.46 | 0.107 | 0.081 | 1.234 | 1.229 | 1.427 | 0.059 | 0.084 | 1.24 | 1.239 | 1.362 | 0.044 | 0.087 |
| CoOptdir | 1.122 | 1.099 | 2.189 | 0.127 | 0.504 | 1.289 | **1.153** | 5.035 | 0.629 | 0.652 | 1.203 | **1.177** | 2.79 | 0.178 | 0.526 |
| CoOptSVD | 1.249 | 1.218 | 1.515 | 0.105 | 1.252 | 1.234 | 1.221 | 1.392 | 0.064 | 1.339 | 1.228 | 1.219 | 1.338 | 0.048 | 1.112 |
| $R$ | Distr.: uniform, Dim.: 6, Order: 2 | | | | | Distr.: uniform, Dim.: 6, Order: 4 | | | | | Distr.: uniform, Dim.: 6, Order: 6 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 2.108 | 2.09 | 2.669 | 0.153 | $\epsilon$ | 1.747 | 1.744 | 1.876 | 0.057 | $\epsilon$ | 1.666 | 1.665 | 1.819 | 0.042 | $\epsilon$ |
| ExSeAll | **1.258** | **1.263** | 1.371 | 0.061 | 0.267 | 1.401 | 1.403 | 1.467 | 0.032 | 38.418 | 1.446 | 1.449 | 1.488 | 0.022 | 555.627 |
| ExSe8 | **1.258** | **1.263** | 1.371 | 0.061 | 0.264 | 1.411 | 1.415 | 1.502 | 0.036 | 0.865 | 1.487 | 1.484 | 1.638 | 0.044 | 0.866 |
| NexSe8 | 1.26 | 1.265 | 1.407 | 0.062 | 0.007 | 1.418 | 1.419 | 1.548 | 0.042 | 0.017 | 1.495 | 1.493 | 1.624 | 0.045 | 0.017 |
| PCA | 1.708 | 1.712 | 1.851 | 0.063 | $\epsilon$ | 1.619 | 1.618 | 1.706 | 0.043 | 0.001 | 1.586 | 1.585 | 1.65 | 0.026 | 0.001 |
| LineCl | 1.903 | 1.746 | 4.341 | 0.58 | 0.144 | 2.035 | 1.898 | 4.311 | 0.562 | 0.163 | 2.472 | 1.932 | 21.792 | 2.218 | 0.179 |
| HybridPC | 1.615 | 1.66 | 1.83 | 0.137 | 0.144 | 1.606 | 1.612 | 1.706 | 0.052 | 0.164 | 1.579 | 1.579 | 1.65 | 0.035 | 0.18 |
| CoOptdir | 1.333 | 1.32 | 1.803 | 0.096 | 3.976 | **1.354** | **1.352** | 1.46 | 0.035 | 3.641 | **1.389** | **1.387** | 1.446 | 0.025 | 3.407 |
| CoOptSVD | 1.48 | 1.449 | 1.798 | 0.153 | 7.421 | 1.419 | 1.396 | 1.678 | 0.083 | 8.369 | 1.424 | 1.405 | 1.65 | 0.068 | 8.476 |
| $R_G$ | Distr.: uniform, Dim.: 10, Order: 5 | | | | | Distr.: uniform, Dim.: 10, Order: 10 | | | | | Distr.: uniform, Dim.: 10, Order: 15 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ |
| ExSe8 | 0.894 | 0.896 | 0.976 | 0.034 | 14.733 | 1.052 | 1.045 | 1.335 | 0.053 | 16.015 | 1.107 | 1.098 | 1.278 | 0.047 | 14.697 |
| NexSe8 | 0.897 | 0.898 | 1.027 | 0.037 | 0.243 | 1.059 | 1.055 | 1.232 | 0.049 | 0.263 | 1.119 | 1.104 | 1.494 | 0.072 | 0.243 |
| PCA | 0.933 | 0.933 | 0.98 | 0.019 | 0.001 | 0.969 | 0.968 | 1.003 | 0.014 | 0.001 | 0.978 | 0.978 | 1.003 | 0.012 | 0.001 |
| LineCl | 1.778 | 1.56 | 6.724 | 0.924 | 0.306 | 2.035 | 1.589 | 22.59 | 2.262 | 0.417 | 1.837 | 1.559 | 14.563 | 1.46 | 0.443 |
| HybridPC | 0.933 | 0.933 | 0.98 | 0.019 | 0.304 | 0.969 | 0.968 | 1.003 | 0.014 | 0.416 | 0.978 | 0.978 | 1.003 | 0.012 | 0.443 |
| CoOptdir | 0.847 | 0.847 | 0.899 | 0.019 | 10.811 | **0.899** | 0.898 | 0.938 | 0.015 | 15.455 | 0.922 | 0.922 | 0.951 | 0.011 | 16.576 |
| CoOptSVD | **0.798** | **0.792** | 0.967 | 0.032 | 28.985 | 0.907 | **0.881** | 0.996 | 0.05 | 25.524 | **0.907** | **0.903** | 0.985 | 0.021 | 26.692 |
| $R_G$ | Distr.: uniform, Dim.: 15, Order: 5 | | | | | Distr.: uniform, Dim.: 15, Order: 10 | | | | | Distr.: uniform, Dim.: 15, Order: 15 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ |
| ExSe8 | 0.956 | 0.951 | 1.094 | 0.044 | 295.49 | 1.148 | 1.141 | 1.319 | 0.055 | 355.923 | 1.232 | 1.215 | 1.452 | 0.062 | 288.544 |
| NexSe8 | 0.97 | 0.957 | 1.163 | 0.057 | 3.843 | 1.154 | 1.147 | 1.311 | 0.055 | 4.056 | 1.246 | 1.237 | 1.509 | 0.073 | 3.72 |
| PCA | 0.93 | 0.93 | 0.962 | 0.015 | 0.001 | 0.968 | 0.967 | 1.006 | 0.01 | 0.001 | 0.978 | 0.978 | 0.998 | 0.008 | 0.001 |
| LineCl | 2.409 | 1.882 | 25.774 | 2.745 | 0.459 | 2.235 | 1.978 | 7.736 | 0.952 | 0.93 | 2.955 | 1.998 | 43.52 | 4.519 | 0.895 |
| HybridPC | 0.93 | 0.93 | 0.962 | 0.015 | 0.464 | 0.968 | 0.967 | 1.006 | 0.01 | 0.924 | 0.978 | 0.978 | 0.998 | 0.008 | 0.946 |
| CoOptdir | 0.848 | 0.848 | 0.884 | 0.013 | 24.459 | **0.896** | **0.894** | 0.921 | 0.011 | 35.235 | **0.915** | **0.915** | 0.94 | 0.009 | 31.141 |
| CoOptSVD | **0.823** | **0.819** | 0.924 | 0.03 | 58.112 | 0.953 | 0.958 | 1.094 | 0.03 | 78.255 | 0.977 | 0.974 | 1.127 | 0.027 | 67.467 |
| $R_G$ | Distr.: uniform, Dim.: 15, Order: 50 | | | | | Distr.: uniform, Dim.: 15, Order: 100 | | | | | Distr.: uniform, Dim.: 15, Order: 300 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** |
| PCA | 0.994 | 0.993 | 1.003 | 0.004 | 0.004 | 0.997 | 0.997 | 1.004 | 0.003 | 0.007 | 0.999 | 0.999 | 1.002 | 0.002 | 0.041 |
| LineCl | 2.257 | 1.715 | 12.39 | 1.76 | 2.032 | 1.715 | 1.475 | 5.496 | 0.715 | 2.687 | 1.513 | 1.115 | 12.791 | 1.393 | 6.253 |
| HybridPC | 0.994 | 0.993 | 1.003 | 0.004 | 2.043 | 0.997 | 0.997 | 1.004 | 0.003 | 2.773 | 0.999 | 0.999 | 1.002 | 0.002 | 6.515 |
| CoOptdir | **0.962** | **0.962** | 0.981 | 0.006 | 49.344 | **0.976** | **0.976** | 0.993 | 0.005 | 49.154 | **0.99** | **0.99** | 0.997 | 0.003 | 63.939 |
| $R_G$ | Distr.: uniform, Dim.: 45, Order: 10 | | | | | Distr.: uniform, Dim.: 45, Order: 15 | | | | | Distr.: uniform, Dim.: 45, Order: 30 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** |
| PCA | **0.925** | **0.926** | 0.935 | 0.005 | 0.004 | **0.963** | **0.963** | 0.973 | 0.004 | 0.004 | **0.976** | **0.976** | 0.982 | 0.002 | 0.006 |
| $R_G$ | Distr.: uniform, Dim.: 60, Order: 10 | | | | | Distr.: uniform, Dim.: 60, Order: 15 | | | | | Distr.: uniform, Dim.: 60, Order: 30 | | | | |
| Method | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time | Mean | Median | Max | Std. | Time |
| Box | 1.0 | 1.0 | 1.0 | 0.0 | $\epsilon$ | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** | 1.0 | 1.0 | 1.0 | 0.0 | **0.001** |
| PCA | **0.924** | **0.924** | 0.936 | 0.004 | 0.004 | **0.963** | **0.963** | 0.972 | 0.003 | 0.005 | **0.976** | **0.976** | 0.98 | 0.002 | 0.006 |

zonotopes of medium order in medium dimensional spaces, the over-approximations computed by $\text{ExSe}_8$ and $\text{NSe}_{8,3}$ are not tight anymore: Both methods perform worse than the box method.

*3) PCA-based Method:* PCA (principle component analysis) is very fast and robust in any dimension and on zonotopes of arbitrary order. It leads to over-approximations that are constantly tighter than the box method (see Table I). In high dimensions, this method results in the tightest over-approximations of all techniques with feasible runtime.

*4) LineCl:* LineCl (line clustering, with 10 runs and $\xi = 10^{-7}$) often results in tight over-approximations but in some cases performs worse than the box method. The runtime of LineCl is quite short. This method can be used on zonotopes of any dimension or order.

*5) HybridPC:* HybridPC (hybrid of line clustering and PCA, with 10 runs and $\xi = 10^{-7}$) leads to good over-approximations, within a reasonable runtime and works on zonotopes of arbitrary orders in any dimension.

*6) Constrained Optimization Methods:* The optimization methods CoOptdir (direct volume optimization) and

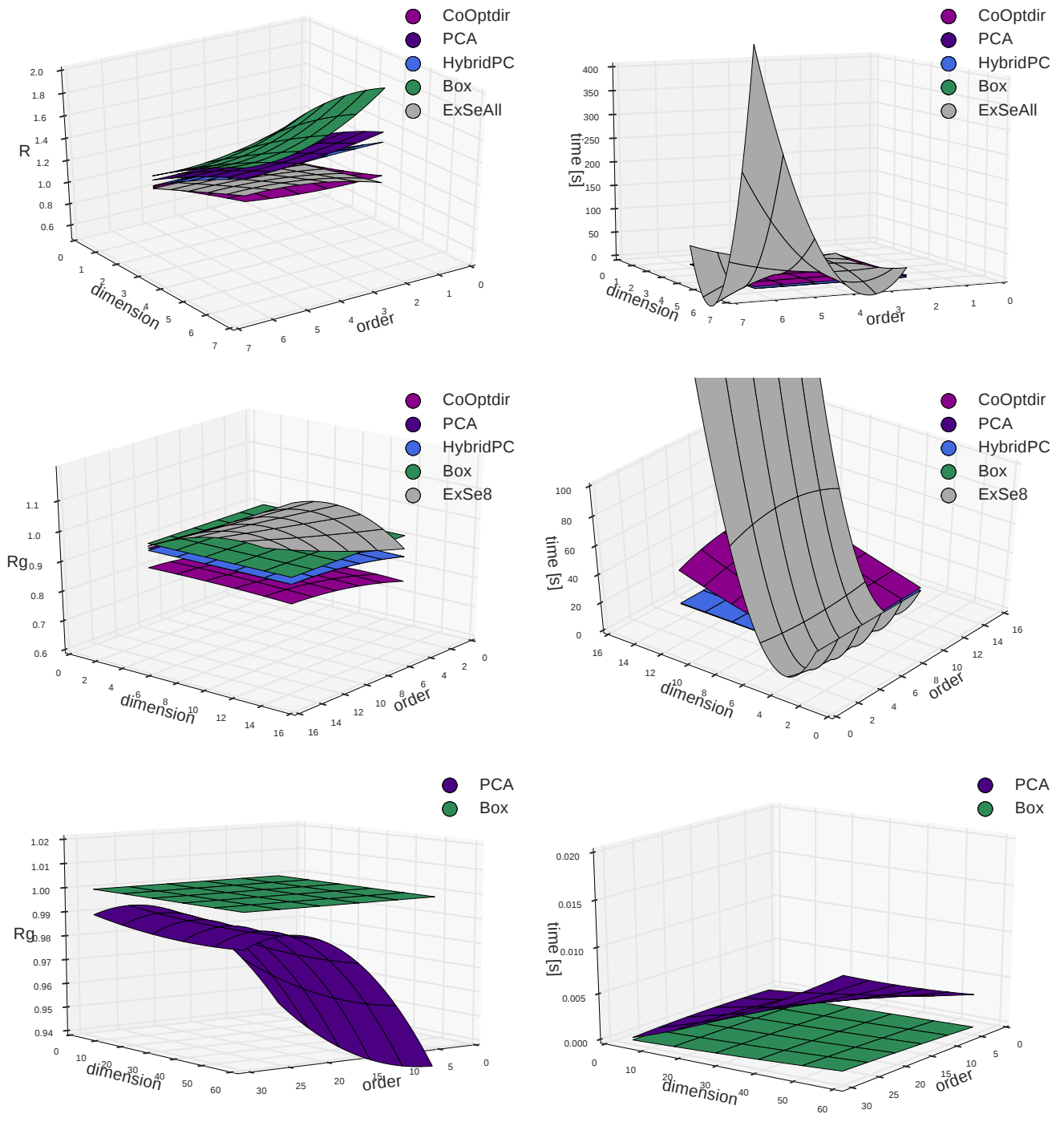


Fig. 5. Performance of the best order reduction methods in low ($n < 10$), medium ($10 \leq n \leq 20$) and high dimensions ($n \geq 20$). Left column: mean volume ratio $R = \left(\frac{V(Z_{\text{red}})}{V(Z)}\right)^{\frac{1}{n}}$ or $R_G = \left(\frac{V(Z_{\text{red}})}{V(Z_G)}\right)^{\frac{1}{n}}$ in low, medium and high-dimensional spaces, right column: mean runtime [s] in low-, medium- and high-dimensional spaces

CoOptSVD (volume optimization based on SVD) use the MATLAB nonlinear programming solver fmincon and the interior-point algorithm with at most $5,000$ iterations and $100,000$ function evaluations. Both techniques lead to very tight over-approximations and have a reasonable runtime, as illustrated in Figure IV-B. CoOptdir sometimes has problems inverting a matrix that is close to singularity but is faster than CoOptSVD. SVD-based constrained optimization (CoOptSVD) can reduce zonotopes of up to medium orders and dimensions. The direct optimization method (CoOptdir) even works on zonotopes of high order and shows the best performance of all methods on zonotopes of low up to

medium order and dimension (see Table I).

*7) The Optimal Method:* The optimal method for over-approximating a zonotope $Z$ by $Z_{\mathrm{red}}$ of smaller order strongly depends on the number of dimensions and on the initial order of $Z$ (see blue values in Table I). In the 2-dimensional space, or on zonotopes with order 2, the exhaustive search methods $\mathrm{ExSe}_{\mathrm{All}}$ and $\mathrm{ExSe}_8$ yield the tightest over-approximations. If the original zonotope has dimension greater than 4 and order greater than 2, the tightest over-approximations are obtained by using the proposed optimization techniques CoOptdir or CoOptSVD. CoOptdir is faster and in most cases slightly better than CoOptSVD. In high-dimensional space ($n \geq 20$), the only methods with feasible runtime are our PCA-based method and the box method. In this case, the PCA-based technique leads to slightly better over-approximations than the box method.

## V. CONCLUSIONS AND FUTURE WORK

This paper has two main contributions: We propose several new methods based on PCA, clustering (LineCl, HybridPC) and constrained optimization (CoOptdir, CoOptSvd) for the tight over-approximation of zonotopes. Furthermore, we provide an overview of the most important known methods for order reduction and compare their performance with our methods in low and high-dimensional spaces. The optimal technique to over-approximate a zonotope $Z$ depends on the dimension and initial order of $Z$. In the low dimensional space, the optimization methods (CoOptdir, CoOptSvd) we propose lead to very tight over-approximations. If the dimension of $Z$ is high, the best choice for obtaining a tight over-approximation in a feasible runtime is our PCA-based order reduction method.

All order reduction techniques reduce the order of a zonotope by over-approximating it. It remains to be answered, if the presented method can be modified such that the reduced zonotope is an under-approximation of the original zonotope in addition.

## VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support by the European Commission project UnCoVerCPS under grant number 643921.

## REFERENCES

[1] T. Alamo, J. M. Bravo, and E. F. Camacho. Guaranteed state estimation by zonotopes. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 5831–5836, 2003.

[2] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. Thesis, Technische Universität München, 2010.

[3] M. Althoff. Formal and compositional analysis of power systems using reachable sets. *IEEE Transactions on Power Systems*, 29(5):2270–2280, 2014.

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

[5] M. Althoff and J. M. Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[6] G. D. Bailey. Coherence and enumeration of tilings of 3-zonotopes. *Discrete & Computational Geometry*, 22:119–147, 1999.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] A. Blum, J. Hopcroft, and R. Kannan. *Foundations of Data Science*. To appear.

[9] J. Bourgain and J. Lindenstrauss. Approximating the ball by a Minkowski sum of segments with equal length. *Discrete & Computational Geometry*, 9:131–144, 1993.

[10] 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.

[11] S. Campi, D. Haas, and W. Weil. Approximation of zonoids by zonotopes in fixed directions. *Discrete & Computational Geometry*, 11:419–431, 1994.

[12] C. Combastel. A state bounding observer based on zonotopes. In *Proc. of the European Control Conference*, pages 2589–2594, 2003.

[13] C. Combastel. A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes. In *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 7228–7234, 2005.

[14] S. Felsner and G. M. Ziegler. Zonotopes associated with higher Bruhat orders. *Discrete Mathematics*, 241:301–312, 2001.

[15] J.-A. Ferrez, K. Fukuda, and T. M. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operational Research*, 166:35–50, 2005.

[16] K. Fukuda. From the zonotope construction to the Minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004.

[17] K. Ghorbal, E. Goubault, and S. Putot. The zonotope abstract domain Taylor1+. In *Proc. of the 21st International Conference on Computer Aided Verification*, pages 627–633, 2009.

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

[19] E. Goubault and S. Putot. Static analysis of numerical algorithms. In *Proc. of the 13th International Static Analysis Symposium*, pages 18–34, 2006.

[20] E. Gover and N. Krikorian. Determinants and the volumes of parallelotopes and zonotopes. *Linear Algebra and its Applications*, 433(1):28–40, 2010.

[21] L. J. Guibas, A. Nguyen, and L. Zhang. Zonotopes as bounding volumes. In *Proc. of the Symposium on Discrete Algorithms*, pages 803–812, 2005.

[22] F. Immler. A verified algorithm for geometric zonotope/hyperplane intersection. In *Proc. of the Conference on Certified Programs and Proofs*, pages 129–136, 2015.

[23] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

[24] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. *Zonotopes: From Guaranteed State-estimation to Control*. Wiley, 2013.

[25] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. Zonotopic guaranteed state estimation for uncertain systems. *Automatica*, 49(11):3418–3424, 2013.

[26] A. Pereira and M. Althoff. Safety control of robots under computed torque control using reachable sets. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 331–338, 2015.

[27] T. Rörig, N. Witte, and G. M. Ziegler. Zonotopes with large 2D-cuts. *Discrete & Computational Geometry*, 42:527–541, 2009.

[28] B. Schürmann and M. Althoff. Convex interpolation control with formal guarantees for disturbed and constrained nonlinear systems. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 121–130, 2017.