A Sum-of-Squares-Based Procedure to Approximate the Pontryagin Difference of Basic Semi-algebraic Sets

Andres Cotorruelo [a,*], Ilya Kolmanovsky [b], Emanuele Garone [a]

[a] *Service d'Automatique et d'Analyse des Systèmes (SAAS), Université Libre de Bruxelles (ULB), Brussels, Belgium*

[b] *Department of Aerospace Engineering, University of Michigan, MI, U.S.A.*

Abstract

The P-difference between two sets $\mathcal{A}$ and $\mathcal{B}$ is the set of all points, $\mathcal{C}$, such that the sum of $\mathcal{B}$ to any of the points in $\mathcal{C}$ is contained in $\mathcal{A}$. Such a set difference plays an important role in robust model predictive control and set-theoretic control. In this paper, we show that an inner approximation of the P-difference between two sets described by collections of polynomial inequalities can be computed using Sums of Squares Programming. The effectiveness of the procedure is shown with some computational examples.

Key words: Pontryagin difference, Sum of Squares, Robust control

1 Introduction

The Pontryagin set difference (or simply P-difference), so named after L.S. Pontryagin who used it in the setting of game theory [1], is a fundamental tool in robust model predictive control (MPC) [2,3] and in set theoretic control [4,5]. Additionally, the P-difference has found important applications in image processing [6] and in path planning [7]. In the literature, the P-difference is also known as *Minkowski set difference* or *set erosion* [8].

It is well known that it is possible to compute the P-difference between polyhedral sets by solving a Linear Programming (LP) problem, as reported in [5]. This approach is broadly used in control and is implemented in many constrained control toolboxes, *e.g.* [9]. In [10], an algorithm is proposed to approximate the P-difference of zonotopes by solving systems of linear equations. In [11], the authors propose a novel method to compute the P-difference of convex polyhedra based on the fact that $\mathcal{A} \ominus \mathcal{B}$ is equivalent to the Minkowski sum of the complement of $\mathcal{A}$ and the symmetric reflection of $\mathcal{B}$. In [12], the authors show that the P-difference between some specific classes of convex sets can be computed using their *support functions*. However these support functions are notoriously hard to compute and closed forms are known only for some specific sets. In the same paper the authors also provide an algorithm to approximate the P-difference in the case of general convex sets. However, this algorithm is based on sampling and thus convenient only for low dimensions. In computer graphics, the erosion of general sets (also known as binary erosion) is typically approached numerically using "brute force" approaches. These approaches are viable only for low-dimensional sets, e.g. for 2D or 3D images [13].

In this paper, we propose a method to obtain inner approximations of the P-difference between two sets described by finitely many polynomial nonstrict inequalities (basic semi-algebraic sets) using Sum of Squares Programming (SOSP). Computational examples are reported to illustrate the effectiveness of the proposed approach.

Notation: The set of all polynomials in the variables $x_1, \ldots, x_N$ and with coefficients in $\mathbb{R}$ is denoted by $\mathbb{R}[x_1, \ldots, x_N]$. A polynomial that can be expressed as the sum of polynomials raised to an even power is referred to as a *Sum of Squares (SOS) polynomial*. The

* Corresponding author.
Email addresses: acotorru@ulb.ac.be (Andres Cotorruelo), ilya@umich.edu (Ilya Kolmanovsky), egarone@ulb.ac.be (Emanuele Garone).

[1] Special thanks to Philip Pugeau for providing some of the case studies used in this paper. This research has been funded partly by Ministerio de Economía y Competitividad of Spain under project DPI2016-76493-C3-1-R co-financed by European FEDER Funds. The second author acknowledges the support of the National Science Foundation grant 1931738.

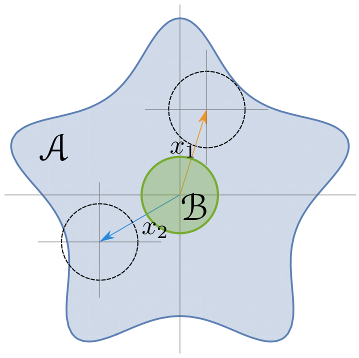

Fig. 1. Pontryagin difference between sets $\mathcal{A}$, the solid blue star, and $\mathcal{B}$, the solid green ball. As it can be seen, x_1 will belong to $\mathcal{A} \ominus \mathcal{B}$ since every vector $z \in \mathcal{B}$ is such that $x_1 + z \in \mathcal{A}$. This can be verified by noting that the set $\mathcal{B}$ translated by x_1 (depicted by an orange arrow and a dashed circle) is completely within $\mathcal{A}$. On the contrary, x_2 does not belong to $\mathcal{A} \ominus \mathcal{B}$ since there exist some $z' \in \mathcal{B}$ such that $x_2 + z' \notin \mathcal{A}$ which means that the set $\mathcal{B}$ translated by x_2 (depicted by a blue arrow and a dashed ball) is not fully contained in $\mathcal{A}$.

set of all SOS polynomials in the variables $x_1, \dots, x_N$ is denoted by $\Sigma[x_1, \dots, x_N]$. For $x \in \mathbb{R}^n$, $\mathbb{R}[x]$ (resp. $\Sigma[x]$) denotes $\mathbb{R}[x_1, \dots, x_n]$ (resp. $\Sigma[x_1, \dots, x_n]$). Given a polynomial p, its overall degree is denoted by $\deg(p)$, its degree in the variable y is denoted as $\deg_y(p)$, and its set of coefficients is denoted as $\mathrm{cf}(p)$. For a number of elements $a_1, \dots, a_n$, $\{a_i\}_{i=1}^n$ denotes the set $\{a_1, \dots, a_n\}$, and every operator applied to it is meant to be understood element-wise, *e.g.* $\{a_i\}_{i=1}^n \geq 0$ means $a_i \geq 0$, $\forall i = 1, \dots, n$. For a finite set of polynomials $P = \{p_i\}_{i=1}^n \in \mathbb{R}[x]$, the cone and the multiplicative monoid generated by P are denoted by $\mathbf{K}(P)$ and $\mathbf{M}(P)$, respectively; these algebraic structures are defined in Appendix A. The sets of all positive and nonnegative integers are denoted by $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$, respectively.

2 Problem statement

For two sets $\mathcal{A} \subset \mathbb{R}^n$ and $\mathcal{B} \subset \mathbb{R}^n$, the P-difference is defined as

$$\mathcal{A} \ominus \mathcal{B} = \{x \in \mathcal{A} : x + z \in \mathcal{A} \ \forall z \in \mathcal{B}\},$$

where typically $0 \in \mathcal{B}$. A simple geometrical way to interpret this operation is that the set $\mathcal{C} = \mathcal{A} \ominus \mathcal{B}$ is a set such that if we select a point in $\mathcal{C}$ and we add an any element of $\mathcal{B}$, the resulting point still belongs to $\mathcal{A}$ (see Fig. 1). The objective of this paper is to solve the following problem:

Problem 2.1 *(Pontryagin difference) Let $\mathcal{A}$ and $\mathcal{B}$ be two basic semi-algebraic sets in the form*

$$\mathcal{A} = \{x \in \mathbb{R}^n : a_i(x) \geq 0, \, i = 1, \dots, m_{\mathcal{A}}\},$$
$$\mathcal{B} = \{x \in \mathbb{R}^n : b_j(x) \geq 0, \, j = 1, \dots, m_{\mathcal{B}}\},$$

where $\{a_i(x)\}_{i=1}^{m_{\mathcal{A}}}, \{b_j(x)\}_{j=1}^{m_{\mathcal{B}}} \in \mathbb{R}[x]$. Determine an inner approximation of $\mathcal{A} \ominus \mathcal{B}$.

3 Computation of the Pontryagin Difference

In this section we propose a way to solve Problem 2.1 based on the Krivine – Stengle Positivstellensatz (P-satz) [14]. To simplify the problem, the first step is to note that the set $\mathcal{A}$ can be represented as $\mathcal{A} = \bigcap_{i=1}^{m_{\mathcal{A}}} \mathcal{A}_i$ where $\mathcal{A}_i = \{x : a_i(x) \geq 0\}$, $i = 1, \dots, m_{\mathcal{A}}$.
Since $\mathcal{A} \ominus \mathcal{B} = \bigcap_{i=1}^{m_{\mathcal{A}}} (\mathcal{A}_i \ominus \mathcal{B})$, we can focus on a single set $\mathcal{A}_i$ at a time without any loss of generality. Considering the P-difference $\mathcal{A}_i \ominus \mathcal{B} = \{x : a_i(x + z) \geq 0 \ \forall z \in \mathcal{B}\}$, a possible way to approximate $\mathcal{A}_i \ominus \mathcal{B}$ from inside is by means of a set $\mathcal{C}_i = \{x : c_i(x) \geq 0\} \subseteq \mathcal{A}_i \ominus \mathcal{B}$, $c_i(x) \in \mathbb{R}[x]$, where the polynomial c_i must be such that

$$c_i(x) \leq \min_{z \in \mathcal{B}} a_i(x + z) \ \forall x \in \mathbb{R}^n. \tag{1}$$

Note that whenever (1) is an equality, $\mathcal{C}_i = \mathcal{A}_i \ominus \mathcal{B}$.

Remark 1 *Note that this choice of $c(x)$ limits the approximation space to basic semi-algebraic sets defined by only one inequality. To the best of the authors' knowledge, it has not been proven that the P-difference of basic semi-algebraic sets is itself a basic semi-algebraic set comprised of the same number of inequalities. Hence, such a choice can be a source of conservatism.*

Condition (1) is equivalent to the following set emptiness condition,

$$\{(x, z) : c_i(x) - a_i(x + z) > 0, \, z \in \mathcal{B}\} = \emptyset. \tag{2}$$

Since in the Krivine–Stengle P-satz, the set required to be empty is described in terms of *equal-to*, *greater-than-or-equal-to*, and *not-equal-to* operators, the set in the left-hand side of (2) is thus rewritten in terms of these operators as

$$\Big\{(x, z) : c_i(x) - a_i(x + z) \geq 0, \\ c_i(x) - a_i(x + z) \neq 0, \, \{b_j(z)\}_{j=1}^{m_{\mathcal{B}}} \geq 0\Big\} = \emptyset. \tag{3}$$

At this point, the Krivine–Stengle P-satz states [14] that (3) is satisfied if and only if there exist two polynomials $p(x, z)$ and $q(x, z)$ such that $p(x, z) + q^2(x, z) = 0$, where $p \in \mathbf{K}\left(\{c_i(x) - a_i(x + z), b_1(z), \dots, b_{m_{\mathcal{B}}}(z)\}\right)$, and $q \in \mathbf{M}\left(c_i(x) - a_i(x + z)\right)$. By conveniently selecting only some of the terms of $\mathbf{K}$ and $\mathbf{M}$ in the above necessary and sufficient condition, we obtain the following sufficient condition,

$$\left(c_i(x) - a_i(x + z)\right)^2 + s_0(x, z)\left(c_i(x) - a_i(x + z)\right) \\ + \sum_{j=1}^{m_{\mathcal{B}}} s_j(x, z) b_j(z)\left(c_i(x) - a_i(x + z)\right) = 0,$$

where $\{s_j(x,z)\}_{j=0}^{m_\mathcal{B}} \in \Sigma[x,z]$. This equation can be further simplified by dividing by $c_i(x) - a_i(x+z)$, which is always a nonzero polynomial, to obtain:

$$c_i(x) - a_i(x+z) + s_0(x,z) + \sum_{j=1}^{m_\mathcal{B}} s_j(x,z) b_j(z) = 0.$$

Since $s_0 \in \Sigma[x,z]$, it follows that the latter is equivalent to

$$P_i(x,z) \overset{\text{def}}{=} a_i(x+z) - c_i(x) - \sum_{j=1}^{m_\mathcal{B}} s_j(x,z) b_j(z) \in \Sigma[x,z]. \quad (4)$$

Finally, since we are interested in the largest inner approximation of $\mathcal{A} \ominus \mathcal{B}$, using (4) we can define the problem of finding $c_i(x)$ as the following Sum of Squares Programming (SOSP) problem

$$\begin{aligned} \max_{\mathrm{cf}(\{s_j\}_{j=1}^{m_\mathcal{B}}),\, \mathrm{cf}(c_i)} \quad & \int_\mathcal{R} c_i(x) \ \mathrm{d}x \\ \text{s.t.} \quad & P_i(x,z) \in \Sigma[x,z] \\ & \{s_j(x,z)\}_{j=1}^{m_\mathcal{B}} \in \Sigma[x,z], \end{aligned} \quad (5)$$

where $\mathcal{R} \supseteq \mathcal{A}$ is a domain such that the expression of $\int_\mathcal{R} c_i(x)\ \mathrm{d}x$ is polynomial. A possible such choice is an outer-bounding box for $\mathcal{A}$, which can be computed using the algorithm presented in [15].

As well known [16], once the structure and the degrees of the decision polynomials $\{s_j\}_{j=1}^{m_\mathcal{B}}$, c_i have been chosen, optimization problem (5) can in turn be cast into a Semi-Definite Programming (SDP) optimization problem that can be solved efficiently using existing SDP solvers *e.g.* [17]. In order for (5) to admit a solution, the following necessary conditions on the degrees of the decision polynomials $c_i(x)$ and $\{s_j\}_{j=1}^{m_\mathcal{B}}$ must hold

$$\begin{aligned} \max_j \{\deg_z(s_j) + \deg_z(b_j)\} &\geq \deg_z(a_i) \\ \deg(c_i) &\geq \max_j \{\deg_x(s_j)\}. \end{aligned} \quad (6)$$

It is worth mentioning that the if degree of (4) is odd, any feasible solution of (5) will be such that the odd terms of the highest degree cancel out, rendering $P_i(x,z)$ of even degree.

Remark 2 *The upper bound on the decision variables of (5) is given by* $\binom{n+d_c}{d_c} + \sum_{j=1}^{m_\mathcal{B}} \binom{n+d_{s_j}}{d_{s_j}}$*, where* $d_c = \deg(c_i)$ *and* $d_{s_j} = \deg(s_j)\ \forall j = 1, \ldots, m_\mathcal{B}$*. It is important to choose the degrees so that this upper bound does not become unreasonably large.*

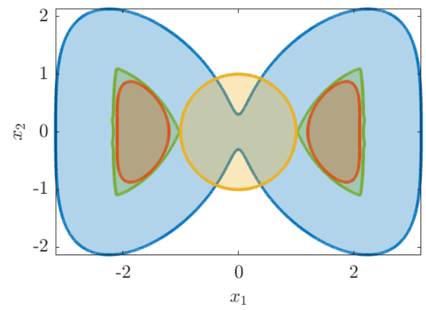

Fig. 2. Result of subtracting the norm-2 ball from the *bow-tie* set.

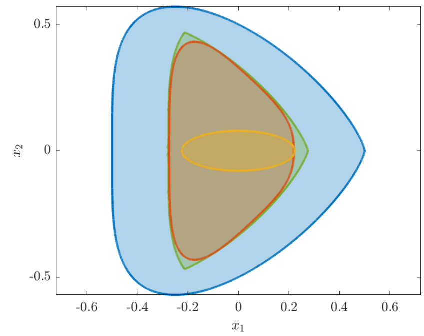

Fig. 3. Result of subtracting an ellipsoid from the *guitar pick* set.

4 Examples

To illustrate the efectiveness of the proposed methodology, in this section we apply it to a number of 2 and 3-dimensional sets. All of the showcased examples depict $\mathcal{C} \approx \mathcal{A} \ominus \mathcal{B}$, where $\mathcal{A} = \{x : a(x) \geq 0\}$, and $\mathcal{B} = \{x : b(x) \geq 0\}$ with varying $a(x)$ and $b(x)$ depending on the example. Table 1 reports the expressions of $a(x)$ and $b(x)$, the chosen degrees of $c(x)$ and the $s_i(x,z)$, the elapsed time to compute the approximation, as well as the following error index

$$e_{max} = \max_{x \in \partial\mathcal{C}} \inf_{y \in \mathcal{B},\, z \in \bar{\mathcal{A}}} ||x + y - z||,$$

where $\partial\mathcal{C}$ denotes the boundary of $\mathcal{C}$ and $\bar{\mathcal{A}}$ denotes the complement of $\mathcal{A}$. This index represents the maximum distance from $\mathcal{C}$ to $\mathcal{A} \ominus \mathcal{B}$. For fairness of comparison, in parentheses we also report e_{max} normalized with respect to the maximum radius of the set $\mathcal{B}$, *i.e.* $e_{max,n} = \frac{e_{max}}{\max_{x \in \mathcal{B}} ||x||}$.
In Figs. 2–6 $\mathcal{A}$ is depicted as a solid blue set, $\mathcal{B}$ as a solid yellow set, $\mathcal{C}$ as a solid orange set, and $\mathcal{A} \ominus \mathcal{B}$ (computed by gridding) as a green set. For space reasons, the expressions of $c(x)$ have been omitted from this paper, but they can be found in the addendum [18]. All SDP optimization problems were solved using Mosek [17] interfaced in Julia 1.5.2 running on an Intel Core i7-7500 at 2.7 GHz with 16 GB of RAM.

Fig.	$a(x)$	$b(x)$	deg(c)	deg(s_j)	t	e_{max} $(e_{max,n})$
2	$0.1 - x_1^4 - x_2^4 + 10x_1^2 - x_2^2$	$1 - x_1^2 - x_2^2$	14	6	2.14 s	0.2864 (0.29)
3	$x_2^4 - (x_1 - 0.5)^3 - (x_1 - 0.5)^4$	$0.1 - 2x_1^2 - 16x_2^2$	10	6	0.77 s	0.053 (0.23)
4	$4 - x_1^2 - x_2^2$	$0.1 - 2.5x_1^2x_2^2$ $-0.05(x_1 + x_2)^2$	10	2	0.15 s	0.0598 (0.09)
5	$-(x_1^2 + x_2^2 + x_3^2)^3 + 3(x_1^2 + x_2^2 + x_3^2)^2$ $-9(x_1^2 + x_2^2 + 3) + 16(x_1^3 - 3x_1x_2^2 + 2x_3^2)$	$0.1 - x_1^2 - x_2^2 - 4x_3^2$	10	4	0.75 s	0.2338 (0.74)
6	$1 - x_1^6 - x_2^6 - x_3^6 + 5x_1^4x_2x_3 - 3x_1^4x_2^2$ $-10x_1^2x_2^3x_3 - 3x_1^2x_2^4 + x_2^5x_3$	$10^{-4} - x_1^6 - x_2^6 - x_3^6$	10	4	8 min 40 s	0.1577 (0.732)

Table 1
Expressions of $a(x)$ and $b(x)$, degrees of $c(x)$ and $s(x, z)$, computational times and maximum and normalized errors for the examples depicted in Figs. 2–6.

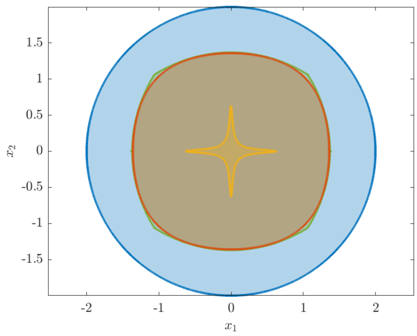

Fig. 4. Result of subtracting a 4-pointed star-shaped set from the norm-2 ball.

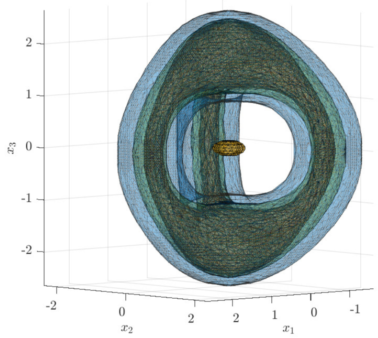

Fig. 5. Result of subtracting an ellipsoid from the 3-dimensional 2-torus.

5 Computational studies

In this section we present two numerical studies demonstrating the effectiveness of the proposed methodology: in the first one we test the accuracy of our method in the case of polyhedral sets, and in the second one we show how our methodology scales with the dimensions of the problem.

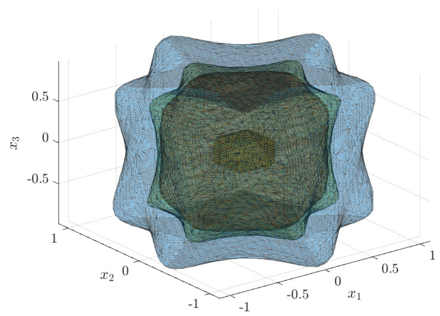

Fig. 6. Result of subtracting the 6-norm ball from the rotated 5-pointed star algebraic cylinder.

5.1 *Polyhedral P-difference*

It is well known that when $\mathcal{A}$ and $\mathcal{B}$ are both polyhedral, the problem of computing $\mathcal{A} \ominus \mathcal{B}$ can be solved exactly and very efficiently using the *ad hoc* algorithm presented in [5] and for which publicly available state-of-the-art implementations like MPT3 [9] or the one present in the library `LazySets.jl` [19] are available. In this subsection we will particularize (5) for the polyhedral case and compare the exact solution computed with the implementation of the P-difference in `LazySets.jl` with that of our methodology. In the case at hand, $a_i(x) = \alpha_i^{\mathrm{T}} x + \alpha_i^0$, with $\alpha_i \in \mathbb{R}^n$, $\alpha_i^0 \in \mathbb{R}$, $i = 1, \ldots, n$; $b_j(x) = \beta_j^{\mathrm{T}} x + \beta_j^0$, $\beta_j \in \mathbb{R}^n$, $\beta_j^0 \in \mathbb{R}$, $j = 1, \ldots, m_{\mathcal{B}}$, and $c_i(x) = \alpha_i^{\mathrm{T}} x + \theta_i$, $\theta_i \in \mathbb{R}$. Under these assumptions, equation (4) becomes

$$\bar{P}_i(z) \overset{\text{def}}{=} \alpha_i^{\mathrm{T}} z + \alpha_i^0 - \theta_i - \sum_{j=1}^{m} \bar{s}_j(z) b(z) \in \Sigma[z], \quad (7)$$

where $\{\bar{s}_j\}_{i=1}^{m_{\mathcal{B}}} \in \Sigma[z]$. In turn, the SOSP optimization problem becomes

$$\begin{aligned} \max_{\mathrm{cf}(\{\bar{s}_j\}_{j=1}^{m_{\mathcal{B}}}),\theta_i} \quad & \theta_i \\ \text{s.t.} \quad & \bar{P}_i(z) \in \Sigma[z] \\ & \{\bar{s}_j(z)\}_{j=1}^{m_{\mathcal{B}}} \in \Sigma[z]. \end{aligned} \quad (8)$$

Remark 3 *Note that, unlike the* $s_j(x, z)$ *in* (5)*, the* $\bar{s}_j(z)$ *in* (7) *do not depend on* x *since none of the other polynomials in* (7) *do, thus drastically reducing the number of decision variables in* (8)*.*

In this study we computed 1000 instances of the P-difference of a randomly oriented halfspace $\mathcal{A} = \{x \in \mathbb{R}^n : \alpha^{\mathrm{T}} x \geq 0\}, \alpha \in \mathbb{R}^n$ and a random irregular n-simplex [2] for increasing n. The results of this study can be found in Table 2. For all dimensions we set $\deg(\{\bar{s}_j\}_{j=1}^{m_{\mathcal{B}}}) = 2$ which ensures that the conditions (6) are satisfied. As can be seen, the SOSP method proposed

n	t	e_{max} ($e_{max,n}$)	t_{LS}
2	1.8 ms	$1.7 \cdot 10^{-10}$ ($3{\cdot}10^{-10}$)	0.6 µs
3	2.8 ms	$1.8{\cdot}10^{-10}$ ($2.2{\cdot}10^{-10}$)	0.9 µs
4	4.2 ms	$5.5{\cdot}10^{-10}$ ($7.3{\cdot}10^{-10}$)	0.95 µs
5	6.2 ms	$7.2{\cdot}10^{-10}$ ($9.6{\cdot}10^{-10}$)	0.98 µs
6	10 ms	$1.1{\cdot}10^{-9}$ ($1.4{\cdot}10^{-9}$)	1.1 µs
7	16 ms	$1.5{\cdot}10^{-9}$ ($2.1{\cdot}10^{-9}$)	1.2 µs
8	28 ms	$2.3{\cdot}10^{-8}$ ($2.9{\cdot}10^{-8}$)	1.2 µs
9	47 ms	$1.8 \cdot 10^{-8}$ ($2.4{\cdot}10^{-8}$)	1.2 µs
10	78 ms	$3.1 \cdot 10^{-8}$ ($4.1{\cdot}10^{-8}$)	1.2 µs

Table 2
Computational time and average maximum and normalized error of our method alongside the computational time of `LazySets.jl` for the study in Section 5.1 for $n \in \{2, 3, \ldots, 10\}$.

in this paper is able to get very close to the exact solution. The difference in computational times is due to the fact that the algorithm that `LazySets.jl` implements is based on solving a much easier LP problem, however, such an approach is only possible with polyhedra.

5.2 Dimensional study

In this subsection we study how the proposed methodology scales with the dimensions of $\mathcal{A}$ and $\mathcal{B}$ for non-polyhedral domains. To do so, we consider the problem of computing $\mathcal{A} \ominus \mathcal{B}$ with $\mathcal{A} = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i^4 \leq 1\}$ and $\mathcal{B} = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 \leq 0.04\}$ for increasing values of n. The obtained results are reported in Table 3.

n	$\deg(s_j)$	$\deg(c)$	e_{max} ($e_{max,n}$)	t
2	6	8	$3 \cdot 10^{-4}$ (0.0015)	0.57 s
3	6	8	$9 \cdot 10^{-4}$ (0.0047)	27 s
4	6	6	0.001 (0.005)	16 min
5	4	6	0.018 (0.088)	3 min
6	4	4	0.015 (0.078)	8 min
7	4	4	0.019 (0.097)	1h 12 min [3]
8	2	4	0.026 (0.13)	18.9 s
9	2	4	0.03 (0.15)	77.54 s
10	2	4	0.035 (0.17)	3 min 15 s

Table 3
Degree of the $s_j(x, z)$, degree of the $c(x)$, maximum and normalized error, and computational time for $n \in \{2, 3, \ldots, 10\}$.

We note that when performing these tests we have never encountered the numerical issues (*e.g.* large residuals, poor convergence) that typically plague naïve formulations of SOS problems. This suggests that the P-difference SOS formulation proposed in this paper tends to be numerically well-conditioned. This fact allowed us to compute P-differences in fairly high number of dimensions and still obtaining good approximations. In fact, the limit we reached and that did not allow us to use higher degrees polynomials (and thus better approximations) for $n \geq 5$ was due to hardware limitations (in particular RAM overflow) and could be arguably overcome by more powerful machines and/or better solvers.

6 Concluding remarks

In this paper, we proposed a systematic approach for inner approximating the Pontryagin difference between two basic semi-algebraic sets based on SOSP. We showcased the effectiveness of this methodology by applying it to several different examples up to dimension 10. Possible applications for this methodology include the analytical determination of an inner approximation of constraint sets in robust control.

References

[1] L. S. Pontryagin, Linear differential games. i, ii, in: Doklady Akademii Nauk, Vol. 175, Russian Academy of Sciences, 1967, pp. 764–766.

[2] B. Kouvaritakis, M. Cannon, Model predictive control, Switzerland: Springer International Publishing (2016).

[3] J. B. Rawlings, D. Q. Mayne, M. Diehl, Model predictive control: theory, computation, and design, Vol. 2, Nob Hill Publishing Madison, WI, 2017.

[4] F. Blanchini, S. Miani, Set-theoretic methods in control, Springer, 2008.

[5] I. Kolmanovsky, E. G. Gilbert, Theory and computation of disturbance invariant sets for discrete-time linear systems, Mathematical problems in engineering 4 (1998).

[2] An n-simplex is the n-dimensional generalization of a triangle: a 2-simplex is a triangle, a 3-simplex is a tetrahedron, a 4-simplex is a 5-cell . . . *etc.*

[3] Note that this high computational time is to be attributed to the high dimensionality and degrees, which translates into a large optimization problem. As per the error indices, convergence is satisfactory.

[6] H. J. Heijmans, Mathematical morphology: A modern approach in image processing based on algebra and geometry, SIAM review 37 (1) (1995) 1–36.

[7] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, D. Manocha, Porca: Modeling and planning for autonomous driving among many pedestrians, IEEE Robotics and Automation Letters 3 (4) (2018) 3418–3425.

[8] R. M. Haralick, S. R. Sternberg, X. Zhuang, Image analysis using mathematical morphology, IEEE transactions on pattern analysis and machine intelligence (4) (1987) 532–550.

[9] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-Parametric Toolbox 3.0, in: Proc. of the European Control Conference, Zürich, Switzerland, 2013, pp. 502–510, `http://control.ee.ethz.ch/~mpt`.

[10] M. Althoff, On computing the minkowski difference of zonotopes, arXiv preprint arXiv:1512.02794 (2015).

[11] H. Barki, F. Denis, F. Dupont, A new algorithm for the computation of the minkowski difference of convex polyhedra, in: 2010 Shape Modeling International Conference, IEEE, 2010, pp. 206–210.

[12] A. Marzollo, A. Pascoletti, Computational procedures and properties of the geometrical difference of sets, Journal of Mathematical Analysis and Applications 54 (3) (1976) 772–785.

[13] R. C. Gonzales, R. E. Woods, Digital image processing (2002).

[14] G. Stengle, A nullstellensatz and a positivstellensatz in semialgebraic geometry, Mathematische Annalen 207 (2) (1974) 87–97.

[15] F. Dabbene, D. Henrion, C. M. Lagoa, Simple approximations of semialgebraic sets and their applications to control, Automatica 78 (2017) 110–118.

[16] P. A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, Ph.D. thesis, California Institute of Technology (2000).

[17] M. ApS, MOSEK Optimizer API for C 9.2.29 (2019).
URL `https://docs.mosek.com/9.2/capi/index.html`

[18] Addendum to "A Sum-of-Squares-Based Procedure to Approximate the Pontryagin Difference of Semialgebraic Sets" (2020).
URL `http://www.gprix.it/SoSPontryagin.pdf`

[19] LazySets.jl github site (2018).
URL `https://github.com/JuliaReach/LazySets.jl`

A Definitions of Cone and Monoid

To make this paper self-contained we now present the definitions of the algebraic structures used in Section 3

Definition 4 *The multiplicative monoid generated by a set of polynomials P, $\mathbf{M}(P)$ with $P = \{p_i\}_{i=1}^m \in \mathbb{R}[x]$, $x \in R^n$ is defined as* $\mathbf{M}(P) = \left\{ \prod_{i=1}^m p_i^{k_i}, \{k_i\}_{i=1}^m \in \mathbb{Z}_{\geq 0} \right\}$.

Definition 5 *The cone generated by a set of polynomials P, $\mathbf{K}(P)$, with $P = \{p_i\}_{i=1}^m \in \mathbb{R}[x]$, $x \in \mathbb{R}^n$ is defined as* $\mathbf{K}(P) = \{s_0 + \sum_{i=1}^r s_i b_i : \{s_i\}_{i=0}^r \in \Sigma[x], \{b_i\}_{i=1}^r \in \mathbf{M}(P)\}$.