NOVEL REPRESENTATIONS OF SEMIALGEBRAIC SETS ARISING IN PLANNING AND CONTROL

by
James Guthrie

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
October 2022

Abstract

The mathematical notion of a set arises frequently in planning and control of autonomous systems. A common challenge is how to best represent a given set in a manner that is efficient, accurate, and amenable to computational tools of interest.

For example, ensuring a vehicle does not collide with an obstacle can be generically posed in multiple ways using techniques from optimization or computational geometry. However these representations generally rely on executing algorithms instead of evaluating closed-form expressions. This presents an issue when we wish to represent an obstacle avoidance condition within a larger motion planning problem which is solved using nonlinear optimization. These tools generally can only accept smooth, closed-form expressions. As such our available representations of obstacle avoidance conditions, while accurate, are not amenable to the relevant tools.

A related problem is how to represent a set in a compact form without sacrificing accuracy. For example, we may be presented with point-cloud data representing the boundary of an object that our vehicle must avoid. Using the obstacle avoidance conditions directly on the point-cloud data would require performing these calculations with respect to each point individually. A more efficient approach is to first approximate the data with simple geometric shapes and perform later analysis with the approximation. Common shapes include bounding boxes, ellipsoids, and superquadrics. These shapes are convenient in that they have a compact representation and we have good heuristic objectives for fitting the data. However, their primitive nature means accuracy of representation may suffer. Most notably, their inherent symmetry makes

them ill-suited for representing asymmetric shapes. In theory we could consider more complicated shapes given by an implicit function $\mathcal{S} = \{x \mid f(x) \leq 1\}$. However we lack reliable methods for ensuring a good fit.

This thesis proposes novel approaches to these problems. Throughout, the sets of interest are described by polynomial inequalities, making them semialgebraic.

Thesis Committee

Enrique Mallada (Primary Advisor)
Associate Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Mahyar Fazlyab
Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins University

Marin Kobilarov
Assistant Professor
Department of Mechanical Engineering
Johns Hopkins University

Acknowledgements

First I would like to thank Enrique Mallada for serving as my advisor. He gave me the freedom to pursue my own research interests as they evolved over time. Throughout, Enrique was always willing to invest time discussing and refining my work and brainstorming new directions. I benefited greatly from his guidance and assistance over the course of my graduate studies.

I would also like to thank my thesis committee for reviewing this work and providing helpful feedback. I appreciate the time sacrifice that this required.

Lastly I would like to thank my wife Emily. She has been a source of constant encouragement during this endeavor.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Sets, operations between sets, and constraints on sets are ubiquitous in planning and control of autonomous systems. A common challenge is how to best represent a given set in a manner that is efficient, accurate, and amenable to computational tools. For example, ensuring an autonomous vehicle does not collide with an obstacle can be generically posed as the constraint $\mathcal{V} \cap \mathcal{O} = \emptyset$, where $\mathcal{V}, \mathcal{O} \subseteq \mathbb{R}^n$ represent the space occupied by the vehicle and obstacle respectively. However, in representing this condition we have multiple options:

- **Minkowski**: Require the Minkowski difference does not contain the origin:

$$\mathcal{V} \cap \mathcal{O} = \emptyset \iff \{v - o \,|\, v \in \mathcal{V}, o \in \mathcal{O}\} \cap 0 = \emptyset \tag{1.1}$$

- **Distance**: Require a positive distance between the sets:

$$\mathcal{V} \cap \mathcal{O} = \emptyset \iff \min\{\|v - o\| \,|\, v \in \mathcal{V}, o \in \mathcal{O}\} > 0 \tag{1.2}$$

- **Support Function**: Require the sets can be separated by a hyperplane:

$$\mathcal{V} \cap \mathcal{O} = \emptyset \Leftarrow \exists\, c \in \mathbb{R}^n,\ \inf_{v \in \mathcal{V}} c^T v > \sup_{o \in \mathcal{O}} c^T o \tag{1.3}$$

While all of these conditions ensure the vehicle avoids the obstacle, none of these representations provide closed-form expressions. Instead they require querying a computational geometry routine (Minkowski) or an optimization solver (Distance, Support Function) to determine if the condition holds. If we wish to incorporate this obstacle avoidance condition within a larger nonlinear optimization problem for motion planning it is not immediately clear how to do so, as these solvers generally require constraints be expressed in closed-form. As such, these representations are accurate and can be efficiently calculated, but they are not amenable to the relevant tools.

A related problem is how to represent a set in a compact form without sacrificing accuracy. For example, we may be presented with point-cloud data representing the boundary of an object that our vehicle must avoid. Using the obstacle avoidance conditions directly on the point-cloud data would require performing these calculations with respect to each point individually. A more efficient approach is to first approximate the data with simple geometric shapes and perform later analysis with the approximation. Common shapes include bounding boxes, ellipsoids, and superquadrics. These shapes are convenient in that they have a compact representation and we have good heuristic objectives for fitting the data. For example, the minimum volume outer ellipsoid (also called the outer Löwner-John ellipsoid) can be found via semidefinite optimization. However, their primitive nature means accuracy of representation may suffer. Most notably, their inherent axis symmetry makes them ill-suited to representing asymmetric shapes. In theory we could consider more complicated shapes given by an implicit function $\mathcal{S} = \{x \mid f(x) \leq 1\}$. However we lack reliable methods for ensuring a good fit.

This thesis proposes novel approaches to these problems. Throughout, the sets of interest are described by polynomial inequalities, making them semialgebraic. Specifically we address the following questions:

- **What is a good heuristic for approximating star-convex sets with an**

implicit function ?

- **How can we efficiently represent sums and products of sets in the complex plane?**
- **How can we represent obstacle avoidance constraints in a manner amenable to nonlinear optimization ?**
- **How can we ensure obstacle avoidance constraints are satisfied in continuous-time despite planning with discrete-time models ?**

1.2 Thesis Outline

This dissertation is composed of four papers which have been edited for consistency of presentation.

Inner and Outer Approximations of Star-Convex Semialgebraic Sets

This chapter considers the problem of approximating a semialgebraic set with a sublevel-set of a polynomial using sum-of-squares optimization. In this setting, it is standard to seek a minimum volume outer approximation or maximum volume inner approximation. This is made difficult by the lack of a known relationship between the coefficients of an arbitrary polynomial and the volume of its sublevel sets. Previous works have proposed heuristics based on the determinant and trace objectives commonly used in ellipsoidal fitting. We propose a novel objective which yields both an outer and an inner approximation while minimizing the ratio of their respective volumes. This objective is scale-invariant and easily interpreted. We provide justification for its use in approximating star-convex sets. Numerical examples demonstrate that the approximations obtained are often tighter than those returned by existing heuristics when applied to convex and star-convex sets. We also provide algorithms for establishing the star-convexity of a semialgebraic set by finding inner and outer approximations of its kernel. Published in IEEE Control Systems Letters.

Outer Approximations of Minkowski Operations on Complex Sets via Sum-of-Squares Optimization

This chapter provides methods for outer approximating operations on sets in the complex plane. Such operations arise in robust analysis for control systems [1], geometric optics [2], and convergence analysis of optimization algorithms [3]. Using polar coordinates, we pose this as an optimization problem in which we find a pair of contours that give lower and upper bounds on the radial distance at a given angle. Through a series of variable transformations we rewrite this as a sum-of-squares optimization problem. Numerical examples are given to demonstrate the performance. Presented at American Control Conference 2021.

Closed-Form Minkowski Sum Approximations for Efficient Optimization-Based Collision Avoidance

Motion planning methods for autonomous systems based on nonlinear programming offer great flexibility in incorporating various dynamics, objectives, and constraints. One limitation of such tools is the difficulty of efficiently repre senting obstacle avoidance conditions for non-trivial shapes. For example, it is possible to define collision avoidance constraints suitable for nonlinear programming solvers in the canonical setting of a circular robot navigating around M convex polytopes over N time steps. However, it requires introducing $(2 + L)MN$ additional constraints and LMN additional variables, with L being the number of halfplanes per polytope, leading to larger nonlinear programs with slower and less reliable solving time. In this chapter, we overcome this issue by building closed-form representations of the collision avoidance conditions by outer-approximating the Minkowski sum conditions for collision. Our solution requires only MN constraints (and no additional variables), leading to a smaller nonlinear program. On motion planning problems for an autonomous car and quadcopter in cluttered environments, we achieve speedups of 4.8x and 8.7x respectively with significantly less variance in solve times and negligible impact on performance arising from the use of outer approximations. Presented at American Control Conference 2022.

A Differentiable Signed Distance Representation for Continuous Collision Avoidance in Optimization-Based Motion Planning

This chapter proposes a new set of conditions for exactly representing collision avoidance constraints within optimization-based motion planning algorithms. The conditions are continuously differentiable and therefore suitable for use with standard nonlinear optimization solvers. The method represents convex shapes using a support function representation and is therefore quite general. For collision avoidance involving polyhedral or ellipsoidal shapes, the proposed method introduces fewer variables and constraints than existing approaches. Additionally the proposed method can be used to rigorously ensure continuous collision avoidance as the vehicle transitions between the discrete poses determined by the motion planning algorithm. Numerical examples demonstrate how this can be used to prevent problems of corner cutting and passing through obstacles which can occur when collision avoidance is only enforced at discrete time steps. Accepted for presentation at IEEE Conference on Decision and Control 2022.

1.3 Additional Publications

Earlier conference publications by the author on power system nonlinear stability [4], and non-convex model predictive control problems [5, 6] are not included in the thesis due to the difference in subject matter. We reference them here for completeness.

Chapter 2

Inner and Outer Approximations of Star-Convex Semialgebraic Sets

2.1 Introduction

Consider a compact, semialgebraic set $\mathcal{X} \subset \mathbb{R}^n$ given by the intersection of the 1-sublevel sets of m polynomial functions $g_i(x) \in \mathbb{R}[x]$:

$$\mathcal{X} = \{x \,|\, g_i(x) \leq 1,\, i \in [m]\}. \tag{2.1}$$

Semialgebraic sets arise naturally in many control applications. The set of coefficients for which a polynomial is Schur or Hurwitz stable is given by a semialgebraic set. For Hurwitz stability, the polynomial inequalities can be derived from the Routh array. These sets are often complicated and cumbersome to analyze. As such, it is common to seek simpler representations which closely approximate the set but are more amenable to further analysis [7]. Examples of “simple” representations include hyperrectangles and ellipsoids.

A number of publications have explored the use of sum-of-squares (SOS) optimization for approximating a semialgebraic set with a simpler representation [7–14]. The most common parameterization is to seek a SOS polynomial whose 1-sublevel set $\mathcal{F} = \{x \,|\, f(x) \leq 1\}$ provides either an inner ($\mathcal{F} \subseteq \mathcal{X}$) or outer ($\mathcal{F} \supseteq \mathcal{X}$) approximation of the set $\mathcal{X}$. In this formulation, an open question is the choice of

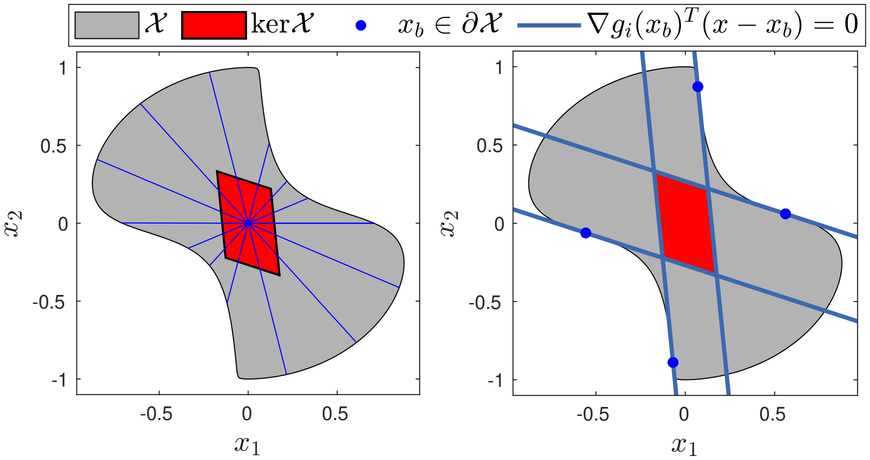

$\mathcal{X}$
$\ker\mathcal{X}$
$x_b \in \partial\mathcal{X}$
$\nabla g_i(x_b)^T(x - x_b) = 0$
1
0.5
0
-0.5
-1
-0.5
0
0.5
x_1
x_2

Figure 2-1. The kernel is the convex set of points $p \in \mathcal{X}$ such that the line segment $\overline{pq} \subseteq \mathcal{X}$ for any $q \in \mathcal{X}$ (left). It is given by the intersection of all linearized active constraints $g_i(x_b) = 1$ defining $\partial\mathcal{X}$ (right).

the objective function. For outer (resp. inner) approximations, a natural objective is to minimize (resp. maximize) the volume of the 1-sublevel set. For an ellipsoid $\mathcal{E} = \{x \,|\, x^T Ax + b^T x + c \leq 1\}$ where $A \succeq 0$, the volume is proportional to $\det A^{-1}$. Using the logarithmic transform, ellipsoidal volume minimization can be posed as the convex objective $-\operatorname{logdet} A$ [8]. More generally, in the case of homogeneous polynomials it is possible to find the minimum volume outer approximation by solving a hierarchy of semidefinite programs [15].

Ellipsoids and homogeneous polynomials are not ideal candidates for approximating asymmetric shapes due to their inherent symmetry. General polynomials offer a more flexible basis for approximating sets. The caveat is that we lack expressions for computing the volume of the 1-sublevel set as a function of the polynomial coefficients. The most common approach is to mimic the determinant ([8, 10]) or trace [7] objectives used in ellipsoidal fitting. These objectives often yield qualitatively good approximations. However, they have no explicit relationship to the volume beyond upper bounding it in some cases [7]. Thus it is difficult to infer the quality of an approximation from the objective value attained.

2.1.1 Contributions

This chapter makes the following contributions:

- We propose and justify an algorithm based on SOS optimization for jointly finding an inner and outer approximation of a semialgebraic set. The algorithm minimizes the volume of the outer approximation relative to the volume of the inner approximation. This objective is easily interpreted and scale-invariant.
- We provide numerical examples showing that our algorithm tends to yield better approximations than existing methods when applied to star-convex sets.
- We provide algorithms for finding inner and outer approximations of the kernel of a star-convex set as shown in Figure 2-1.

The chapter is organized as follows. Section II defines the problem we address and reviews the notion of star-convexity. Section III surveys existing volume heuristics for SOS-based set approximation. Section IV proposes a new volume heuristic for finding outer and inner approximations. Section V provides methods for approximating the kernel of a star-convex set. Section VI provides numerical examples. Section VII concludes the chapter.

2.1.2 Notation

Let $i \in [k] := \{1, \ldots, k\}$. Let $\mathbb{Z}^+$ denote the set of positive integers. Let $S^{n-1} := \{x \in \mathbb{R}^n \mid \|x\| = 1\}$. The notation $P \succeq 0$ indicates that the symmetric matrix P is positive semidefinite (PSD). Given a compact set $\mathcal{X} \subset \mathbb{R}^n$, its volume (formally, Lebesgue measure) is denoted $\operatorname{vol} \mathcal{X} := \int_{\mathcal{X}} dx$. Let $\sigma_{\mathcal{X}}(c) := \max_{x \in \mathcal{X}} c^T x$ denote the support function of $\mathcal{X}$ where $c \in S^{n-1}$. Given sets $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^n$ the (bi-directional) Hausdorff distance is $d_H(\mathcal{A}, \mathcal{B}) := \max(h(\mathcal{A}, \mathcal{B}), h(\mathcal{B}, \mathcal{A}))$ where $h(\mathcal{A}, \mathcal{B}) := \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} \|a - b\|_2$.

The α-sublevel set of a function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is $\{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$. For $x \in \mathbb{R}^n$, let $\mathbb{R}[x]$ denote the set of polynomials in x with real coefficients. Let $\mathbb{R}_d[x]$ denote the set of all polynomials in $\mathbb{R}[x]$ of degree less than or equal to d. A polynomial $p(x) \in \mathbb{R}[x]$ is a SOS polynomial if there exists polynomials $q_i(x) \in \mathbb{R}[x], i \in [j]$ such that $p(x) = q_1^2(x) + \ldots + q_j^2(x)$. We use $\Sigma[x]$ to denote the set of SOS polynomials in x. A polynomial of degree $2d$ is a SOS polynomial if and only if there exists $P \succeq 0$ (the Gram matrix) such that $p(x) = z(x)^T P z(x)$ where $z(x)$ is the vector of all monomials of x up to degree d [16]. Letting $m := \binom{n+d}{d}$ denote the length of $z(x)$, we have that $P \in \mathbb{R}^{m \times m}$. To minimize notational clutter, we will sometimes list a polynomial $f(x)$ as a decision variable. It is implied that a degree is specified and matrix P is introduced as a decision variable such that $f(x) = z(x)^T P z(x)$.

2.2 Problem Statement

Definition 2.1 (Star-Convex Set [17])**.** A set $\mathcal{S} \subseteq \mathbb{R}^n$ is star-convex if it has a non-empty kernel. The kernel is

$$\ker \mathcal{S} := \{x \mid tx + (1-t)y \in \mathcal{S} \, \forall \, t \in [0,1], y \in \mathcal{S}\}. \tag{2.2}$$

The kernel is the set of points in $\mathcal{S}$ from which one can "see" all of $\mathcal{S}$ as shown in Figure 2-1. It is easily shown that the kernel is convex. If $\mathcal{S}$ is convex then $\ker \mathcal{S} = \mathcal{S}$.

We will be interested in approximating the set (2.1) for the case in which it is star-convex with respect to the origin.

Problem 2.1 (Star-Convex Set Approximation)**.** Given a compact, semialgebraic set $\mathcal{X}$ with $0 \in \text{int}\mathcal{X} \cap \ker\mathcal{X}$ and $d \in \mathbb{Z}^+$ find a polynomial $f_o(x) \in \mathbb{R}_{2d}[x]$ ($f_i(x) \in \mathbb{R}_{2d}[x]$) whose 1-sublevel set $\mathcal{F}_o$ ($\mathcal{F}_i$) is of minimum (maximum) volume and is an outer (inner) approximation of $\mathcal{X}$:

$$\min_{f_o(x) \in \mathbb{R}_{2d}[x]} \text{vol } \mathcal{F}_o \text{ s.t. } \mathcal{X} \subseteq \mathcal{F}_o$$

$$\left(\max_{f_i(x) \in \mathbb{R}_{2d}[x]} \text{vol } \mathcal{F}_i \text{ s.t. } \mathcal{F}_i \subseteq \mathcal{X} \right).$$

To establish star-convexity of $\mathcal{X}$, we seek polytopic approximations of its kernel.

Problem 2.2 (Kernel Approximation)**.** Given a semialgebraic set $\mathcal{X} \subset \mathbb{R}^n$ find a polytope $\mathcal{K}_o$ ($\mathcal{K}_i$) of minimum (maximum) volume that is an outer (inner) approximation of ker$\mathcal{X}$:

$$\min \text{vol } \mathcal{K}_o \text{ s.t. } \ker\mathcal{X} \subseteq \mathcal{K}_o$$

$$\left(\max \text{vol } \mathcal{K}_i \text{ s.t. } \mathcal{K}_i \subseteq \ker\mathcal{X}\right).$$

2.3 Existing Volume Heuristics for Set Approximation

We review existing heuristics for approximating semialgebraic set $\mathcal{X}$ using SOS optimization. Each of these methods finds an even-degree polynomial $f(x) = z(x)^T P z(x)$. The variations between the methods largely relate to the objective applied to Gram matrix P. For general polynomials, there is no known relationship between P and the volume of the sublevel sets. Thus the following objectives are all heuristics in some sense.

2.3.1 Determinant Maximization

In [8], the authors propose maximizing the determinant of the Hessian $\nabla^2 f(x)$ of SOS polynomials. If f is a polynomial of degree 2, this reduces to the ellipsoidal objective $-\det A$ for $\mathcal{E} = \{x \,|\, x^T A x + b^T x + c \leq 1\}, A \succeq 0$. As the Hessian must be PSD, the outer approximation is convex. This makes it ill-suited to approximating non-convex shapes.

In [10], the authors propose performing determinant maximization directly on the Gram matrix P. The Hessian is no longer required to be PSD. This allows non-convex outer approximations to be found.

2.3.2 Inverse Trace Minimization

The determinant maximization objective minimizes the product of the eigenvalues of P^{-1}. In [10], the authors propose an alternative heuristic of minimizing the sum of the eigenvalues of P^{-1}. This requires an additional matrix variable V and constraint $V \succeq P^{-1}$. Using the Schur complement this can be written as a block matrix constraint involving V and P (vice P^{-1}). The objective min tr$V$ then indirectly minimizes the sum of the eigenvalues of P^{-1}.

2.3.3 L1 Minimization

In [7] the authors propose minimizing the l_1 norm of a polynomial evaluated over a bounding box $\mathcal{B} \supseteq \mathcal{X}$. This approach was first introduced in [18] for approximating the volume of semialgebraic sets. Using hyperrectangles as bounding boxes, one can integrate the polynomial over $\mathcal{B}$. The resulting objective $l_1(f(x)) := \int_{\mathcal{B}} f(x)\,dx$ is linear in terms of P. The outer approximation consists of the intersection of the 1-superlevel set of $f(x)$ and $\mathcal{B}$:

$$\mathcal{X} \subseteq (\mathcal{B} \cap \{x \,|\, f(x) \geq 1\}). \tag{2.3}$$

This differs from other objectives which do not rely on bounding boxes as part of the set approximation.[1] In this setting, $f(x)$ is approximating the indicator function of $\mathcal{X}$ over a compact set $\mathcal{B}$. Convergence of $f(x)$ to the true indicator function in the limit (as degree $d \to \infty$) can be shown by leveraging the Stone-Weierstrass theorem. The asymptotic rate of convergence is at least $O(1/\log \log d)$ [19]. Inner approximations can be found by outer approximating the complement of $\mathcal{X}$.

[1] One application of approximating semialgebraic sets is to yield a single sufficient condition for ensuring $x \notin \mathcal{X}$, which can be incorporated into a nonlinear optimization problem (e.g. obstacle avoidance in motion planning [13]). The presence of the bounding box in the resulting set description would require logical constraints to represent $(f(x) < 1 \vee x \notin \mathcal{B}) \implies x \notin \mathcal{X}$ which are generally unsupported in nonlinear optimization solvers.

2.4 Inner and Outer Approximations of Star-Convex Sets

We propose a new volume heuristic for solving Problem 2.1. Our heuristic is inspired by the following two lemmas.

Lemma 2.1. *Let $\mathcal{X}, \mathcal{F}$ be compact sets in $\mathbb{R}^n$ such that $\mathcal{F} \subseteq \mathcal{X}$. Let $0 \in \operatorname{int} \mathcal{F}$. Then there exists a scaling $s \geq 1$ such that $\mathcal{X} \subseteq s\mathcal{F}$.*

Lemma 2.2. *Let $\mathcal{X} \subset \mathbb{R}^n$. Let $s\mathcal{X} = \{sx \mid x \in \mathcal{X}\}$ denote the scaled set where $s \geq 0$. Then* $\operatorname{vol} s\mathcal{X} = s^n \cdot \operatorname{vol}\mathcal{X}$.

Thus given an inner approximation $\mathcal{F}$, we can obtain an outer approximation $s\mathcal{F}$ for some $s \geq 1$ with relation

$$\frac{\operatorname{vol} s\mathcal{F}}{\operatorname{vol} \mathcal{F}} = s^n. \tag{2.4}$$

By minimizing s we minimize the ratio of the outer approximation volume to the inner approximation volume. Figure 2-2 visualizes this intuitive heuristic for approximating a set.

We seek a polynomial $f : \mathbb{R}^n \to \mathbb{R}$ whose 1-sublevel set $\mathcal{F} = \{x \mid f(x) \leq 1\}$ is an inner approximation of $\mathcal{X}$. We turn this into a condition involving the complement of $\mathcal{X}$:

$$\mathcal{F} \subseteq \mathcal{X} \iff f(x) > 1 \,\forall\, x \in \mathcal{X}^{\mathsf{c}}. \tag{2.5}$$

Optimization methods require non-strict inequalities. We approximate the strict inequality by introducing a small constant $\epsilon > 0$ and working with the closure of the complement of $\mathcal{X}$. Define the following:

$$\bar{\mathcal{X}} = \bigcup_{i \in [m]} \{x \mid g_i(x) \geq 1\}. \tag{2.6}$$

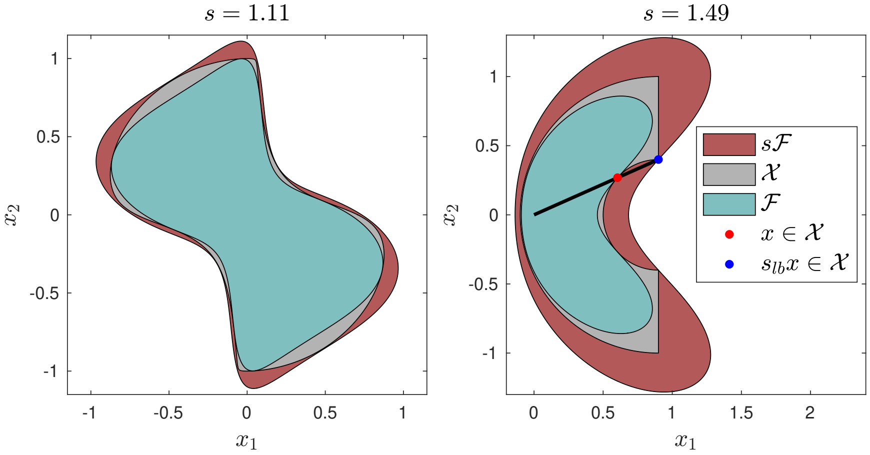


Figure 2-2. 4th-order approximations of star-convex set (left) and non-star-convex set (right) found by minimizing scaling term s. The non-star-convex set has a lower bound $s_{lb} > 1$ on the achievable approximation scaling s.

We then use the following approximation of (2.5):

$$\mathcal{F} \subset \text{int}\mathcal{X} \Leftarrow f(x) \geq 1 + \epsilon \, \forall x \in \bar{\mathcal{X}}. \tag{2.7}$$

Next, we scale the set $\mathcal{F}$ by a scaling variable $s > 1$ to obtain an outer approximation:

$$s\mathcal{F} \supseteq \mathcal{X} \iff f(\tfrac{x}{s}) \leq 1 \, \forall x \in \mathcal{X}. \tag{2.8}$$

Combining the above we arrive at the following:

$$\begin{aligned} \min_{f(x),\, s} \quad & s \\ \text{s.t.} \quad & \\ & f(x) \geq 1 + \epsilon \, \forall x \in \bar{\mathcal{X}}, \\ & f(\tfrac{x}{s}) \leq 1 \qquad \forall x \in \mathcal{X}. \end{aligned} \tag{2.9}$$

Remark 2.1. Our scaling heuristic is applicable to approximating any compact set containing the origin in its interior. However, it is best suited to approximating star-convex sets in which $0 \in \text{int}\mathcal{X} \cap \ker \mathcal{X}$ as visualized in Figure 2-2. Otherwise there exists a lower bound s_{lb} such that $1 < s_{lb} \leq s$ in (2.9).

Lemma 2.3. *Let $\mathcal{X}$ and $\mathcal{F}$ be compact sets in $\mathbb{R}^n$. Let $\mathcal{F} \subseteq \mathcal{X} \subseteq s^\star \mathcal{F}$ for some $s^\star > 1$. Let $0 \in \operatorname{int} \mathcal{F}$. Let $x, sx \in \mathcal{X}$ and $tx \notin \mathcal{X} \, \forall \, t \in (1, s)$ for some $s > 1, x \neq 0$. Then $s^\star \geq s$.*

We let s_{lb} denote the greatest lower bound given by Lemma 2.3. This imposes a minimum volume ratio between the inner and outer approximation. Figure 2-2 (right) visualizes this result. The set is not star-convex and therefore $0 \notin \ker \mathcal{X}$. The black line segment connecting the origin to point $s_{lb}x$ is not contained in $\mathcal{X}$. This point imposes a lower bound on s, preventing the inner and outer approximations from coming closer together.

We introduce SOS polynomials $\lambda_i(x), \mu_i(x), i \in [m]$ and replace the set-containment conditions in (2.9) with SOS conditions.[2] If s is left as a decision variable, we would have bilinear terms involving the coefficients of $f(x)$ and s. Instead we perform a bisection over s, solving a feasibility problem at each iteration as given by (2.10). Algorithm 1 details the bisection method.

Optimization Problem: FindApprox(s, g_i)

$$\begin{aligned}
\min_{f(x), \lambda_i(x), \mu_i(x)} \quad & 0 \\
\text{s.t.} \quad & \\
& f(x) - (1+\epsilon) - \lambda_i(x)(g_i(x) - 1) \in \Sigma[x], i \in [m], \\
& 1 - f(\tfrac{x}{s}) - \sum_{i=1}^{m} \mu_i(x)(1 - g_i(x)) \in \Sigma[x], \\
& \lambda_i(x), \mu_i(x) \in \Sigma[x], \qquad i \in [m].
\end{aligned} \tag{2.10}$$

Remark 2.2. The objective is scale-invariant. Let solution $(f^*(x), s^*)$ define an outer and inner approximation of $\mathcal{X}$. Scale $\mathcal{X}$ by $\alpha > 0$, replacing constraints $g_i(x)$ with

[2] For the outer approximation of the compact set $\mathcal{X}$, the SOS conditions are necessary and sufficient by Putinar's Positivstellensatz when $\mu_i(x)$ is of high-enough degree and the defining polynomials g_i satisfy the Archimedean assumption [20]. The inner approximation constraint involves an unbounded set. The associated SOS reformulation utilizes the generalized $\mathcal{S}$-procedure which is only sufficient [16].

Algorithm 1 Inner and Outer Approximation of $\mathcal{X}$

```
Input: $\mathcal{X} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 1, i \in [m]\}$, $s_{tol} > 0$
Output: $\mathcal{F}, s\mathcal{F}$ s.t. $\mathcal{F} \subseteq \mathcal{X} \subseteq s\mathcal{F}$
    $s_{ub} \leftarrow 1 + s_{tol}, s_{lb} \leftarrow 1$
    while FindApprox$(s_{ub}, g_i)$ = Infeasible do
        $s_{lb} \leftarrow s_{ub}$
        $s_{ub} \leftarrow 2s_{ub}$
    while $s_{ub} - s_{lb} > s_{tol}$ do
        $s_{try} \leftarrow 0.5(s_{ub} + s_{lb})$
        if FindApprox$(s_{try}, g_i)$ = Infeasible then
            $s_{lb} \leftarrow s_{try}$
        else
            $s_{ub} \leftarrow s_{try}$
    return FindApprox$(s_{ub}, g_i)$
```

$g_i(\frac{x}{\alpha})$. Then the solution pair $(f^*(\frac{x}{\alpha}), s^*)$ defines the new approximation, where the objective value remains unchanged. The objective is not translation-invariant however. For example, assume we approximate a star-convex set exactly with $(f^\star(x), s^\star = 1)$. Translate $\mathcal{X}$ by $t \in \mathcal{X} \setminus \ker\mathcal{X}$, replacing $g_i(x)$ with $g_i(x - t)$. Then $0 \notin \ker\mathcal{X}$ and $s^\star > 1$ for any approximation by Lemma 2.3.

Remark 2.3. If $\mathcal{F}$ is convex we can relate the scaling s to the Hausdorff distance between the approximations.

Lemma 2.4. *Let $\mathcal{F} \subset \mathbb{R}^n$ be a convex, compact set and $s \geq 1$. Then the following holds:*

$$d_H(s\mathcal{F}, \mathcal{F}) = (s - 1) \cdot \max_{x \in \mathcal{F}} \|x\|_2. \tag{2.11}$$

2.5 Sampling-Based Approximations of the Kernel

Algorithm 1 assumed the set $\mathcal{X}$ contained the origin in its kernel. If this does not hold, but there exists a point $x^\star \in \ker\mathcal{X} \cap \mathrm{int}\mathcal{X}$ we can apply Algorithm 1 to the translated set $\{x - x^* \mid x \in \mathcal{X}\}$. As our objective is not invariant with respect to translation, it is useful to approximate the kernel to establish possible choices for $x^\star$.[3] In this section

[3] A practical heuristic is to let $x^\star$ be the Chebyshev center of $\ker\mathcal{X}$.

we provide algorithms for finding polytopic approximations of $\ker \mathcal{X}$.

It will be convenient to represent the boundary of $\mathcal{X}$ in terms of the inequality that is active. Define the following:

$$\partial \mathcal{X}_i = \{x \mid g_i(x) = 1, g_j(x) \leq 1, j \in [m] \setminus i\}. \tag{2.12}$$

The boundary of $\mathcal{X}$ is given by the union

$$\partial \mathcal{X} = \bigcup_{i \in [m]} \partial \mathcal{X}_i. \tag{2.13}$$

Lemma 2.5. *Let $\mathcal{X}$ be a semialgebraic set as defined in* (2.1)*. Let $\nabla g_i(x_b) \neq 0 \, \forall \, x_b \in \partial \mathcal{X}_i, i \in [m]$. The kernel of $\mathcal{X}$ is given by the following semialgebraic set:*

$$\text{ker}\mathcal{X} = \{x_k \mid \nabla g_i(x_b)^T (x_k - x_b) \leq 0 \, \forall \, x_b \in \partial \mathcal{X}_i, i \in [m]\}.$$

Remark. From Lemma 2.5 we see that the kernel of $\mathcal{X}$ is defined by cutting-planes tangent to the active constraint $g_i(x_b) = 1, x_b \in \partial \mathcal{X}$ as shown in Figure 2-1.

Remark. Lemma 2.5 assumes the gradient of an active constraint is non-zero. While restrictive, we note that this assumption is typically satisfied in sets of practical interest.

We provide sampling-based algorithms for finding outer and inner approximations of this set. If the outer approximation is empty, this is sufficient to conclude that the set $\mathcal{X}$ is not star-convex. Conversely, if the inner approximation is not empty this is sufficient to establish that $\mathcal{X}$ is star-convex. In the case that the outer approximation is not empty and the inner approximation is empty we cannot conclude anything about the star-convexity of the set.

2.5.1 Outer Approximation

We assume the existence of an oracle Sample($\partial \mathcal{X}$) which allows us to randomly sample points $x_b \in \partial \mathcal{X}$ and identify the set of active constraints $\mathcal{I} = \{i \mid i \subseteq [m], g_i(x_b) = 1\}$.[4]

[4] Starting from a point in the interior of $\mathcal{X}$, one can choose a direction and find a boundary point via bisection. Alternatively, nonlinear optimization may be used to find boundary points.

From Lemma 2.5, each sample defines a cutting plane satisfied by $\ker\mathcal{X}$. We collect these constraints to form an outer approximation $\mathcal{K}_o \supseteq \ker\mathcal{X}$. If at any point, $\mathcal{K}_o = \emptyset$ (which can be determined using Farkas' Lemma) we terminate as this implies $\ker\mathcal{X} = \emptyset$. Algorithm 2 summarizes the method.

Algorithm 2 Outer Approximation of $\ker\mathcal{X}$

```
Input: $\mathcal{X} = \{x \in \mathbb{R}^n \,|\, g_i(x) \le 1, i \in [m]\}$, $n_s \ge 1$
Output: Outer Approximation $\mathcal{K}_o \supseteq \ker\mathcal{X}$
    $\mathcal{K}_o \leftarrow \mathbb{R}^n$
    for $j = 1$ to $n_s$ do
        $x_b, \mathcal{I} \leftarrow$ Sample($\partial\mathcal{X}$)
        $\mathcal{K}_o \leftarrow \mathcal{K}_o \bigcap \{x \,|\, \nabla g_i^T(x_b)(x - x_b) \le 0, i \in \mathcal{I}\}$
        if ($\mathcal{K}_o = \emptyset$) then
            return $\mathcal{K}_o$
    return $\mathcal{K}_o$
```

2.5.2 Inner Approximation

Consider finding a point $x_k \in \ker\mathcal{X}$ that maximizes a linear cost $c^T x_k$ where $c \in S^{n-1}$ (i.e. the support function of $\ker\mathcal{X}$). From Lemma 2.5, the resulting convex optimization problem requires set containment constraints:

$$\begin{aligned} \max_{x_k} \quad & c^T x_k \\ \text{s.t.} \quad & \\ & -\nabla g_i(x)^T (x_k - x) \ge 0 \,\forall\, x \in \partial\mathcal{X}_i, i \in [m]. \end{aligned} \tag{2.14}$$

We replace the set containment conditions with SOS conditions using Putinar's Positivstellensatz [20].

Optimization Problem: FindSupport(c, g_i)

$$\begin{aligned} \max_{x_k, \lambda_j^{(i)}(x)} \quad & c^T x_k \\ \text{s.t.} \quad & \\ & -\nabla g_i(x)^T (x_k - x) - \sum_{j=1}^{m} \lambda_j^{(i)}(x)(1 - g_j(x)) \in \Sigma[x], i \in [m] \\ & \lambda_j^{(i)}(x) \in \Sigma[x], \quad i \in [m], j \in [m] \setminus i. \end{aligned} \tag{2.15}$$

For a given direction $c \in S^{n-1}$ this program lower bounds the support function of $\ker\mathcal{X}$. The lower bound monotonically increases with $\deg(\lambda_j^{(i)})$. If the problem is feasible, the maximizing argument x_k belongs to $\ker\mathcal{X}$ and therefore $\mathcal{X}$ is star-convex. If infeasible we cannot make any conclusions about the star-convexity of $\mathcal{X}$. By solving for random directions $c_i \in S^{n-1}, i \in [n_s]$ the convex hull of points x_k provides an inner approximation of the kernel as given by Algorithm 3.

Algorithm 3 Inner Approximation of $\ker\mathcal{X}$

```
Input: $\mathcal{X} = \{x \in \mathbb{R}^n | g_i(x) \leq 1, i \in [m]\}, \{c_i\} \subset S^{n-1}, i \in [n_s]$
Output: Inner Approximation $\mathcal{K}_i \subseteq \ker\mathcal{X}$
    $\mathcal{K}_i \leftarrow \emptyset$
    for $j = 1$ to $n_s$ do
        $x_k \leftarrow$ FindSupport$(c_j, g_i)$
        if FindSupport$(c_j, g_i)$ = Infeasible then
            return $\mathcal{K}_i = \emptyset$
        $\mathcal{K}_i \leftarrow$ conv$(\mathcal{K}_i, x_k)$
    return $\mathcal{K}_i$
```

2.5.3 Kernel of Unions and Intersections

Given sets $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^n$ and their kernels, we can find inner approximations of the kernel of their intersection and union using the following lemma.

Lemma 2.6. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^n$. Then the following holds:*

$$\ker(\mathcal{A} \cap \mathcal{B}) \supseteq \ker\mathcal{A} \cap \ker\mathcal{B} \tag{2.16}$$

$$\ker(\mathcal{A} \cup \mathcal{B}) \supseteq \ker\mathcal{A} \cap \ker\mathcal{B}. \tag{2.17}$$

Thus if $\mathcal{A}, \mathcal{B}$ are star-convex and have kernels that intersect, their union and intersection is also star-convex. This is useful for establishing star-convexity without resorting to numerical algorithms.

2.6 Examples

We evaluate Algorithm 1 on various examples and compare the results to the existing heuristics reviewed in Section III.[5] We focus our comparison on outer approximations as more heuristics apply to this case. We use percent error as our metric, calculated as $100 \times \frac{\text{vol}\mathcal{F}_o - \text{vol}\mathcal{X}}{\text{vol}\mathcal{X}}$ where $\mathcal{F}_o$ is the outer approximation of $\mathcal{X}$. We first consider approximating two examples from the literature with polynomials of increasing degree. In all instances, our algorithm yielded the tightest outer approximation as shown in Figure 2-3.[6] Next we consider 100 randomly generated convex polytopes in $\mathbb{R}^2$. In the majority of cases, our heuristic yielded the tightest outer approximation as shown in Table 2-I. Lastly, we approximate a set that is not star-convex. Our heuristic degrades with increasing lower bound s_{lb} as suggested by Lemma 2.3.

2.6.1 Polynomial matrix inequality

$$\mathcal{X} = \{x \in \mathbb{R}^2 \mid \begin{bmatrix} 1 - 16x_1x_2 & x_1 \\ x_1 & 1 - x_1^2 - x_2^2 \end{bmatrix} \succeq 0\}.$$

Using Algorithms 2 and 3 we find the kernel

$$\mathcal{K}_o = \mathcal{K}_i = \text{conv}\{\pm(-0.1752, 0.3335), \pm(0.1268, 0.2213)\}$$

as shown in Figure 2-1. Figure 2-2 (left) shows the 4th-order approximation obtained with Algorithm 1. Figure 2-3 shows the percent error as we increase the degree. Although each objective value (not shown) decreases monotonically with increasing degree, the percent error occasionally increases. This demonstrates the heuristic nature of the objectives for minimizing volume.

[5] For the bounding box $\mathcal{B}$ required by the l_1 objective, we used the smallest hyperrectangle $\mathcal{B} \supseteq \mathcal{X}$ unless noted otherwise.

[6] We forego comparing 2nd-order polynomials as the determinant maximization objective exactly minimizes volume in this case.

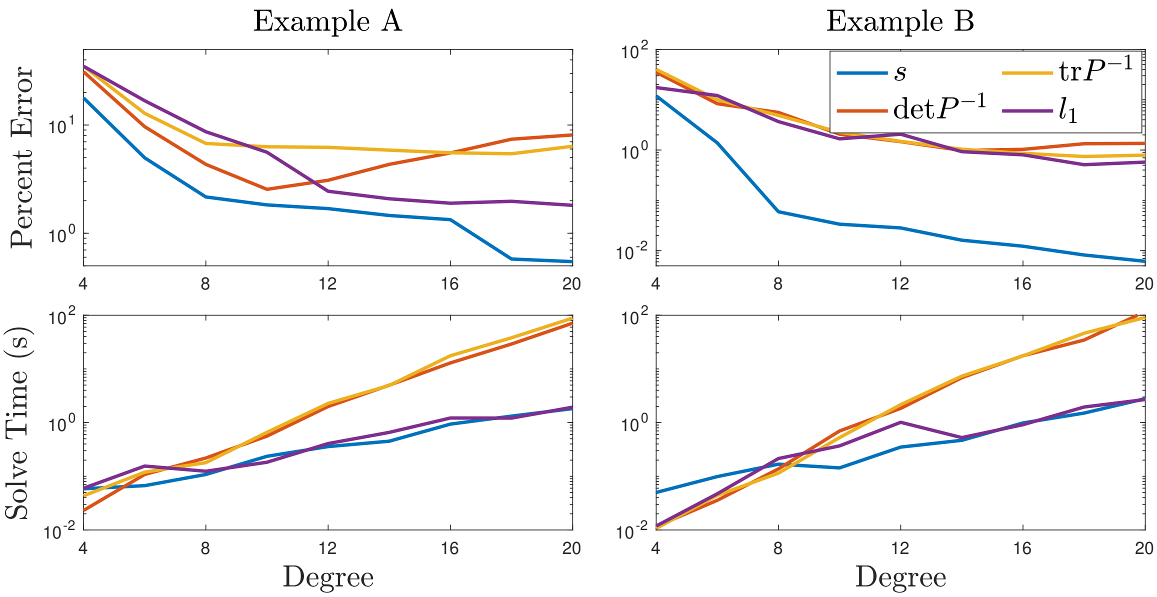


Figure 2-3. Approximation percent error and solve times for examples A and B. Solve times shown for objective s are for one FindApprox(s, g_i) iteration.

2.6.2 Discrete-time stabilizability region

$$\begin{aligned}\mathcal{X} = \{x \in \mathbb{R}^2 \,|\, 1 + 2x_2 \geq 0, 2 - 4x_1 - 3x_2 \geq 0,\\ 10 - 28x_1 - 5x_2 - 24x_1x_2 - 18x_2^2 \geq 0,\\ 1 - x_2 - 8x_1^2 - 2x_1x_2 - x_2^2 - 8x_1^2x_2 - 6x_1x_2^2 \geq 0\}.\end{aligned}$$

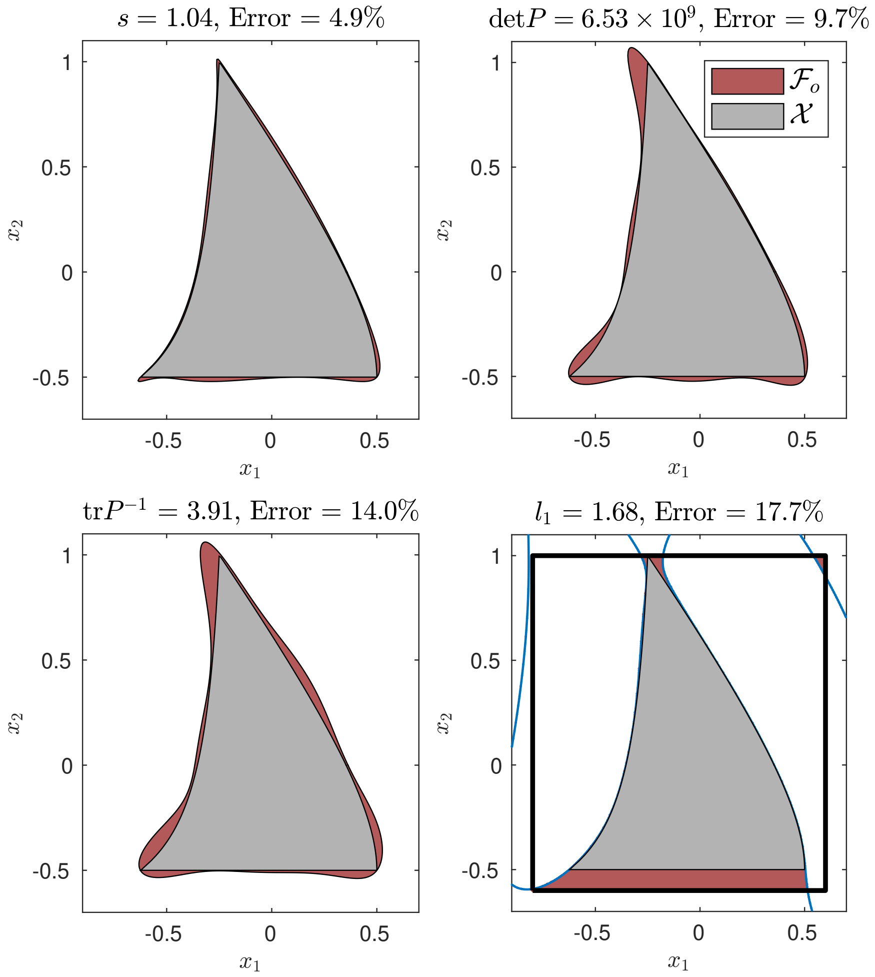
The set contains the origin in its kernel. Figure 2-3 shows the percent error for increasing degree. Figure 2-4 shows the 6th-order approximations obtained with each objective. For the l_1 approximation we also show the bounding box from [7].

2.6.3 Convex Polytopes

We generate 100 random convex polytopes in $\mathbb{R}^2$ with their Chebyshev center at the origin. We find outer approximations using the different objectives. Table 2-I lists the number of times each objective obtained the smallest percent error relative to the other objectives for a given polytope.

Table 2-I. Instances In Which Objective Obtained Smallest Error

Deg.	# Trials	s	$-\det P$	$\mathrm{tr} P^{-1}$	l_1
4	100	73	13	0	14
6	100	98	0	0	2


$s = 1.04$, Error $= 4.9\%$

$\det P = 6.53 \times 10^9$, Error $= 9.7\%$

$\mathrm{tr} P^{-1} = 3.91$, Error $= 14.0\%$

$l_1 = 1.68$, Error $= 17.7\%$

Figure 2-4. 6th-order outer approximations of example B

Table 2-II. Percent Error of Outer Approximations of Example E

r	Degree	$s(s^\star/s_{lb})$	$-\text{det}P$	$\text{tr}P^{-1}$
0.1	4	12.0 (1.096 / 1.025)	13.0	11.8
0.2	4	13.6 (1.104 / 1.104)	16.1	14.0
0.3	4	35.1 (1.250 / 1.250)	18.5	17.8
0.4	4	81.7 (1.492 / 1.492)	17.3	22.9

2.6.4 Non-Star-Convex Set

$$\mathcal{X} = \{x \in \mathbb{R}^2 \,|\, r^2 \leq (x_1 - c)^2 + x_2^2 \leq 1, x_1 \leq c\}.$$

Let $0 < r < c < 1$ so the origin is in the interior of the set. Figure 2-2 shows the set for the case in which $c = 0.9$ and $r = 0.4$. Points $(c, \pm r) \in \partial\mathcal{X}$ yield cutting planes $x_2 \geq r$ and $x_2 \leq -r$ such that $\text{ker}\mathcal{X} = \emptyset$. Table 2-II gives the outer approximation error for $c = 0.9$ and varying r.[7] For the scaling objective, we also report the objective value $s^\star$ and its lower bound s_{lb}.[8] As s_{lb} increases the percent error increases, confirming our heuristic is best suited to star-convex sets.

2.6.5 Solver Performance

Figure 2-3 shows the solve times for the various objectives on a logarithmic scale. Applied to a matrix $P \in \mathbb{R}^{m\times m}$, the $-\text{det}P$ and $\text{tr}P^{-1}$ objectives introduce a PSD matrix $H \in \mathbb{R}^{2m\times 2m}$ due to reformulations involving the exponential cone [21] and Schur complement [10] respectively. In contrast, the scaling (s) and l_1 objectives work directly with P, yielding smaller semidefinite programs. The l_1 objective has the best computational performance. Due to the use of bisection, the total solve time for the scaling objective is an integer multiple of the time shown in Figure 2-3. Accounting for this, the scaling objective still remains competitive with the $-\text{det}P$ and $\text{tr}P^{-1}$

[7] The l_1 objective failed to improve upon the bounding box $\mathcal{B}$ supplied.

[8] The line segments connecting $(0, 0)$ to $(c, \pm r)$ define the maximum lower bound on s in Lemma 2.3. It can be shown that $s_{lb} = \frac{\|p_2\|}{\|p_1\|}$ where $p_2 = (c, r), p_1 = (c + r\cos\phi, r\sin\phi)$ and $\phi = \frac{\pi}{2} + 2\arctan\frac{r}{c}$.

objectives.

2.6.6 Implementation Details

YALMIP [22] and MOSEK [21] were used to solve the SOS programs. Volumes of non-star-convex sets were approximated by evaluating the indicator function over a discrete grid. Volumes of star-convex sets were approximated using numerical integration in polar coordinates.

2.7 Conclusions

An algorithm for finding approximations of semialgebraic sets using sum-of-squares optimization was proposed. The algorithm relies on a novel objective which minimizes the scaling necessary to transform an inner approximation into an outer approximation of the set. Numerical examples demonstrated this objective often finds tighter approximations compared to existing heuristics when applied to star-convex sets. Applied to non-star-convex sets, our proposed heuristic performs poorly. A promising direction to address this is through star-convex decompositions [23]. We leave this exploring this option for future work.

Chapter 3

Outer Approximations of Minkowski Operations on Complex Sets via Sum-of-Squares Optimization

3.1 Introduction

Set operations on complex sets naturally arise in many control applications [1, 24]. The most prominent is robustness analysis in which the Nyquist criterion is used to assess the stability of a control system. Given a plant $P(s)$ and associated controller $C(s)$, the Nyquist stability criterion involves plotting their product as s travels along a contour of the right half plane [25]. If both plant and controller are known exactly, the numerical evaluation of this criterion at a given s involves a simple product of two points in the complex plane. Uncertainty in the plant and controller leads to these points becoming sets in the complex plane. Evaluation of the stability criterion then involves determining all possible complex products of points drawn from the two sets. Beyond multiplication, forming parallel or feedback connections of uncertain transfer functions leads to addition and division operations applied to sets. Following [26], we refer to these various operations on complex sets as Minkowski operations.

Minkowski operations on complex sets are relevant to other domains including computer-aided design [27] and geometric optics [2]. More recently, the authors of [3] use Minkowski products in analyzing the convergence of optimization algorithms. The authors introduce the *Scaled Relative Graph* which visualizes nonlinear operators as sets in the complex plane. Composition of these operators then involves computing Minkowski products. This can be used to provide formal proofs of convergence with geometric arguments.

Closed-form expressions of the sets resulting from Minkowski operations are not known except for cases involving relatively simple sets. The most widely studied case involves discs in the complex plane which are parameterized by their center and radius. This is sometimes referred to as complex circular arithmetic [28]. The results of [1–3] are limited to operations involving such disks.

When exact closed-form expressions are not attainable, one may instead seek to find an outer approximation. If done through manual derivation, this quickly becomes a time-intensive process which requires dedicated efforts for each class of sets considered. For example, in [1], the authors develop an outer approximation for the sum of two complex discs.

As an alternative to manual derivation, an optimization-based approach offers the promise of automating this process. A recent body of literature demonstrates the versatility of sum-of-squares (SOS) optimization for approximating semi-algebraic sets with polynomial functions. Applications include encapsulating 3D point clouds [10], bounding regions of stability for PID controllers [7], and representing unions of sets with a single polynomial [14]. The main contribution of this chapter is a method for finding outer approximations of Minkowski operations of addition, multiplication, and division of an arbitrary number of complex sets that belong to a fairly general class.

The rest of the chapter is organized as follows. Section II defines the sets and Minkowski operations considered and reviews the generalized $\mathcal{S}$-procedure for SOS optimization. Section III sets up the problem and develops SOS-based optimization problems for finding outer approximations to the Minkowski operations. Section IV provides examples of the resulting outer approximations. Section V concludes the chapter and discusses future directions.

3.1.1 Notation

Let $\mathbf{r} = x + iy$ be a complex number with magnitude $r = \sqrt{x^2 + y^2}$ and angle $\theta = \arctan(y/x)$. For $\xi \in \mathbb{R}^n, \mathbb{R}[\xi]$ is the set of polynomials in ξ with real coefficients. The subset $\sum[\xi] = \{p = p_1^2 + p_2^2 + \ldots + p_n^2 : p_1, \ldots, p_n \in \mathbb{R}[\xi]\}$ of $\mathbb{R}[\xi]$ is the set of SOS polynomials in ξ. $\mathbb{Z}\,(\mathbb{Z}_+)$ is the set of non-negative (positive) integers. For convenience, we define the following sets of indices

$$\mathcal{H} = \{0, 1, \ldots, m\}\,,$$
$$\mathcal{J} = \{1, 2, \ldots, n\}\,,$$
$$\mathcal{K} = \{n+1, n+2, \ldots, m\}\,.$$

We use $x_{[j]}$ to denote element j of vector $x \in \mathbb{R}^n$. Similarly we use $x_{[j:k]}$ to denote the vector $[x_{[j]}\ x_{[j+1]}\ \ldots\ x_{[k]}]^T$ Instead of $\sum_{j=1}^{n} x_{[j]}$, we use $\sum_{j} x_{[j]}$, when the dimension n is implicit from the context.

3.2 Preliminaries

3.2.1 Representation of Complex Sets

Let $\mathcal{R}$ denote the set of points in the complex plane between two polar contours, $r^l(\theta)e^{i\theta}$ and $r^u(\theta)e^{i\theta}$, evaluated over the angle range $\theta \in [\theta^l,\ \theta^u]$, i.e.,

$$\mathcal{R} = \{re^{i\theta} | 0 \leq r^l(\theta) \leq r \leq r^u(\theta), \theta^l \leq \theta \leq \theta^u\}\,. \tag{3.1}$$

Throughout we use the superscripts l and u to denote lower and upper bounds. We use subscripts where appropriate to distinguish between different sets of this form. Figure 3-1 provides an example of this notation for the following set:

$$\mathcal{R} = \{re^{i\theta} | 1 + \frac{1}{4}\sin\theta \leq r \leq \frac{3}{2} - \frac{1}{4}\cos\theta, 0 \leq \theta \leq \frac{\pi}{3}\}\,. \tag{3.2}$$

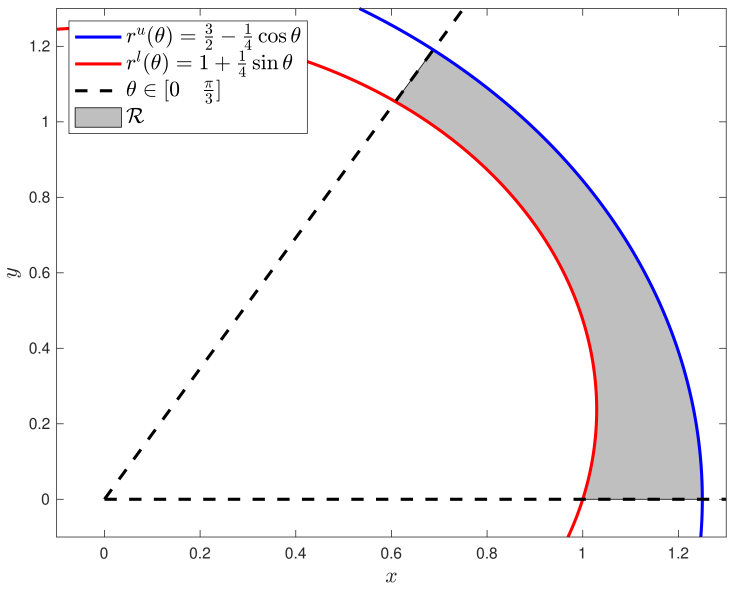


Figure 3-1. Complex set of the form (3.1).

Remark 3.1. Our focus on sets of the form (3.1) is motivated by applications in robust control where there is uncertainty about the gain (r) and phase (θ) of a transfer function at a given frequency. Lacking additional insight, a common assumption is that these variations in gain and phase are independent and are described by simple interval bounds [25]. In (3.1) this corresponds to constant values for r^l and r^u. Our setting is more flexible in that it allows the gain variation to be a function of the phase.

3.2.2 Minkowski Operations on Complex Sets

Consider a family of n sets of the form (3.1) and let $\mathcal{S}_{\bigotimes}$ denote the set obtained by forming all possible complex products. Following [2] we refer to this as the Minkowski product

$$\mathcal{S}_{\bigotimes} = \{ \prod_{j \in \mathcal{J}} \mathbf{r}_j | \mathbf{r}_j \in \mathcal{R}_j, j \in \mathcal{J} \}\,. \tag{3.3}$$

Similarly, we define Minkowski division as the set obtained by forming all possible pair-wise complex divisions between two sets:

$$\mathcal{S}_{\div} = \{\mathbf{r}_1\mathbf{r}_2^{-1} | \mathbf{r}_1 \in \mathcal{R}_1, \mathbf{r}_2 \in \mathcal{R}_2\}. \tag{3.4}$$

The Minkowski sum is defined as follows:

$$\mathcal{S}_{\bigoplus} = \{\sum_{j \in \mathcal{J}} \mathbf{r}_j | \mathbf{r}_j \in \mathcal{R}_j, j \in \mathcal{J}\} \tag{3.5}$$

In this work we focus on two operations that often arise in control applications. The first operation contains multiplication and division as special cases:

$$\mathcal{S}_{\frac{\bigotimes}{\bigotimes}} = \{\prod_{j \in \mathcal{J}} \mathbf{r}_j \prod_{k \in \mathcal{K}} \mathbf{r}_k^{-1}, \mathbf{r}_j \in \mathcal{R}_j, \mathbf{r}_k \in \mathcal{R}_k, j \in \mathcal{J},\, k \in \mathcal{K}\} \tag{3.6}$$

The second operation extends the Minkowski sum to allow inversion of some sets.

$$\mathcal{S}_{\bigoplus + \bigoplus^{-1}} = \{\sum_{j \in \mathcal{J}} \mathbf{r}_j + \sum_{k \in \mathcal{K}} \mathbf{r}_k^{-1} | \mathbf{r}_j \in \mathcal{R}_j, \mathbf{r}_k \in \mathcal{R}_k, j \in \mathcal{J},\, k \in \mathcal{K}\} \tag{3.7}$$

3.2.3 Generalized S-Procedure and SOS Optimization

In the development that follows, we will be interested in solving optimization problems of the following form:

$$\begin{aligned} \min_{\alpha^h} \; & \sum_{h=1}^{j} c_h^T \alpha^h \\ \text{s.t. } & g_1(\xi_1, \alpha^1) d_1(\xi_1) - f_1(\xi_1) \geq 0 \quad \forall\, \xi_1 \in \mathcal{X}_1 \\ & g_2(\xi_2, \alpha^2) d_2(\xi_2) - f_2(\xi_2) \geq 0 \quad \forall\, \xi_2 \in \mathcal{X}_2 \\ & \qquad \vdots \\ & g_j(\xi_j, \alpha^j) d_j(\xi_j) - f_j(\xi_j) \geq 0 \quad \forall\, \xi_j \in \mathcal{X}_j \end{aligned} \tag{3.8}$$

where

$$\mathcal{X}_h = \{\xi_h | h_{h,k}(\xi_h) \geq 0, k = 1, \ldots, n_h\}. \tag{3.9}$$

In each constraint, $\xi_j \in \mathbb{R}^{n_j}$ is a vector of free variables and $g_j(\xi_j, \alpha^j), d_j(\xi_j), f_j(\xi_j)$, $h_{j,k}(\xi_j) \in \mathbb{R}[\xi_j]$ are polynomials of these variables. The coefficients α^j of $g_j(\xi_j, \alpha^j)$ are explicitly listed to highlight that they are decision variables. The objective is linear with each c_j being a given weighting of the decision variable vector α^j. The constraints consist of non-negativity conditions that must hold for all ξ_j in the semi-algebraic set $\mathcal{X}_j$ which is described by polynomial inequalities of ξ_j. This is a set-containment condition.

The generalized $\mathcal{S}$-procedure provides a sufficient condition for the set-containment to hold [16]. For each polynomial inequality $h_{j,k}(\xi_j)$ describing the set $\mathcal{X}_j$, we introduce a non-negative polynomial $s_{j,k}(\xi_j, \beta^{j,k})$ with coefficients $\beta^{j,k}$ as decision variables. We can then remove the set-containment conditions and solve the following problem.

$$
\begin{aligned}
\min_{\alpha^h, \beta^{j,k}} \; & \sum_{h=1}^{j} c_h^T \alpha^h \\
\text{s.t. } & g_1(\xi_1, \alpha^1) d_1(\xi_1) - f_1(\xi_1) \\
& \qquad - \sum_k s_{1,k}(\xi_1, \beta^{1,k}) h_{1,k}(\xi_1) \geq 0 \quad \forall\, \xi_1 \in \mathbb{R}^{n_1} \\
& \qquad s_{1,k}(\xi_1, \beta^{1,k}) \geq 0 \quad \forall\; \xi_1 \in \mathbb{R}^{n_1}, k \in 1, \ldots, n_1 \\
& g_2(\xi_2, \alpha^2) d_2(\xi_2) - f_2(\xi_2) \\
& \qquad - \sum_k s_{2,k}(\xi_2, \beta^{2,k}) h_{2,k}(\xi_2) \geq 0 \quad \forall\, \xi_2 \in \mathbb{R}^{n_2} \\
& \qquad s_{2,k}(\xi_2, \beta^{2,k}) \geq 0 \quad \forall\; \xi_2 \in \mathbb{R}^{n_2}, k \in 1, \ldots, n_2 \\
& \qquad \vdots \\
& g_j(\xi_j, \alpha^j) d_j(\xi_j) - f_j(\xi_j) \\
& \qquad - \sum_k s_{j,k}(\xi_j, \beta^{j,k}) h_{j,k}(\xi_j) \geq 0 \quad \forall\, \xi_j \in \mathbb{R}^{n_j} \\
& \qquad s_{j,k}(\xi_j, \beta^{j,k}) \geq 0 \quad \forall\; \xi_j \in \mathbb{R}^{n_j}, k \in 1, \ldots, n_j
\end{aligned}
\tag{3.10}
$$

The left hand side of each inequality j describes a polynomial of free variables ξ_j with decision variables α^j and $\beta^{j,k}$ entering linearly. We can replace each non-negativity constraint with the more restrictive condition that the polynomial be a SOS polynomial. The resulting optimization problem can then be written as a semidefinite program and solved.

Although we only show inequality constraints above, any equality constraint

$h(\xi) = 0$ can be represented by two constraints $h(\xi) \geq 0, h(\xi) \leq 0$. In the development that follows we focus on transforming problems of interest into the form of (3.8). Once in this form, the subsequent application of the $\mathcal{S}$-procedure and SOS conditions is straight-forward. Due to page limits we do not explicitly include this step.

3.3 Main Results

We now develop a method for finding outer approximations of sets arising from the Minkowski operations defined in Section 3.2.2. Through a series of variable transformations we pose this as a polynomial optimization problem with set-containment constraints. The generalized $\mathcal{S}$-procedure outlined in Section 3.2.3 is then applied to obtain a convex optimization problem which is readily solved.

3.3.1 Problem Setup

In general, closed-form expressions do not exist for the sets $\mathcal{S}_\bullet$ resulting from the Minkowski operation denoted by $\bullet$. Here we focus on finding a set $\mathcal{R}_\bullet$ of the form (3.1) that provides an outer approximation of $\mathcal{S}_\bullet$. A natural objective is to minimize the area of $\mathcal{R}_\bullet$ subject to the set-containment condition $\mathcal{S}_\bullet \subseteq \mathcal{R}_\bullet$. This can be posed as an optimization problem:

$$\begin{aligned} \min_{r^l(\theta), r^u(\theta)} \quad & \int_{\theta^l}^{\theta^u} r^u(\theta) - r^l(\theta) d\theta \\ \text{s.t.} \quad & \mathcal{S}_\bullet \subseteq \mathcal{R}_\bullet \end{aligned} \tag{3.11}$$

where

$$\mathcal{R}_\bullet = \{re^{i\theta} | 0 \leq r^l(\theta) \leq r \leq r^u(\theta), \theta^l \leq \theta \leq \theta^u\}. \tag{3.12}$$

We wish to transform (3.11) into a polynomial optimization problem which we can solve. To do so, we must choose a basis for the functions $r^l(\theta)$ and $r^u(\theta)$ which form our outer approximation $\mathcal{R}_\bullet$. We additionally assume the sets being operated on are represented by contours which share this chosen basis.

Assumption 3.1. *We assume that each contour $r(\theta, \alpha)$ is a function of $\cos\theta$ and $\sin\theta$ with associated real coefficient vector α, i.e.,*

$$\begin{aligned} r(\theta, \alpha) &= \alpha_{[1]} + \alpha_{[2]} \cos\theta + \alpha_{[3]} \sin\theta + \alpha_{[4]} (\cos\theta)^2 + \dots \\ &= \sum_j \alpha_{[j]} (\cos\theta)^{u_j} (\sin\theta)^{v_j} , \quad \alpha_{[j]} \in \mathbb{R}, u_j, v_j \in \mathbb{N}. \end{aligned}$$

We will sometimes refer to this parameterization as a polynomial of $\cos\theta$ and $\sin\theta$, as introducing independent variables for each would yield a polynomial expression. This parameterization readily admits an upper bound which we will utilize.

Lemma 3.1. *Let $r(\theta, \alpha)$ be a polynomial function of $\cos\theta$ and $\sin\theta$ with associated real coefficient vector α. The following inequality holds:*

$$r(\theta, \alpha) \leq \bar{r} \tag{3.13}$$

where:

$$\bar{r} = \sum_j \left| \alpha_{[j]} \right| \tag{3.14}$$

Proof. Note the following inequality:

$$\left| \alpha_{[j]} (\cos\theta)^m (\sin\theta)^n \right| \leq \left| \alpha_{[j]} \right| \quad \forall\, \theta \in \mathbb{R}, m, n \in \mathbb{N} \tag{3.15}$$

The inequality for the polynomial follows immediately. □

Assumption 3.2. *We assume that any set which is inverted has a known, positive lower bound for $r^l(\theta)$ which we denote $\underline{r}^l$.*

$$r^l(\theta) \geq \underline{r}^l > 0 \quad \forall\, \theta^l \leq \theta \leq \theta^u \tag{3.16}$$

Assumption 3.2 ensures the set does not contain the origin and therefore its inverse is bounded. The sets resulting from the introduced Minkowski operations are then bounded as well. This is important as seeking an outer approximation of an unbounded set would be trivially infeasible. Knowledge of the constant $\underline{r}^l$ allows us to calculate an upper bound as given by the following lemma.

Lemma 3.2. *Let* $\mathbf{r}$ *be a point in* $\mathcal{S}_{\bigoplus + \bigoplus^{-1}}$ *as defined by (3.7). Let Assumptions 3.1 and 3.2 hold. Then the following inequality holds:*

$$|\mathbf{r}| \leq \sum_{j \in \mathcal{J}} (\bar{r}_j^u) + \sum_{k \in \mathcal{K}} (\underline{r}_k^l)^{-1}, \quad \forall \, \mathbf{r} \in \mathcal{S}_{\bigoplus + \bigoplus^{-1}}. \tag{3.17}$$

Proof. Given that $\mathbf{r} \in \mathcal{S}_{\bigoplus + \bigoplus^{-1}}$, there exists points $\mathbf{r}_j \in \mathcal{R}_j, \mathbf{r}_k \in \mathcal{R}_k, j \in \mathcal{J}$, $k \in \mathcal{K}$ such that the following equality holds:

$$\begin{aligned} |\mathbf{r}| &= \left| \sum_{j \in \mathcal{J}} \mathbf{r}_j + \sum_{k \in \mathcal{K}} \mathbf{r}_k^{-1} \right| \\ &\leq \sum_{j \in \mathcal{J}} \left| \mathbf{r}_j \right| + \sum_{k \in \mathcal{K}} \left| \mathbf{r}_k^{-1} \right| \\ &\leq \sum_{j \in \mathcal{J}} (\bar{r}_j^u) + \sum_{k \in \mathcal{K}} (\underline{r}_k^l)^{-1} \quad \text{Lem. 1, Asm. 2} \end{aligned} \tag{3.18}$$

□

Assumption 3.3. *Let* Θ *denote the set of angles in* $\mathcal{S}_\bullet$*:*

$$\Theta = \{\arctan(\mathbf{r}) | \mathbf{r} \in \mathcal{S}_\bullet\} \tag{3.19}$$

We assume that we know Θ *exactly so that we can specify the lower and upper bounds* θ^l, θ^u *in our objective.*

The range of possible angles is easy to calculate for the product and division of complex sets as angles simply add and subtract. For Minkowski sums of complex sets the set of possible angles is not easily calculated. We discuss methods for doing so in section 3.3.4.

3.3.2 Minkowski Product and Division of Complex Sets

We seek to minimize the area of an outer approximation of $\mathcal{S}_{\frac{\otimes}{\otimes}}$. This can be posed as follows:

$$\begin{aligned} &\min_{\alpha^u,\alpha^l} \int_{\theta^l}^{\theta^u} r^u(\theta,\alpha^u) - r^l(\theta,\alpha^l) d\theta \\ &\text{s.t. } r^l(\theta_0,\alpha^l) \leq \left| \frac{\prod_{j\in\mathcal{J}} r_j e^{i\theta_j}}{\prod_{k\in\mathcal{K}} r_k e^{i\theta_k}} \right| \leq r^u(\theta_0,\alpha^u) \\ &\quad \forall(\theta_{[0:m]}, r_{[1:m]}) \in \mathcal{X} \end{aligned} \tag{3.20}$$

where $\mathcal{X}$ is the semi-algebraic set:

$$\begin{aligned} \mathcal{X} =& \{(\theta_{[0:m]}, r_{[1:m]}) \ : \theta_0 = \sum_{j\in\mathcal{J}} \theta_j - \sum_{k\in\mathcal{K}} \theta_k, \\ & r^l_j(\theta_j) \leq r_j \leq r^u_j(\theta_j), \theta^l_j \leq \theta_j \leq \theta^u_j, j \in \mathcal{J} \\ & r^l_k(\theta_k) \leq r_k \leq r^u_k(\theta_k), \theta^l_k \leq \theta_k \leq \theta^u_k, k \in \mathcal{K}\} \end{aligned} \tag{3.21}$$

Given that we know the bounds θ_l, θ_u, we can evaluate the integral within our objective to eliminate the dependency on θ. This yields a linear objective in terms of the coefficients.

$$\int_{\theta^l}^{\theta^u} r^u(\theta,\alpha^u) - r^l(\theta,\alpha^l) d\theta = c_l^T\alpha^l + c_u^T\alpha^u$$

We introduce intermediate variables ϕ_j such that the sum of angles defining θ_0 can be written as the sum of two angles.

$$\phi_j = \sum_{h=j}^{m} c_h\theta_h \tag{3.22}$$

where

$$c_h = \begin{cases} +1, & \text{if } h \in \mathcal{J} \\ -1, & \text{if } h \in \mathcal{K} \end{cases} \tag{3.23}$$

The angle summation can then be replaced with the following semi-algebraic set:

$$\begin{aligned} \mathcal{Z} = \{(\theta_{[0:m]}, \phi_{[2:m-1]}) | \theta_0 &= c_1\theta_1 + \phi_2, \\ \phi_2 &= c_2\theta_2 + \phi_3, \\ &\dots \\ \phi_{m-2} &= c_{m-2}\theta_{m-2} + \phi_{m-1}, \\ \phi_{m-1} &= c_{m-1}\theta_{m-1} + c_m\theta_m\} \end{aligned} \tag{3.24}$$

We then obtain a superset of $\mathcal{Z}$ by replacing each equality constraint with two constraints involving cos and sin.

$$\begin{aligned} \mathcal{Y} = \{(\theta_{[0:m]}, \phi_{[2:m-1]}) | & \\ \cos\theta_0 &= \cos(c_1\theta_1 + \phi_2)\,, \\ \sin\theta_0 &= \sin(c_1\theta_1 + \phi_2)\,, \\ \cos\phi_2 &= \cos(c_2\theta_2 + \phi_3)\,, \\ \sin\phi_2 &= \sin(c_2\theta_2 + \phi_3)\,, \\ &\dots \\ \cos\phi_{m-2} &= \cos(c_{m-2}\theta_{m-2} + \phi_{m-1})\,, \\ \sin\phi_{m-2} &= \sin(c_{m-2}\theta_{m-2} + \phi_{m-1})\,, \\ \cos\phi_{m-1} &= \cos(c_{m-1}\theta_{m-1} + c_m\theta_m)\,, \\ \sin\phi_{m-1} &= \sin(c_{m-1}\theta_{m-1} + c_m\theta_m)\} \end{aligned} \tag{3.25}$$

Remark 3.2. $\mathcal{Y}$ is a superset of $\mathcal{Z}$ as the trigonometric identities still hold when angles have multiples of 2π added. Given we are working with periodic functions (Assumption 3.1) this is a subtlety of no consequence.

Recall the following trigonometric identities involving angles a and b with signs $c_a, c_b \in \{-1, 1\}$:

$$\cos(c_a a + c_b b) = \cos a \cos b - c_a c_b \sin a \sin b\,, \tag{3.26}$$

$$\sin(c_a a + c_b b) = c_a \sin a \cos b + c_b \cos a \sin b\,. \tag{3.27}$$

Applying these identities we can write the constraints defining $\mathcal{Y}$ in terms of $\cos\theta_h, \sin\theta_h$, $\cos\phi_l, \sin\phi_l$. We then eliminate the trigonometric terms by introducing new variables along with a quadratic equality constraint.

$$\begin{aligned} z_{c\theta_h} = \cos\theta_h, z_{s\theta_h} = \sin\theta_h, z_{c\theta_h}^2 + z_{s\theta_h}^2 = 1 \\ \forall\, h \in 0 \cup \mathcal{J} \cup \mathcal{K} \\ z_{c\phi_l} = \cos\phi_l, z_{s\phi_l} = \sin\phi_l, z_{c\phi_l}^2 + z_{s\phi_l}^2 = 1 \\ \forall\, l = 2, \ldots, m-1 \end{aligned}$$

Next, we rewrite the angle constraints $\theta_h^l \le \theta_h \le \theta_h^u$ in terms of $z_{c\theta_h}, z_{s\theta_h}$. In the new variables, the points satisfying the angle interval constraint can be represented by the intersection of the quadratic equality constraint and a halfplane that passes through the points $(\cos\theta_h^l, \sin\theta_h^l)$ and $(\cos\theta_h^u, \sin\theta_h^u)$. Figure 3-2 visualizes this for $\theta^l = 0, \theta^u = \frac{\pi}{3}$. Defining the midpoint angle $\theta_h^m = \frac{1}{2}(\theta_h^l + \theta_h^u)$, it can be shown that the halfplane is the set of points $(z_{c\theta_h}, z_{s\theta_h})$ satisfying:

$$a_h^c z_{c\theta_h} + a_h^s z_{s\theta_h} \ge b_h \tag{3.28}$$

where

$$a_h^c = \cos\theta_h^m, a_h^s = \sin\theta_h^m, b_h = \cos\theta_h^m \cos\theta_h^u + \sin\theta_h^m \sin\theta_h^m \tag{3.29}$$

With this change of variables, the optimization problem is rewritten as follows:

$$\begin{aligned} & \min_{\alpha^l, \alpha^u} c_l^T \alpha^l + c_u^T \alpha^u \\ & \text{s.t.} \qquad r^l(z_{c\theta_0}, z_{s\theta_0}, \alpha^l) \prod_{k \in \mathcal{K}} r_k \le \prod_{j \in \mathcal{J}} r_j \\ & \qquad\qquad r^u(z_{c\theta_0}, z_{s\theta_0}, \alpha^u) \prod_{k \in \mathcal{K}} r_k \ge \prod_{j \in \mathcal{J}} r_j \\ & \forall (z_{c\theta_{[0:m]}}, z_{s\theta_{[0:m]}}, z_{c\phi_{[2:m-1]}}, z_{s\phi_{[2:m-1]}}, r_{[1:m]}) \in \mathcal{W} \end{aligned} \tag{3.30}$$

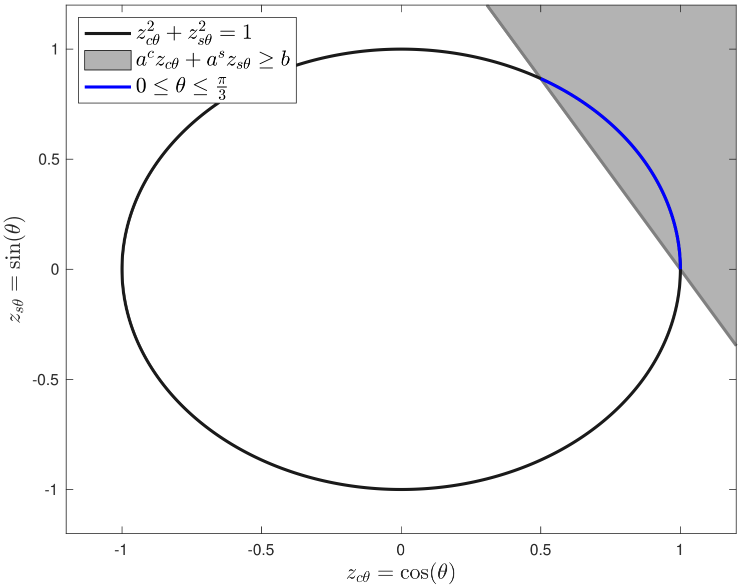


Figure 3-2. Constraints for angle interval.

where:

$$
\begin{aligned}
\mathcal{W} = \{\big(z_{c\theta_{[0:m]}}, z_{s\theta_{[0:m]}}, z_{c\phi_{[2:m-1]}}, z_{s\phi_{[2:m-1]}}, r_{[1:m]}\big) :& \\
z_{c\theta_0} = z_{c\theta_1} z_{c\phi_2} - c_1 z_{s\theta_1} z_{s\phi_2}& \\
z_{s\theta_0} = c_1 z_{s\theta_1} z_{c\phi_2} + z_{c\theta_1} z_{s\phi_2}& \\
z_{c\phi_l} = z_{c\theta_l} z_{c\phi_{l+1}} - c_l z_{s\theta_l} z_{s\phi_{l+1}}, \quad l \in 2, \ldots, m-2& \\
z_{s\phi_l} = c_l z_{s\theta_l} z_{c\phi_{l+1}} + z_{c\theta_l} z_{s\phi_{l+1}}, \quad l \in 2, \ldots, m-2& \\
z_{c\phi_{m-1}} = z_{c\theta_{m-1}} z_{c\theta_m} - c_{m-1} c_m z_{s\theta_{m-1}} z_{s\theta_m}& \\
z_{s\phi_{m-1}} = c_{m-1} z_{s\theta_{m-1}} z_{c\theta_m} + c_m z_{c\theta_{m-1}} z_{s\theta_m}& \\
z_{c\theta_h}^2 + z_{s\theta_h}^2 = 1 \quad h \in 0, \ldots, m& \\
z_{c\phi_l}^2 + z_{s\phi_l}^2 = 1 \quad l \in 2, \ldots, m-1& \\
r_h^l(z_{c\theta_h}, z_{s\theta_h}) \leq r_h \leq r_h^u(z_{c\theta_h}, z_{s\theta_h}) \quad h \in 1, \ldots, m& \\
a_h^c z_{c\theta_h} + a_h^s z_{s\theta_h} \geq b_h \quad h \in 1, \ldots, m\}&
\end{aligned}
\tag{3.31}
$$

This is a polynomial optimization problem with set-containment constraints of the form (3.8). As outlined in section 3.2.3, applying the $\mathcal{S}$-procedure and replacing non-negativity conditions with SOS constraints yields a semidefinite optimization problem which can be solved.

3.3.3 Minkowski Sum of Complex Sets

Calculating the Minkowski sum of complex sets is more involved as we must convert between polar and Euclidean coordinates. In (3.7), points from sets $\mathcal{R}_j, j \in \mathcal{J}$ are directly summed while points from sets $\mathcal{R}_k, k \in \mathcal{K}$ are first inverted and then summed. The resulting Euclidean coordinates (x, y) are given by:

$$x_j = r_j \cos\theta_j, y_j = r_j \sin\theta_j, \quad \forall\, r_j \in \mathcal{R}_j, j \in \mathcal{J},$$
$$x_k = \cos\theta_k / r_k, y_k = -\sin\theta_k / r_k, \quad \forall\, r_k \in \mathcal{R}_k, k \in \mathcal{K}.$$

We sum the Euclidean coordinates to obtain the point $(x_0 + iy_0) \in \mathcal{S}_{\bigoplus + \bigoplus^{-1}}$. We must then determine the angle θ_0 and non-negative radial distance of this point. This is achieved with the following equations:

$$x_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} x_h, \quad y_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} y_h$$
$$x_0 = r_0 \cos\theta_0, \quad y_0 = r_0 \sin\theta_0, \quad r_0 \geq 0$$

The optimization problem is then:

$$\begin{aligned} \min_{\alpha^l, \alpha^u} & \int_{\theta^l}^{\theta^u} r^u(\theta, \alpha^u) - r^l(\theta, \alpha^l) d\theta \\ \text{s.t. } & r^l(\theta_0, \alpha^l) \leq r_0 \leq r^u(\theta_0, \alpha^u)\,, \\ & \quad \forall\, (\theta_{[0:m]}, r_{[0:m]}, x_{[0:m]}, y_{[0:m]}) \in \mathcal{X} \end{aligned} \tag{3.32}$$

where $\mathcal{X}$ is the semi-algebraic set

$$\begin{aligned}
\mathcal{X} = \{(\theta_{[0:m]}, r_{[0:m]}, x_{[0:m]}, y_{[0:m]}) \ &: \\
r_0 \geq 0, r_0 \cos\theta_0 = x_0, r_0 \sin\theta_0 = y_0 \\
x_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} x_h, y_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} y_h \\
r_j \cos\theta_j = x_j, r_j \sin\theta_j = y_j \forall\, j \in \mathcal{J} \\
r_k x_k = \cos\theta_k, r_k y_k = -\sin\theta_k \forall\, k \in \mathcal{K} \\
r_j^l(\theta_j) \leq r_j \leq r_j^u(\theta_j), \theta_j^l \leq \theta_j \leq \theta_j^u, \forall\, j \in \mathcal{J} \\
r_k^l(\theta_k) \leq r_k \leq r_k^u(\theta_k), \theta_k^l \leq \theta_k \leq \theta_k^u, \forall\, k \in \mathcal{K}\}
\end{aligned} \tag{3.33}$$

Following a similar procedure as before, we first integrate the objective to eliminate the dependence on θ. We then introduce new variables for the trigonometric terms:

$$\begin{aligned}
z_{c\theta_h} = \cos\theta_h, z_{s\theta_h} = \sin\theta_h \\
z_{c\theta_h}^2 + z_{s\theta_h}^2 = 1 \qquad \forall\, h \in 0 \cup \mathcal{J} \cup \mathcal{K}
\end{aligned}$$

With this change of variables the optimization problem is rewritten as:

$$\begin{aligned}
\min_{\alpha^l, \alpha^u} \ & c_l^T \alpha^l + c_u^T \alpha^u \\
\text{s.t.} \qquad & r^l(z_{c\theta_0}, z_{s\theta_0}, \alpha^l) \leq r_0 \\
& r^u(z_{c\theta_0}, z_{s\theta_0}, \alpha^u) \geq r_0 \\
\forall\, (z_{c\theta_{[0:m]}}, z_{s\theta_{[0:m]}}, & r_{[0:m]}, x_{[0:m]}, y_{[0:m]}) \in \mathcal{W}
\end{aligned} \tag{3.34}$$

where $\mathcal{W}$ is the semialgebraic set:

$$\begin{aligned}\mathcal{W} = \{(z_{c\theta_{[0:m]}}, z_{s\theta_{[0:m]}}, r_{[0:m]}, x_{[0:m]}, y_{[0:m]}) \ :& \\ r_0 \geq 0, x_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} x_h, \quad y_0 = \sum_{h \in \mathcal{J} \cup \mathcal{K}} y_h& \\ r_0 z_{c\theta_0} = x_0, r_0 z_{s\theta_0} = y_0,& \\ r_j z_{c\theta_j} = x_j, r_j z_{s\theta_j} = y_j, \quad \forall\, j \in \mathcal{J}& \\ r_k x_k = z_{c\theta_k}, r_k y_k = -z_{s\theta_k}, \quad \forall\, k \in \mathcal{K}& \\ z_{c\theta_h}^2 + z_{s\theta_h}^2 = 1 \quad h \in 0, \ldots, m& \\ r_h^l(z_{c\theta_h}, z_{s\theta_h}) \leq r_h \leq r_h^u(z_{c\theta_h}, z_{s\theta_h}) \quad h \in 1, \ldots, m& \\ a_h^c z_{c\theta_h} + a_h^s z_{s\theta_h} \geq b_h \quad h \in 1, \ldots, m\}&\end{aligned} \tag{3.35}$$

As before, applying the $\mathcal{S}$-procedure followed by replacing the non-negativity conditions with SOS constraints yields a semidefinite optimization problem which can be solved.

3.3.4 Determining the Angle Interval

As stated in Assumption 3-2, we assume that we know the exact set of angles Θ contained in the set $\mathcal{S}_\bullet$. For the Minkowski sum this is not readily calculated. Here we outline an iterative approach for conservatively bounding Θ within an interval $\tilde{\Theta} = [\tilde{\theta}^l, \, \tilde{\theta}^u]$ such that $\Theta \subseteq \tilde{\Theta}$.

We initialize our estimate to $\tilde{\Theta} = [0, \, 2\pi]$. If Θ is a strict subset of this interval, then there exists an angle ψ such that $\psi \in \tilde{\Theta} \setminus \Theta$. Along this angle, there is no element of $\mathcal{S}_\bullet$ constraining $r^l(\psi, \alpha^u)$ and $r^u(\psi, \alpha^u)$. Thus our objective which minimizes $r^u(\theta, \alpha^u)$ and maximizes $r^l(\theta, \alpha^l)$ would be unbounded. To resolve this, we add a known upper bound on $r^l(\theta, \alpha^l)$ and a known lower bound on $r^u(\theta, \alpha^u)$. For $r^u(\theta, \alpha^u)$ we use the trivial lower bound of zero. For $r^l(\theta, \alpha^l)$ we make use of the bound provided by Lemma 3.2. To enforce these bounds, we augment problem (3.34) with the following conditions in which θ is replaced by $z_{c\theta_0}, z_{s\theta_0}$:

$$\begin{aligned} r^l(z_{c\theta_0}, z_{s\theta_0}, \alpha^l) &\leq \sum_{j\in\mathcal{J}}(\bar{r}_j^u) + \sum_{k\in\mathcal{K}}(\underline{r}_k^l)^{-1} \,\forall\, (z_{c\theta_0}, z_{s\theta_0}) \in \mathcal{V} \\ r^u(z_{c\theta_0}, z_{s\theta_0}, \alpha^u) &\geq 0 \quad \forall\, (z_{c\theta_0}, z_{s\theta_0}) \in \mathcal{V} \end{aligned} \tag{3.36}$$

where

$$\mathcal{V} = \{(z_{c\theta_0}, z_{s\theta_0}) | z_{c\theta_0}^2 + z_{s\theta_0}^2 = 1\}. \tag{3.37}$$

We solve this augmented problem and then examine the bounding contours $r^l(\theta, \alpha^l)$, $r^u(\theta, \alpha^u)$. For any angles ψ at which the lower bound exceeds the upper bound ($r^l(\psi) > r^u(\psi)$), we can conclude that $\psi \notin \Theta$ and update our angle interval $\tilde{\Theta}$ appropriately. We then repeat this process, solving the augmented problem with the tighter approximation of Θ, examining the resulting bounds to further tighten the interval $\tilde{\Theta}$ and repeating. We stop once the returned bounds satisfy $r^l(\theta) \leq r^u(\theta) \,\forall\, \theta \in \tilde{\Theta}$.

As an aside we note that determining the range of angles in $\mathcal{S}_{\bigoplus + \bigoplus^{-1}}$ can also be solved via global optimization methods using branch-and-bound techniques. Our initial experience with this approach yielded solutions in under a second for the examples considered herein.

3.4 Examples

3.4.1 Minkowski Product Example

Consider the following set formed from Minkowski products and division:

$$\mathcal{S} = \mathcal{R}_1 \bigotimes \mathcal{R}_2 \bigotimes (\mathcal{R}_3 \bigotimes \mathcal{R}_4)^{-1} \tag{3.38}$$

where each set $\mathcal{R}_j$ is as shown in Figure 3-1.

$$\begin{aligned} \mathcal{R}_j = \{re^{i\theta} | 1 + \frac{1}{4}\sin\theta \leq r \leq \frac{3}{2} - \frac{1}{4}\cos\theta, 0 \leq \theta \leq \frac{\pi}{3}\} \\ j = 1, 2, 3, 4 \end{aligned} \tag{3.39}$$

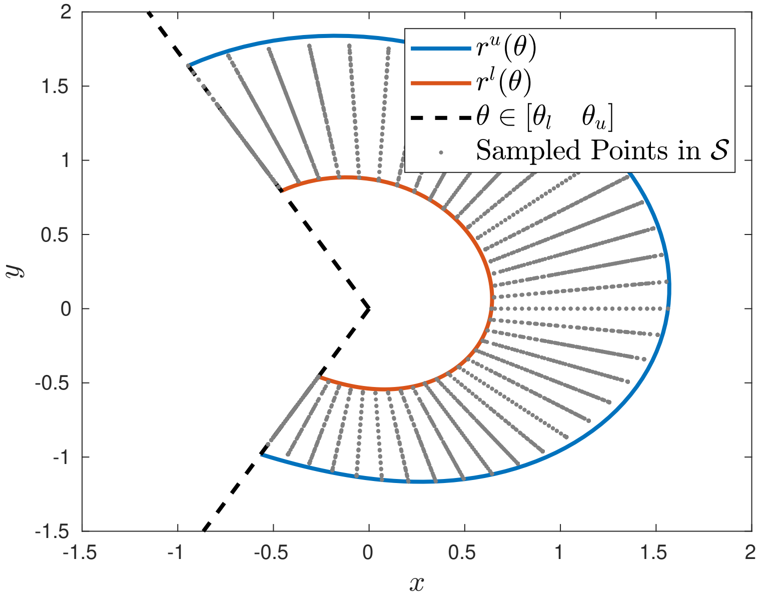


Figure 3-3. Outer bound of Minkowski product (3.38).

By inspection, the possible angles of $\mathcal{S}$ are $\Theta \in [-\frac{2\pi}{3}, \frac{2\pi}{3}]$. Limiting ourselves to 4th-order contours we solve the SOS form of (3.30). Figure 3-3 plots the resulting contour along with points sampled from $\mathcal{S}$. Empirically the outer approximation is close to the true contour suggested by the sampled points.

3.4.2 Minkowski Sum Example

Using the same sets as in the previous example, we now find an outer approximation for the following Minkowski sum

$$\mathcal{S} = \mathcal{R}_1 \oplus \mathcal{R}_2 \oplus (\mathcal{R}_3)^{-1} \oplus (\mathcal{R}_4)^{-1} \tag{3.40}$$

We do not know the possible range Θ of $\mathcal{S}$ so we use the iterative approach previously outlined. For the given set operation, it is straight-forward to obtain an upper bound on r of $2 \times 1.75 + 2 \times (0.75)^{-1} = 6.1667$. We impose the conditions $r^l(\theta) \leq 6.1667$ and $r^u(\theta) \geq 0$. We then solve the SOS form of (3.34) conservatively assuming $\theta^l = 0, \theta^u = 2\pi$ and augmenting the problem with the bounds of (3.36). Figure 3-4 plots the resulting bounds as a function of θ. Examining the plot it is seen that

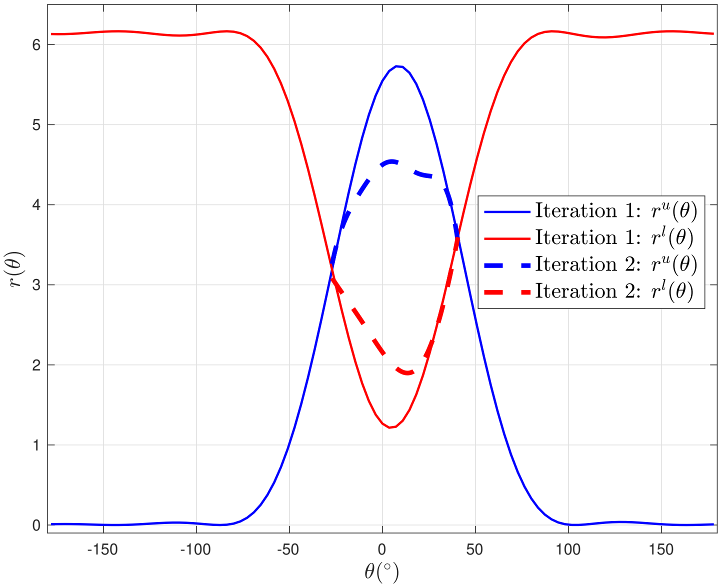


Figure 3-4. Iterative bounds of Minkowski sum.

$r^l(\theta) \leq r^u(\theta)$ for $\theta \in [-27.1^\circ, 40.6^\circ]$. Outside of this interval, $r^l(\theta)$ approaches its upper bound of 6.1667 and $r^u(\theta)$ approaches its lower bound of zero. We again solve the problem now with $\theta^l = -27.1^\circ, \theta^u = 40.6^\circ$ and obtain the dashed lines in Figure 3-4. With the new bounds, $r^l(\theta) \leq r^u(\theta)$ for $\theta \in [-27.1^\circ, 40.4^\circ]$. We again solve the problem with our slightly tightened angle interval. The resulting bounds have $r^l(\theta) \leq r^u(\theta)$ for all $\theta \in [-27.1^\circ, 40.4^\circ]$. At this point we can no longer improve our estimate of Θ so we stop. Figure 3-5 plots the resulting contour along with points sampled from $\mathcal{S}$. Empirically the outer approximation is close to the true contour suggested by the sampled points.

3.4.3 Implementation Details

All examples were solved on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 CPU. The SOS module of YALMIP [22] was used in conjunction with MOSEK [21]. Solving the Minkowski product example took 53 seconds. Solving the Minkowski sum example took 801 seconds for a single iteration (three iterations total).

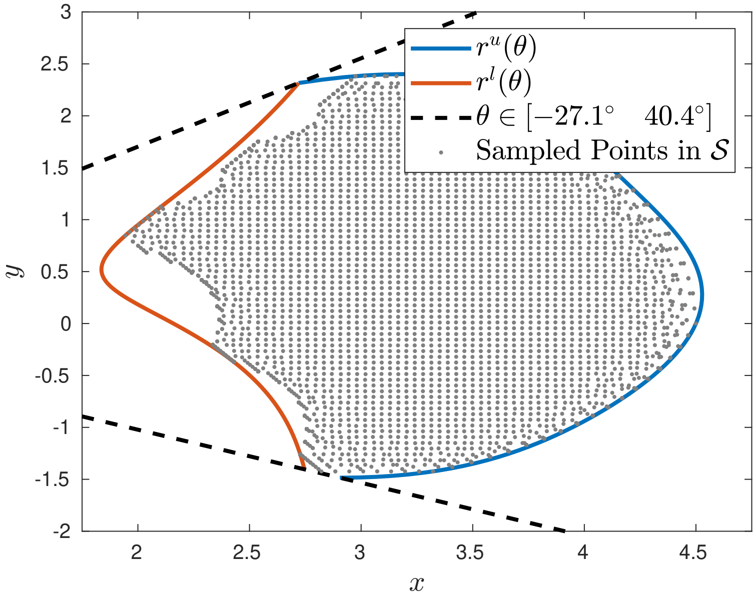


Figure 3-5. Outer bound of Minkowski sum (3.40).

3.4.4 Computational Complexity

Our current implementation utilizes a dense monomial basis for each multiplier $s(\xi, \beta)$. This consists of all possible monomials formed from the free variables ξ up to a given degree (2 in the examples herein). This grows combinatorially as we introduce more sets (and associated free variables). The resulting increase in the semidefinite program size limits scalability. This can be partially improved by using a more informed approach to selecting the underlying basis [29]. More promisingly, our chosen problem formulation provides a natural decomposition method. As each operation (sum, product) returns a set that is of the same form as the input, we can easily decompose a problem consisting of many terms by first forming outer approximations of sub-expressions. We can then solve the full problem with the sub-expressions replaced by their outer approximations.

3.5 Conclusions

In this work we developed optimization-based methods for finding outer approximations of Minkowski sums and products of complex sets. These operations are relevant to problems arising in robust control. Through appropriate variable transformations we posed this as a sum-of-squares optimization problem which is readily solved by off-the-shelf solvers. Examples provided empirical evidence that the resulting approximations are good.

In the future we plan to improve the scalability of our method by considering problem decompositions. Additionally, while our current form assumes the sets are modeled with polar coordinates, we plan to extend our method to supports sets that are more naturally described using Euclidean coordinates. Lastly we plan to use these techniques as building blocks for certifying the robust stability of networked dynamic systems.

Chapter 4

Closed-Form Minkowski Sum Approximations for Efficient Optimization-Based Collision Avoidance

4.1 Introduction

Motion planning is a central task of most autonomous systems, including robots, drones, and autonomous vehicles. Of the many approaches to motion planning, techniques based on nonlinear programming (NLP) such as direct multiple shooting [30] and direct collocation [31] generally offer the most flexibility in regards to choice of objectives and constraints imposed. As high-quality NLP solvers and supporting automatic differentiation tools have become available, it has become feasible to utilize these optimization-based approaches for real-time motion planning or trajectory generation [32].

Despite the flexibility that NLP solvers provide, it can be difficult to efficiently represent obstacle avoidance constraints. Due to their reliance on gradient and Hessian information, most NLP solvers require the objective and constraints to be twice continuously differentiable expressions. This presents a challenge for collision avoidance constraints which often cannot be represented in smooth closed-forms. We

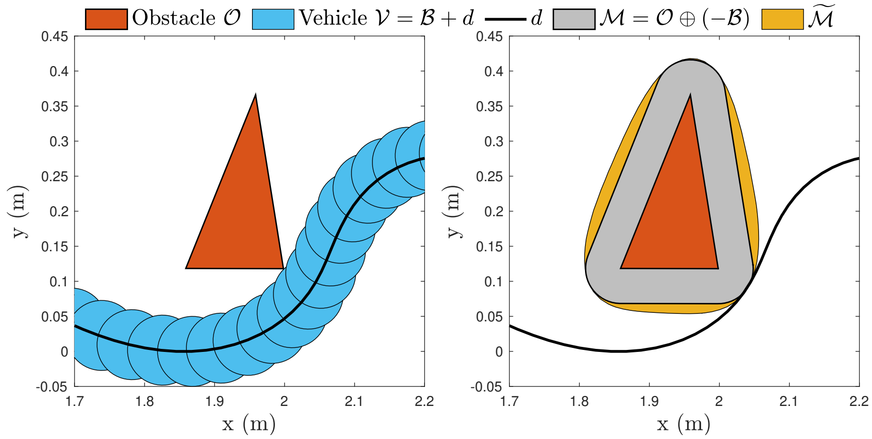


Figure 4-1. Obstacle avoidance in workspace (left) and C-space (right).

briefly review two viable approaches and discuss their advantages and limitations.

Distance Formulation: Collision avoidance can be viewed as ensuring the minimum distance between an obstacle $\mathcal{O}$ and a vehicle $\mathcal{V}$ is greater than zero. In robotics this would be classified as performing collision checking directly in the workspace [33] as shown in the left subplot of Figure 4-1. When the obstacle and the vehicle have convex shapes, the distance between these sets can be computed through convex optimization [34]. Using this formulation as a constraint leads to a bi-level NLP for which we lack reliable solvers. By leveraging strong duality [35], it is possible to reformulate the collision avoidance conditions into expressions amenable to a NLP solver. This is done at the expense of introducing new variables and constraints. In practice, the solver performance can be highly sensitive to the initialization of these variables [36] and the resulting increase in problem complexity can be problematic for real-time motion planning in cluttered environments. Similar remarks hold for collision avoidance reformulations based on Farkas' Lemma [37] or polar set representations [38].

Minkowski Sum Formulation: Collision avoidance can alternatively be viewed through the lens of computational geometry as shown in the right subplot of Figure 4-1. Given the vehicle position $d \in \mathbb{R}^n$, and shapes $\mathcal{B}, \mathcal{O} \subset \mathbb{R}^n$ of vehicle and obstacle, respectively, collision avoidance can be posed as ensuring $d \notin \mathcal{M}$ where

$\mathcal{M} = \mathcal{O} \oplus (-\mathcal{B})$, with $\oplus$ being the Minkowski sum operation [33]. In robotics this is often referred to as the configuration-space (C-space) approach. Incorporating this as a constraint in an NLP solver would require a closed-form, smooth representation of the indicator function of this set. In general, this does not exist as the sets are semialgebraic, involving multiple polynomial (in)equalities. A notable exception is the case of bodies whose boundary surface are smooth and admit both implicit and parametric representations [39], which includes ellipsoids and convex superquadrics [40]. However, many implicit surfaces do not admit a parametric representation and for others obtaining one is an open problem [41]. Additionally, this approach cannot address the practical case of non-smooth boundaries such as convex polytopes.

4.1.1 Contributions

In this work we propose efficient collision avoidance conditions based on closed-form, outer approximations $\widetilde{\mathcal{M}} \supseteq \mathcal{M}$ of the Minkowski sum. We focus on the important case in which the obstacle is a bounded, convex polytope and the vehicle is represented by Euclidean balls (possibly multiple). Building upon recent successes of sum-of-squares (SOS) optimization for outer approximating semialgebraic sets [7, 8, 10, 12, 42], we develop SOS programs for finding $\widetilde{\mathcal{M}}$. Figure 4-1 shows an example of the resulting outer approximations. We then use $\widetilde{\mathcal{M}}$ to perform optimization-based motion planning of an autonomous car and quadcopter navigating cluttered environments. Compared to the exact method [35], our approximate method solves 4.8x (car) and 8.7x (quadcopter) faster while introducing minimal conservatism arising from the use of outer approximations.

The rest of the chapter is organized as follows. Section II reviews relevant aspects of convex sets, Minkowski sums, and SOS optimization. Section III defines the motion planning problem. Section IV poses the obstacle avoidance constraints using Minkowski sums and provides methods for outer approximating the set. Section V applies our approach to motion planning for an autonomous car and quadcopter. Section VI concludes the chapter with a discussion of future directions.

4.2 Preliminaries

We briefly review some basic properties of convex sets, Minkowski sums, and sum-of-squares polynomials. This is mostly done to setup our notation. The reader is referred to [16, 34, 43] for proofs and further details.

4.2.1 Set Definitions

Definition 4.1 (Convex Hull)**.** The convex hull of a set $\mathcal{B}$ is defined as: $\mathbf{conv}\ \mathcal{B} = \{\theta_1 x_1 + \ldots + \theta_k x_k \ \mid x_i \in \mathcal{B}, \theta_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \theta_i = 1\}$. Let $\mathcal{C}$ be any convex set that contains $\mathcal{B}$. The convex hull is the smallest convex set that contains $\mathcal{B}$:

$$\mathcal{B} \subseteq \mathcal{C} \Leftrightarrow \mathbf{conv}\ \mathcal{B} \subseteq \mathcal{C} \tag{4.1}$$

Definition 4.2 (α-sublevel Set)**.** The α-sublevel set of a function $f : \mathbb{R}^n \to \mathbb{R}$ is: $B_\alpha = \{x \,|\, f(x) \leq \alpha\}$. We denote the boundary of the set as $\partial B_\alpha = \{x \,|\, f(x) = \alpha\}$.

Lemma 4.1. *Let $\mathcal{B}_\alpha$ be a convex set that is the α-sublevel set of a function $f : \mathbb{R}^n \to \mathbb{R}$. Then $\mathcal{B}_\alpha = \mathbf{conv}\ \partial\mathcal{B}_\alpha$.*

We use the notation $-\mathcal{B} = \{-b \,|\, b \in \mathcal{B}\}$ to represent the set $\mathcal{B}$ reflected about the origin. Note that $-\mathcal{B}$ is convex if and only $\mathcal{B}$ is convex.

Definition 4.3 (Polytope)**.** A polytope $\mathcal{P}$ is defined as the solution set of j linear inequalities in $\mathbb{R}^n$. This set is convex by construction. We impose the additional requirement that the set is bounded. The linear inequalities give the halfspace representation

$$\mathcal{P} = \{x \,|\, Ax \leq b\} \tag{4.2}$$

where $A \in \mathbb{R}^{j \times n}, b \in \mathbb{R}^j$. Alternatively, the polytope can be represented by the convex hull of its k vertices

$$\mathcal{P} = \mathbf{conv}\,\{v_1, v_2, \ldots, v_k\} \tag{4.3}$$

where $v_i \in \mathbb{R}^n, i \in [k] := \{1, \ldots, k\}$.

4.2.2 Minkowski Sum Properties

Definition 4.4 (Minkowski Sum)**.** Given two sets $\mathcal{A}, \mathcal{B}$, their Minkowski sum is defined as follows:

$$\mathcal{A} \oplus \mathcal{B} = \{a + b \,|\, a \in \mathcal{A}, b \in \mathcal{B}\} \tag{4.4}$$

Lemma 4.2. *If A and B are convex sets then $A \oplus B$ is convex.*

Lemma 4.3. *For any sets $\mathcal{A}, \mathcal{B}$ the following equality holds:*

$$\mathbf{conv}\,(A \oplus B) = \mathbf{conv}\,(A) \oplus \mathbf{conv}\,(B) \tag{4.5}$$

4.2.3 Sum-of-Squares Optimization

For $x \in \mathbb{R}^n$, let $\mathbb{R}[x]$ denote the set of polynomials in x with real coefficients.

Definition 4.5 (Sum-of-Squares Polynomial)**.** A polynomial $p(x) \in \mathbb{R}[x]$ is a sum-of-squares (SOS) polynomial if there exists polynomials $q_i(x) \in \mathbb{R}[x], i \in [j]$ such that $p(x) = \sum\limits_{i \in [j]} q_i^2(x)$. We use $\Sigma[x]$ to denote the set of SOS polynomials in x. A polynomial of degree $2d$ is a SOS polynomial if and only if there exists a positive semi-definite matrix P (the Gram matrix) such that $p(x) = z(x)^T P z(x)$ where $z(x)$ is the vector of all monomials of x up to degree d [16].

Note that a polynomial being SOS is a sufficient condition for the polynomial to be non-negative (i.e. $p(x) \geq 0 \,\forall\, x$).

Definition 4.6 (SOS-Convex)**.** A polynomial $p(x)$ is SOS-convex if the following holds

$$u^T \nabla^2 p(x) u \in \Sigma[x, u] \tag{4.6}$$

where $u, x \in \mathbb{R}^n$. SOS-convexity is a sufficient condition for the Hessian of $p(x)$ to be positive semi-definite and therefore $p(x)$ to be convex.

In the development that follows, we will be interested in solving slight variations of the following problem.

$$\begin{aligned} \min_{P} \quad & -\log \det P && \text{(4.7a)} \\ \text{s.t.} \quad & \\ & P \succeq 0, \quad p(x) = z(x)^T P z(x), && \text{(4.7b)} \\ & 1 - p(x) \geq 0 \quad \forall\, x \in \mathcal{X}, && \text{(4.7c)} \end{aligned}$$

Here $\mathcal{X}$ is a semialgebraic set defined by n_i polynomial inequalities and n_j polynomial equalities.

$$\mathcal{X} = \{x \mid g_i(x) \geq 0, i \in [n_i], h_j(x) = 0, j \in [n_j]\} \tag{4.8}$$

Equation (4.7b) constrains $p(x)$ to be a SOS polynomial. Equation (4.7c) is a set-containment condition. The generalized $\mathcal{S}$-procedure provides a sufficient condition for the set-containment to hold [16]. For each polynomial equality $g_i(x)$ or inequality $h_j(x)$ describing the set $\mathcal{X}$, we introduce a non-negative polynomial $\lambda_i(x)$ or polynomial $\mu_j(x)$ respectively. The generalized $\mathcal{S}$-procedure involves replacing (4.7c) with the following:

$$1 - p(x) - \sum_i \lambda_i(x) g_i(x) - \sum_j \mu_j(x) h_j(x) \geq 0, \tag{4.9}$$

$$\lambda_i(x) \geq 0 \quad i \in [n_i] \tag{4.10}$$

By replacing the non-negativity constraints in (4.9), (4.10) with the more restrictive condition that the expressions be SOS polynomials, we obtain a semidefinite program which is readily solved.

$$
\begin{aligned}
&\min_{P,\, \lambda_{[1:n_i]}(x),\, \mu_{[1:n_j]}(x)} \quad -\log\det P \\
&\qquad\text{s.t.} \\
&P \succeq 0, \quad p(x) = z(x)^T P z(x), && (4.11) \\
&1 - p(x) - \sum_i \lambda_i(x) g_i(x) - \sum_j \mu_j(x) h_j(x) \in \sum[x], \\
&\lambda_i(x) \in \Sigma[x], \quad i \in [n_i]\,.
\end{aligned}
$$

Note when a polynomial is listed as a decision variable, e.g., $\lambda_{[1:n_i]}(x)$ and $\mu_{[1:n_j]}(x)$ underneath the min, it is implied that the monomial basis is specified and the coefficients are decision variables.

Remark 4.1. Representing an equality constraint requires introducing a polynomial $\mu(x)$. In contrast, representing an inequality requires introducing a SOS polynomial $\lambda(x)$ which has a smaller feasible set and creates an additional semidefinite constraint. As such, it is generally advantageous to represent sets using equalities when applying the generalized $\mathcal{S}$-procedure.

In the development that follows we focus on transforming problems of interest into the form of (7). Once in this form, the subsequent application of the generalized $\mathcal{S}$-procedure is mechanical.

4.3 Problem Statement

We now setup the problem of optimization-based motion planning with collision avoidance constraints. For convenience, our notation closely follows that of [36].

4.3.1 Vehicle and Obstacle Models

Consider a vehicle with states $x_k \in \mathbb{R}^{n_x}$ and inputs $u_k \in \mathbb{R}^{n_u}$ at time step k. The dynamics evolve according to $x_{k+1} = f(x_k, u_k)$ where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. The vehicle occupies space in $\mathbb{R}^n$. The vehicle's shape is assumed to be represented by n_b Euclidean balls with radii $r^{(i)}$.

$$\mathcal{B}^{(i)} = \{y \in \mathbb{R}^n \mid \|y\|_2 \leq r^{(i)}\},\ i \in [n_b]. \tag{4.12}$$

The center of each ball is a function of the vehicle's state as given by $t^{(i)} : \mathbb{R}^{n_x} \to \mathbb{R}^n$. At time index k, the space occupied by ball i is given by:

$$\mathcal{V}^{(i)}(x_k) = \mathcal{B}^{(i)} + t^{(i)}(x_k)\,. \tag{4.13}$$

The union $\bigcup\limits_i \mathcal{V}^{(i)}(x_k)$ gives the total space occupied by the vehicle at time index k. For ease of exposition, in what follows we focus w.l.o.g. on the case when the vehicle is represented by a single ball and drop the superscript (i).

We assume there are M obstacles present in the environment indexed by $m \in [M]$. Each obstacle $\mathcal{O}^{(m)}$ is a polytope (closed, convex) with $k^{(m)}$ vertices $\{v_1, \ldots, v_{k^{(m)}}\}$ defining the convex hull as in (4.3). Equivalently represented in halfspace form (4.2), the obstacle m is defined by $j^{(m)}$ constraints given by $A^{(m)} \in \mathbb{R}^{j^{(m)} \times n}, b^{(m)} \in \mathbb{R}^{j^{(m)}}$.

4.3.2 Optimal Control Problem

We consider an optimal control problem of controlling the vehicle over N steps. The vehicle starts at state x_S and must end at final state x_F. Let X, U denote the vector of all states and controls respectively, $X = [x_0^T, \ldots, x_N^T]^T, U = [u_0^T, \ldots, u_{N-1}]^T$. We seek to minimize an objective $l(X, U)$ where $l : X \times U \to \mathbb{R}$. Additionally, the vehicle is subject to n_h constraints given by $h(X, U) \leq 0$ where $h : X \times U \to \mathbb{R}^{n_h}$ and the inequality is interpreted element-wise. We assume that $l(X, U)$ and $h(X, U)$ are continuously differentiable and therefore suitable for nonlinear programming solvers which utilize gradient and Hessian information. Lastly, we enforce collision avoidance constraints between each obstacle and the vehicle. The resulting optimization problem takes the following form:

$$
\begin{aligned}
\min_{X,U} \quad & l(X,U) && (4.14a)\\
\text{s.t.} \quad & && \\
& x_0 = x_S, \quad x_N = x_F, && (4.14b)\\
& x_{k+1} = f(x_k, u_k), \quad k = 0, \ldots, N-1 && (4.14c)\\
& h(X,U) \leq 0, && (4.14d)\\
& \mathcal{V}(x_k) \cap \mathcal{O}^{(m)} = \emptyset, \quad k \in [N],\ m \in [M]. && (4.14e)
\end{aligned}
$$

Equation (4.14e) represents the collision avoidance constraints which are non-convex and non-smooth in general. In [35], the authors provide an exact, smooth reformulation of these constraints. As the distance between two convex shapes can be computed using convex optimization, the authors leverage strong duality to develop necessary and sufficient conditions for a Euclidean ball of radius r to not intersect a given convex shape. This requires introducing dual variables associated with the halfspace constraints representing each obstacle $\lambda_k^{(m)} \in \mathbb{R}^{j^{(m)}}$, $k \in [N]$, $m \in [M]$ and replacing (4.14e) with the following constraints.

$$
\begin{aligned}
(A^{(m)} t(x_k) - b^{(m)})^T \lambda_k^{(m)} > r,\\
\|A^{(m)^T} \lambda_k^{(m)}\|_2^2 \leq 1,\\
\lambda_k^{(m)} \geq 0,\\
k \in [N],\ m \in [M].
\end{aligned}
\tag{4.15}
$$

If each obstacle has L halfspace constraints, this method introduces $(2 + L)MN$ constraints and LMN dual variables which can result in a large nonlinear program that is computationally intensive. In the following, we present a method for approximating the collision avoidance constraints while introducing only MN constraints and no additional variables.

4.4 Collision Avoidance via Minkowski Sums

We will utilize Minkowski sums to represent the collision avoidance constraints between a closed, convex polytope obstacle $\mathcal{O} = \{y \in \mathbb{R}^n \,|\, a_i^T y \leq b_i, i \in [L]\}$ and a vehicle with shape given by the Euclidean ball $\mathcal{B} = \{w \in \mathbb{R}^n \,|\, w^T w \leq r^2\}$. We first review a fundamental result from computational geometry.

Lemma 4.4. *Let $\mathcal{O}$ and $\mathcal{B}$ be sets in $\mathbb{R}^n$. Let $\mathcal{V} = \mathcal{B} + d$ be the set $\mathcal{B}$ translated by $d \in \mathbb{R}^n$. Then the following relation holds:*

$$\mathcal{O} \cap \mathcal{V} \neq \emptyset \Leftrightarrow d \in \mathcal{O} \oplus (-\mathcal{B}) \tag{4.16}$$

Proof. See, e.g. [33, 44] □

In words, when $\mathcal{B}$ is located at position d, it makes contact with $\mathcal{O}$ if and only if d is in the Minkowski sum $\mathcal{O} \oplus (-\mathcal{B})$. Thus collision avoidance with respect to obstacle $\mathcal{O}$ is equivalent to ensuring $d \notin \mathcal{O} \oplus (-\mathcal{B})$.

When $\mathcal{O}$ is a polytope and $\mathcal{B}$ is a Euclidean ball, the set $\mathcal{O} \oplus (-\mathcal{B})$ is semialgebraic. As such we cannot directly include the condition $d \notin \mathcal{O} \oplus (-\mathcal{B})$ as a constraint in a nonlinear optimization problem which requires closed-form, twice differentiable expressions.

Instead we propose to find an outer approximation $\mathcal{O} \oplus (-\mathcal{B}) \subseteq \widetilde{\mathcal{M}} \subset \mathbb{R}^n$ where $\widetilde{\mathcal{M}}$ is defined as the 1-sublevel set of a function $p : \mathbb{R}^n \to \mathbb{R}$. Recall in our setting the translation of the ball at time index k is a function of the vehicle's state x_k as given by $t : \mathbb{R}^{n_x} \to \mathbb{R}^n$. Collision avoidance with respect to obstacle $\mathcal{O}$ can then be ensured by imposing the constraint $p(t(x_k)) > 1 \Leftrightarrow t(x_k) \notin \widetilde{\mathcal{M}} \Rightarrow t(x_k) \notin \mathcal{O} \oplus (-\mathcal{B}) \Leftrightarrow \mathcal{O} \cap \mathcal{V} = \emptyset$.

If multiple obstacles $\mathcal{O}^{(m)}, m \in [M]$ are present, we repeat this process for each obstacle and denote the associated function as $p^{(m)}(x)$. In our trajectory optimization problem we replace (4.14e) with MN constraints.

$$p^{(m)}(t(x_k)) > 1, \quad k \in [N], \quad m \in [M]. \tag{4.17}$$

4.4.1 Outer Approximations of the Minkowski Sum

We would like our outer approximations to closely approximate the true set. To do so, we pose an optimization problem in which we minimize the volume of the outer approximation.

$$
\begin{aligned}
&\min_{p(x)} \quad \operatorname{vol} \widetilde{\mathcal{M}} \\
&\text{s.t.} \\
&1 - p(x) \geq 0 \qquad\quad \forall\, x \in \mathcal{O} \oplus (-\mathcal{B}), \\
&\widetilde{\mathcal{M}} = \{x \,|\, p(x) \leq 1\}
\end{aligned}
\tag{4.18}
$$

In general we cannot solve this optimization problem. To arrive at a tractable formulation, we apply the generalized $\mathcal{S}$-procedure. We first parameterize the polynomial as $p(x) = z(x)^T P z(x)$ where $z(x)$ is a monomial basis chosen by the user and P is a positive semi-definite matrix of appropriate dimension. For arbitrary polynomials, we lack an expression for minimizing the volume of the 1-level set. Various heuristics have been proposed [7, 8, 10, 42]. We have found maximizing the determinant of P, as proposed in [10], to work well for the problems herein. The resulting optimization problem is

$$
\begin{aligned}
&\min_{P} \quad - \log \det P \\
&\text{s.t.} \\
&p(x) = z(x)^T P z(x), \quad P \succeq 0, \\
&1 - p(x) \geq 0 \qquad\quad \forall\, x \in \{y - w \,|\, a_i^T y \leq b_i, \\
&\qquad\qquad\qquad\qquad\qquad\qquad w^T w \leq r^2, i \in [L]\}
\end{aligned}
\tag{4.19}
$$

where we have explicitly written the set resulting from the Minkowski sum in terms of y and w along with inequalities that ensure $y \in \mathcal{O}$ and $w \in \mathcal{B}$.

We apply the $\mathcal{S}$-procedure to replace the set-containment condition with a sufficient condition. This requires introducing multipliers $\lambda(y, w)$. We then replace the non-negativity conditions with the sufficient condition that the expression admits a SOS decomposition in terms of variables y and w.

Optimization Problem 1: Outer Approximation

$$\begin{aligned}
&\min_{P,\lambda_{[0:L]}(y,w)} \quad -\log\det P \\
&\quad\text{s.t.} \\
&p(x) = z(x)^T P z(x), \quad P \succeq 0, \qquad\qquad \text{(OA)} \\
&1 - p(y-w) - \lambda_0(y,w)(r^2 - w^T w) \\
&\qquad - \sum_{i=1}^{L} \lambda_i(y,w)(b_i - a_i^T y) \in \sum[y,w] \\
&\lambda_i(y,w) \in \Sigma[y,w] \quad i = 0,\dots,L
\end{aligned}$$

The formulation given by (OA) is viable but computationally expensive because the SOS decompositions involve both w and y giving $2n$ free variables for $x \in \mathbb{R}^n$. As we seek higher-order approximations, the monomial basis grows rapidly in size leading to large semidefinite programs. We now develop a computationally cheaper program by leveraging convexity.

4.4.2 Convex Outer Approximations of the Minkowski Sum

In developing an efficient method for outer approximating the Minkowski sum, we will utilize the following Lemma.

Lemma 4.5. *Let $\mathcal{O} \subset \mathbb{R}^n$ be a polytope with K vertices $\{v_i\}, i \in [K]$. Let $\mathcal{B} \subset \mathbb{R}^n$ be a convex set that is the α-sublevel set of a function $f : \mathbb{R}^n \to \mathbb{R}$. Let S be any convex set in $\mathbb{R}^n$. Then the following relation holds:*

$$\mathcal{O} \oplus (-\mathcal{B}) \subseteq S \Leftrightarrow \{v_i\} \oplus (-\partial\mathcal{B}) \subseteq S \tag{4.20}$$

Proof. First represent $\mathcal{O} \oplus (-\mathcal{B})$ in terms of its convex hull.

$$\begin{aligned}
\mathcal{O} \oplus (-\mathcal{B}) &= \mathbf{conv}\{v_i\} \oplus \mathrm{conv}(-\partial\mathcal{B}), && \textbf{Lemma 4.1} \\
&= \mathrm{conv}[\{v_i\} \oplus (-\partial\mathcal{B})], && \textbf{Lemma 4.3}
\end{aligned}$$

Next apply the property of the convex hull (4.1).

$$\operatorname{conv}[\{v_i\} \oplus (-\partial\mathcal{B})] \subseteq S \Leftrightarrow \{v_i\} \oplus (-\partial\mathcal{B}) \subseteq S$$

□

Lemma 4.5 provides a more efficient condition for outer approximating the Minkowski sum with a set $\widetilde{\mathcal{M}} = \{x \,|\, p(x) \leq 1\}$ as we only have to consider the vertices of $\mathcal{O}$ and the boundary of $\mathcal{B}$ in our set-containment constraint as follows.

$$1 - p(x) \geq 0 \quad \forall\, x \in v_i \oplus (-\partial\mathcal{B}) \quad i \in [K]. \tag{4.21}$$

However, it requires the condition that $\widetilde{\mathcal{M}}$ be a convex set. We argue that this is a reasonable constraint as $\mathcal{O} \oplus (-\mathcal{B})$ is itself convex. It is difficult to impose the condition that the 1-sublevel set of $p(x)$, i.e., $\widetilde{\mathcal{M}}$, is convex. Recalling Definition 4.6, we will instead impose the sufficient condition that the function $p(x)$ be sos-convex.

As done previously, we rewrite the set-containment conditions using the generalized $\mathcal{S}$-procedure. We then replace the non-negativity conditions with SOS conditions.

Optimization Problem 2: Convex Outer Approximation

$$\begin{aligned}
&\min_{P,\, \mu_{[1:K]}(w)} \quad -\log\det P \\
&\quad\text{s.t.} \\
&P \succeq 0,\ p(x) = z(x)^T P z(x) \\
&1 - p(v_i - w) - \mu_i(w)(r^2 - w^T w) \in \Sigma[w], i \in [K] \\
&u^T \nabla^2 p(x) u \in \Sigma[x, u]
\end{aligned} \tag{COA}$$

Remark 4.2. The formulation of (COA) is advantageous in that the multipliers $\mu(w)$ do not have to be SOS and they only depend on n free variables ($w \in \mathbb{R}^n$). In contrast, (OA) requires SOS multipliers $\lambda(w, y)$ which depend on $2n$ free variables ($w, y \in \mathbb{R}^n$). The former leads to smaller semidefinite programs which scale better with respect to the dimension n or the complexity of $\mathcal{O}$. This is numerically illustrated in the following example.

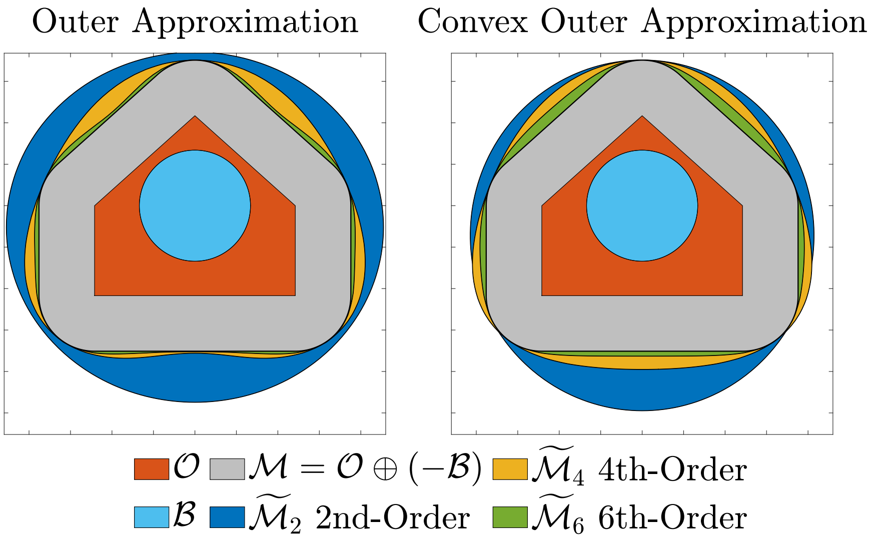


Figure 4-2. Outer approximation of Minkowski sum.

4.4.3 2D Example

We generate 1000 random test cases in $\mathbb{R}^2$. For each case we generate a polytope $\mathcal{O}$ with $n \in \{3, 4, \ldots, 12\}$ vertices $v_i \in [-1, 1]^2, i \in [n]$ along with a disk $\mathcal{B}$ with radius $r \in [0, 1]$. We form outer approximations $\widetilde{\mathcal{M}} = \{x \,|\, p(x) \leq 1\}$ of the set $\mathcal{M} = \mathcal{O} \oplus (-\mathcal{B})$ using both (OA) and (COA). For each we consider polynomials $p(x)$ of degree 2, 4 and 6. To assess the accuracy of our outer approximations, we compute the approximation error as $100 \times \frac{\text{Area}(\widetilde{\mathcal{M}}) - \text{Area}(\mathcal{M})}{\text{Area}(\mathcal{M})}$. Table 4-I lists the mean approximation error of the 1000 test cases. Empirically, as we increase the polynomial order, the approximation error is reduced, indicating we are getting better outer approximations. Table 4-II lists the mean solve times. As expected, (COA) has significantly faster solve times than (OA) due to the smaller semidefinite program. Figure 4-2 provides an example of the results.

Table 4-I. Mean Approximation Error of Minkowski Sums

Polynomial Degree	2	4	6
Outer Approximation	40%	9%	3%
Convex Outer Approximation	25%	9%	5%

Remark. For the case when $p(x)$ is a quadratic, the resulting minimum volume $\widetilde{\mathcal{M}}$

Table 4-II. Mean Solve Times (s) of Optimization Problems 1 & 2

Polynomial Degree	2	4	6
Outer Approximation	0.020	0.174	0.925
Convex Outer Approximation	0.004	0.014	0.049

can be found exactly using the semidefinite program for finding the minimum volume outer ellipsoid (MVOE) covering a union of ellipsoids [34]. In this case each ellipsoid is a ball of radius r centered at vertex v_i. Our convex formulation (COA) can be seen as a generalized form of this result. The non-convex case (OA) has a smaller feasible set due to the reliance on SOS multipliers and does not return the minimum volume outer ellipsoid in general. Thus (OA) is only advantageous when seeking non-ellipsoidal approximations.

4.5 Motion Planning Examples

We demonstrate our proposed obstacle avoidance conditions on an autonomous car and quadcopter example. We solve (14) using both the exact representation (4.15) and the approximate representation (4.17) of the collision avoidance constraints. We compute the sub-optimality of the approximate method relative to the exact method as $100 \times \frac{J_{approx} - J_{exact}}{J_{exact}}$ where J_{approx}, J_{exact} are the value of the objective function $l(X, U)$ for the respective solutions.

In each example, the dynamic constraints (14c) are implemented using a 4th-order Runge-Kutta integrator with a time-step of $0.02s$ over $N = 150$ steps giving a 3s time horizon. We use $A^\star$ [45] to find a minimum-distance collision free path on a discretized representation of the environment.[1] This path does not consider the dynamics and is generally not kinematically feasible. We use this to initialize our guess for the

[1] The $A^\star$ computation time takes an average of 7ms in the case of 10 obstacles for the car example and 41ms for the quadcopter example. In both cases the $A^\star$ step is less than 1/10th the NLP runtime. Bypassing this guess generation and using a naive initialization by linearly interpolating from the initial state x_S to the final state x_F resulted in poor solver reliability.

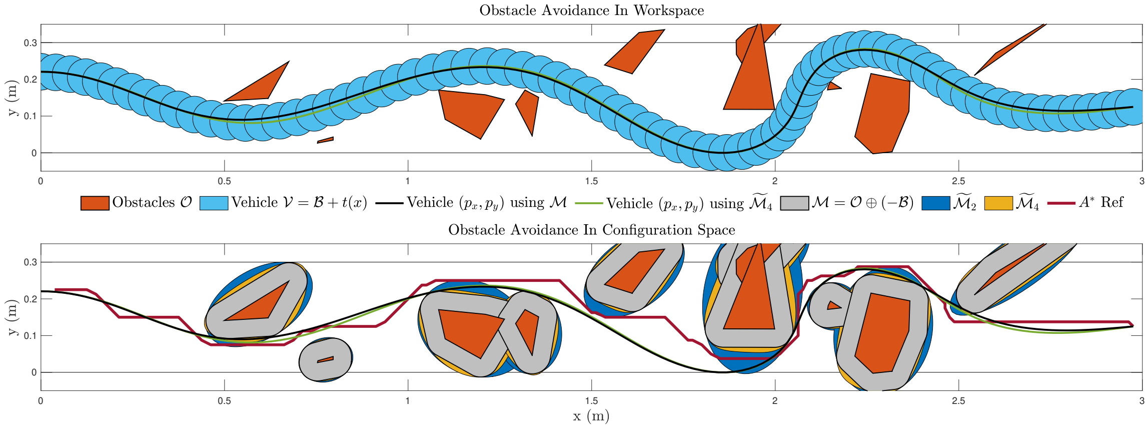


Figure 4-3. Autonomous car navigating obstacles in workspace (upper) and configuration space (lower).

vehicle's states over time. The approximate representation utilizes 4th-order, convex polynomials to represent the Minkowski sums. For the exact method, similar to [35], we initialize the dual variables λ to 0.05.

4.5.1 Autonomous Car

We adopt the autonomous racing car model from [32]. The model has 6 states, $x = \begin{bmatrix} p_x & p_y & \psi & v_x & v_y & \omega \end{bmatrix}^T$ consisting of position (p_x, p_y), orientation (ψ), body velocities (v_x, v_y) and yaw rate (ω). The inputs are motor duty cycle (d) and steering angle (δ). We represent the vehicle's shape as a single disk $\mathcal{B}$ of radius $r = 0.05$m. The center of the disk at time step k is the $(p_{x,k}, p_{y,k})$ position of the vehicle: $t(x_k) = \begin{bmatrix} p_{x,k} & p_{y,k} \end{bmatrix}^T$.

We consider a situation in which the vehicle is making forward progress along a straight track while navigating obstacles. The objective is to minimize the 2-norm of the input, $l(X, U) = \|U\|_2^2$. The vehicle starts at state $x_S = \begin{bmatrix} 0 & p_{y,S} & 0 & 1 & 0 & 0 \end{bmatrix}^T$ and must end at position $(3, p_{y,F})$. At each step $k = 0, \ldots, N-1$, the vehicle is subject to box constraints on the position $p_{x,k} \in [0, 3], p_{y,k} \in [0, 0.3]$ and inputs $d_k \in [-0.1, 1], \delta_k \in [-1, 1]$.

We consider scenarios consisting of $M \in \{1, 2, \ldots, 10\}$ obstacles. For each scenario, we generate 100 random test cases in which we vary the start and final y position,

$p_{y,S}, p_{y,F} \in [0, 0.3]$ along with the placement and shapes of the M obstacles. Figure 4-3 shows a scenario in which the vehicle navigates ten obstacles. We plot the obstacles along with the exact and approximate Minkowski sums of each obstacle and the vehicle $\mathcal{B}$. As the exact method is equivalent to ensuring the vehicle's position (p_x, p_y) remains outside the exact Minkowski sums $\mathcal{M}$, this helps to visualize the conservatism of our outer approximations. The 4th-order approximate representations $\widetilde{\mathcal{M}}_4$ are quite tight and are only visible as thin yellow borders around the exact Minkowski sums in gray. For reference, we also plot 2nd-order, ellipsoidal approximations $\widetilde{\mathcal{M}}_2$ which are unacceptably conservative as the vehicle cannot progress beyond $p_x = 1.9$ without violating constraints. As the objective penalizes large steering and acceleration commands, the vehicle naturally makes tight maneuvers around the obstacles. The exact method returns a slightly better trajectory because the configuration-space obstacles it must avoid are smaller, requiring less maneuvering. The approximate method makes slightly wider turns, resulting in a 4% sub-optimal trajectory. However the difference is minor and the resulting trajectories are nearly identical.

Figure 4-4 shows the solve time statistics of the approximate and exact methods as we vary the number of obstacles present. Comparing median solve times, the approximate method solves 1.6x faster than the exact method when just one obstacle is present. With ten obstacles present, the approximate method solves 4.8x faster. The approximate method shows less variability in the solve times, with a maximum solve time of 0.84s and no failed instances.[2] For the exact method, 240 of the 1000 cases either did not converge or exceeded the maximum allowed solve time of 5s. For 746 of the 760 cases in which the exact method was successfully solved, the approximate method returned a trajectory less than 5% sub-optimal. The worst-case sub-optimality

[2] The solver times reported for the approximate method only reflect the time spent solving the nonlinear program. We do not include the time required to compute the outer approximations. In a real-time motion planning problem, these approximations would only be performed once per obstacle, either offline or online. We note that based on Table 4-II, approximating an obstacle with a 4th-order convex polynomial takes 0.014s. If this computation time were included in Figure 4-4, the approximate method would still be consistently faster than the exact method.

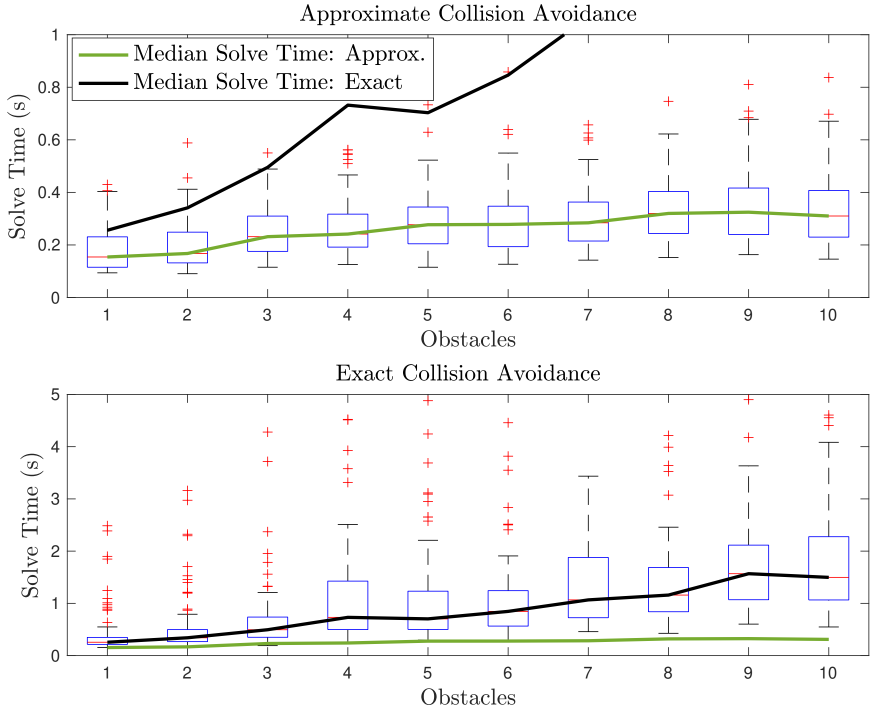


Figure 4-4. Solve time statistics for autonomous car example.

was 18%.

4.5.2 Quadcopter

We consider the quadcopter model from [46]. The model has 12 states consisting of position (p_x, p_y, p_z), velocity (v_x, v_y, v_z), Euler angles (ϕ, θ, ψ), and body rates (p, q, r). The inputs are the four rotor speeds $\omega_i, i \in [4]$ in scaled values. We represent the quadcopter's shape as a single ball $\mathcal{B}$ of radius $r = 0.25$m. The center of the ball at time step k is the position of the quadcopter: $t(x_k) = \begin{bmatrix} p_{x,k} & p_{y,k} & p_{z,k} \end{bmatrix}^T$.

We consider a situation in which the quadcopter is navigating a cluttered room with dimensions $10 \times 10 \times 5$. The quadcopter starts at the origin with state $x_S = \mathbf{0}_{12}$ and must end at state $x_F = \begin{bmatrix} p_{x,F} & p_{y,F} & p_{z,F} & \mathbf{0}_9 \end{bmatrix}^T$ while avoiding any obstacles. Here $\mathbf{0}_i$ denotes the zero vector in $\mathbb{R}^i$. The objective is to minimize the 2-norm of the rotor speed deviation from a trim condition $l(X, U) = \|U - 4.5\|_2^2$ where $\omega_i = 4.5, i \in [4]$ achieves a steady-state, hover condition. At each step $k = 0, \ldots, N-1$, the vehicle is

Table 4-III. Solve Times Statistics for Quadcopter Example

Collision Avoidance	Min. (s)	Median (s)	Max. (s)
Approximate (4th-Order)	0.47	0.74	2.76
Exact	2.37	6.48	18.89

subject to box constraints on the position $p_{x,k} \in [0, 10], p_{y,k} \in [0, 10], p_{z,k} \in [0, 5]$ and inputs $\omega_{i,k} \in [1.2, 7.8], i \in [4]$.

The environment contains 30 obstacles. We consider 174 different final positions. Table 4-III lists the resulting solve times of the nonlinear program. The approximate method solved 8.7x faster than the exact method with respect to median solve times. In 9 instances the exact method failed or exceeded the maximum solve time of 20s. The approximate method was less than 5% sub-optimal for 155 of the remaining 165 test cases. The worst-case was 16% sub-optimal. The upper plot of Figure 4-5 shows the quadcopter navigating the cluttered environment. The lower plot gives the configuration-space view with the Minkowksi sum approximations shown in yellow.

4.5.3 Autonomous Car with Multiple-Disc Geometry

The previous examples used vehicle geometries consisting of a single Euclidean ball. We briefly demonstrate how our method can handle vehicle geometries consisting of multiple Euclidean balls. Returning to the autonomous car example, we introduce two additional discs, each with radius 0.05m, to form an “L” shape. The center of each disc is a function of the vehicle’s position (p_x, p_y) and orientation ψ:

$$t^{(i)}(x) = \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} l_x^{(i)} \\ l_y^{(i)} \end{bmatrix} \tag{4.22}$$

where

$$l_{x,y}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ l_{x,y}^{(2)} = \begin{bmatrix} 0.05 \\ 0 \end{bmatrix},\ l_{x,y}^{(3)} = \begin{bmatrix} 0 \\ 0.05 \end{bmatrix}, \tag{4.23}$$

Because the radius of each disc is identical, we can reuse the same approximation $\widetilde{\mathcal{M}} = \{y \in \mathbb{R}^2 \,|\, p^{(m)}(y) \leq 1\}$ for the Minkowski sum of a given obstacle m and a disc

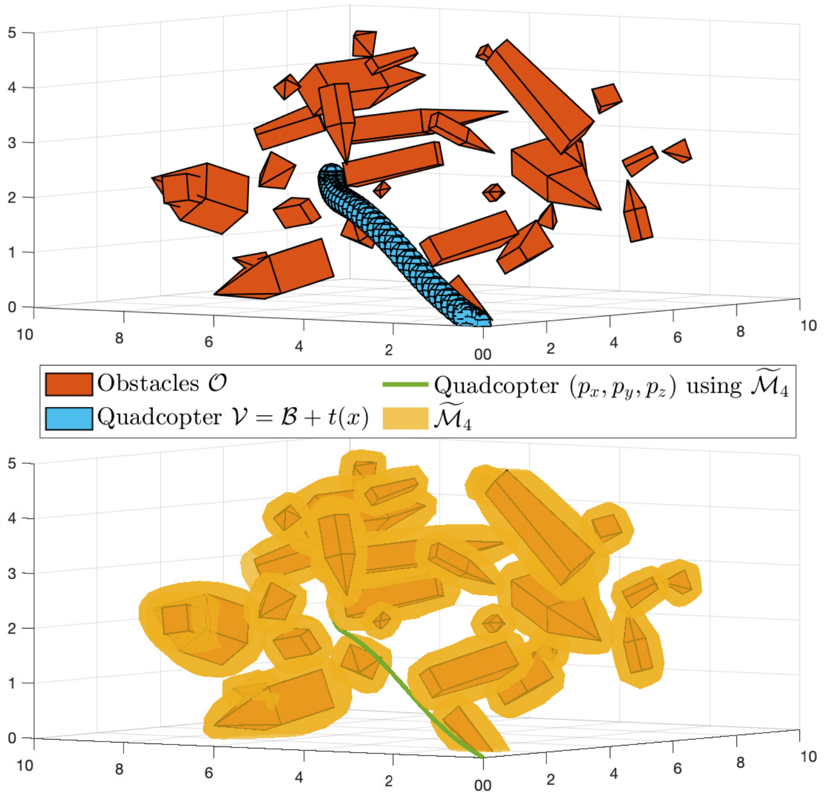


Figure 4-5. Quadcopter navigating in workspace (upper) and C-space (lower).

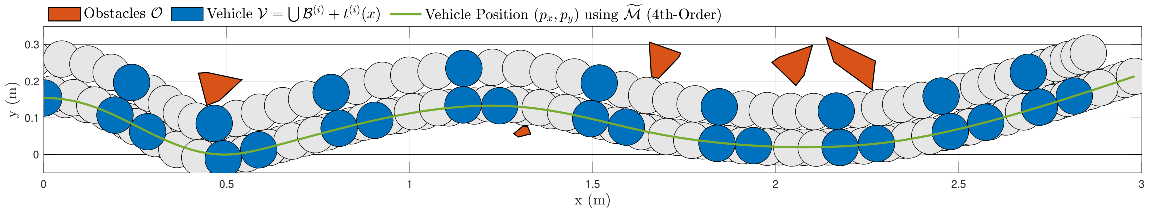


Figure 4-6. L-shaped autonomous car navigating obstacles in workspace.

of radius 0.05m.[3] We simply change the argument $t^{(i)}(x)$ supplied to p, representing the center of the ball i as a function of the vehicle's state x. In this setting (4.17) is replaced with the following:

$$p^{(m)}(t^{(i)}(x_k)) > 1,\ k \in [N],\ m \in [M],\ i \in [3]. \tag{4.24}$$

Figure 4-6 shows the L-shaped vehicle navigating obstacles.

4.5.4 Scaling

One caveat of using SOS polynomials to approximate indicator functions of sets is that the polynomial may return large values for points far outside of the set. This may be problematic for NLP solvers which are often sensitive to the scaling of the problem. To improve the scaling, we can apply any smooth function $q(z) : \mathbb{R}_{\geq 0} \to \mathbb{R}$ to (4.17) which is strictly-increasing for $z \geq 0$. In the examples shown we have utilized $q(z) = -\exp(-z)$, replacing (4.17) with:

$$-\exp(-p^{(m)}(t(x_k))) > -\exp(-1), \quad k \in [N], \quad m \in [M]. \tag{4.25}$$

As p is a SOS polynomial, the left-hand side of this equivalent formulation takes on values in the range $[-1, 0]$.

4.5.5 Implementation Details

All examples were solved on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 CPU. YALMIP [22] was used in conjunction with MOSEK [21] to solve the SOS optimization problems. IPOPT [47] with the MA27 linear solver was used to solve the nonlinear optimization problems with exact gradients and Hessians supplied by CasADi [48].

[3] If this were not the case we would have to compute separate approximations for each unique radius value.

4.6 Conclusions

This work presented novel obstacle avoidance conditions based on outer approximations of Minkowski sums. This method is advantageous in that it yields a much smaller nonlinear program compared to exactly representing the collision avoidance conditions. On motion planning problems for an autonomous vehicle and quadcopter, the approximate method solved 4.8x and 8.7x faster respectively when navigating cluttered environments. The resulting trajectories exhibited minimal sub-optimality compared to using exact collision avoidance conditions. Currently our method is limited to cases in which the vehicle is represented by a union of Euclidean balls and the obstacle is a bounded, convex polytope. In future work we plan to consider representations of non-convex obstacles.

Chapter 5

A Differentiable Signed Distance Representation for Continuous Collision Avoidance in Optimization-Based Motion Planning

5.1 Introduction

The previous chapter introduced the problem of representing collision avoidance constraints within motion planning algorithms based on nonlinear optimization. This is challenging as we generally lack smooth, closed-form representations of the condition $\mathcal{V} \cap \mathcal{O} = \emptyset$, where $\mathcal{V}$ represents the vehicle's occupied space and $\mathcal{O}$ represents the obstacle. A solution was proposed based on approximating the indicator function of the Minkowski sum condition for collision avoidance. This approach is appealing as it does not require introducing additional variables.

A series of works have shown how collision avoidance conditions can be suitably represented within an NLP problem by introducing a set of differentiable conditions and auxiliary variables that collectively ensure $\mathcal{V} \cap \mathcal{O} = \emptyset$. All of these methods focus on specific classes of convex sets and then leverage various results from convex analysis which provide certificates that two sets do not intersect. In [38] the authors utilize a polar set representation of polyhedrons to establish differentiable conditions for ensuring a point mass vehicle does not make contact with a polyhedral obstacle. In [37] the authors leveraged Farkas' Lemma to arrive at conditions ensuring collision avoidance between a polyhedral robot and polyhedral obstacle. In [35] the authors utilized the dual formulation of distance calculations as given in [34] to ensure a minimum signed distance (a generalization of collision avoidance) between convex objects modeled as the intersection of linear and second-order cone constraints. All of these works require introducing additional variables and constraints into the problem. Generally the number of variables is proportional to the complexity of the geometry being represented. The resulting growth in problem size can quickly become burdensome.

Beyond the challenge of computational complexity, all of these methods only address collision avoidance at discrete time instances arising from the transcription method utilized. The solver may exploit this discrete approximation of a continuous-time problem and return solutions which cut corners or pass through thin walls in an attempt to minimize the objective. In computational geometry this is a well-studied problem known as "tunneling" as it can occur when a fast-moving bullet in a video game passes through thin walls. Continuous collision detection refers to the class of algorithms in computational geometry which ensure robust collision checking at all time instances, not just discrete time points (e.g. between frame updates in a video game). These methods often rely on various approximations of the swept volume [49].

Ensuring continuous collision detection within optimization-based motion planners is an open issue. In [50], the authors present an exact approach for a specific class of dynamic models controlling point-mass vehicles navigating circles and cylinders. In [51] the authors develop a trajectory optimization algorithm that approximately ensures continuous collision detection for polyhedral robots navigating polyhedral obstacles. The method utilizes a linear approximation of the non-differentiable signed distance function. At points of non-differentiability, the resulting gradient information is inaccurate making the method ill-suited for use with standard NLP solvers which expect exact gradients. Instead the authors provide a custom solver based on sequential convex optimization. Beyond this, to the author's knowledge, no other methods exist for rigorously addressing continuous collision avoidance within optimization-based motion planners. Instead, various heuristic fixes are generally utilized. The most common is to inflate obstacles along with introducing velocity constraints on the vehicle to prevent it from passing through an obstacle in one time step [33]. However, this artificially reduces the configuration space of the problem, making tight maneuvering infeasible. Additionally it typically requires a smaller time step, leading to more decision variables in the transcription method and therefore larger (slower) optimization problems.

5.1.1 Contributions

In this work, we propose a novel formulation of signed distance constraints for collision avoidance by deriving necessary and sufficient conditions related to the support function representation of convex sets. These conditions are continuously differentiable and can be utilized within standard optimization-based motion planning algorithms based on nonlinear programming. Compared to existing approaches [35, 37, 38], our method introduces fewer variables and constraints leading to smaller nonlinear programs. Additionally our formulation allows us to represent sets given by the convex hull of other sets. We utilize this capability to develop sufficient conditions for ensuring continuous collision avoidance within an optimization-based motion planning algorithm. To our knowledge, this is the first method for rigorously ensuring continuous collision avoidance within optimization-based motion planners for arbitrary vehicle dynamics and full-dimensional (vice point mass) geometries. We demonstrate its use on an autonomous vehicle model performing tight maneuvering around obstacles in which a discrete collision avoidance approach fails.

5.2 Background

5.2.1 Notation

Let $[n] := \{1, 2, \dots, n\}$. Let $\mathbb{S}^n_{++}$ denote the set of $n \times n$ positive definite matrices. Let $SO(n)$ denote the special-orthogonal group in dimension n. Let $\|c\| := \|c\|_2$, denote the Euclidean norm of $c \in \mathbb{R}^n$. Let $B_r := \{x \mid \|x\|_2 \leq r\}$. Given $\mathcal{A} \subset \mathbb{R}^n$, $R \in \mathbb{R}^{n \times n}$, and $v \in \mathbb{R}^n$, let $R\mathcal{A} + v := \{Rx + v \mid x \in \mathcal{A}\}$.

5.2.2 Signed Distance

Let $\mathcal{V}, \mathcal{O} \subset \mathbb{R}^n$ be compact sets. The distance between the two objects is

$$\operatorname{dist}(\mathcal{V}, \mathcal{O}) := \min_v \{\|v\| \mid (\mathcal{V} + v) \cap \mathcal{O} \neq \emptyset\}. \tag{5.1}$$

If both $\mathcal{V}$ and $\mathcal{O}$ are convex, the distance can be calculated using convex optimization. The penetration depth is

$$\operatorname{pen}(\mathcal{V}, \mathcal{O}) := \min_v \{\|v\| \mid (\mathcal{V} + v) \cap (\mathcal{O} \setminus \partial\mathcal{O}) = \emptyset\}. \tag{5.2}$$

The penetration depth is the minimum translation needed for $\mathcal{V}$ to not touch the interior of $\mathcal{O}$. Unlike distance calculations involving convex sets, calculating the penetration depth of two convex sets is a non-convex optimization problem with possibly multiple local minima. The signed distance combines the notions of distance and penetration and is given by

$$\operatorname{sd}(\mathcal{V}, \mathcal{O}) := \operatorname{dist}(\mathcal{V}, \mathcal{O}) - \operatorname{pen}(\mathcal{V}, \mathcal{O}). \tag{5.3}$$

A positive signed distance indicates two objects are separated, a negative signed distance indicates they overlap, and a signed distance of zero indicates their boundaries touch.

5.2.3 Support and Cost Functions

Let $\mathcal{A} \subseteq \mathbb{R}^n$ and $c \in \mathbb{R}^n \setminus 0$. The support function of $\mathcal{A}$ is

$$\sigma_{\mathcal{A}}(c) := \sup_{x \in \mathcal{A}} c^T x. \tag{5.4}$$

We will find it convenient to define the following function, which we refer to as the cost function of $\mathcal{A}$:

$$\mu_{\mathcal{A}}(c) := \inf_{x \in \mathcal{A}} c^T x. \tag{5.5}$$

These are related by $\mu_{\mathcal{A}}(c) = -\sigma_{\mathcal{A}}(-c)$. When $\mathcal{A}$ is a convex set, the support and cost functions are convex optimization problems parameterized by the vector c.

The support and cost functions have a number of useful properties which we will utilize in our development.

Proposition 1. *Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$ be convex sets. Let $t, c \in \mathbb{R}^n, R \in \mathbb{R}^{m \times n}, k \in \mathbb{R}_{\geq 0}$. The support and cost functions satisfy the following properties [43]:*

- *Scaling:*

$$\sigma_{k\mathcal{A}}(c) = k\sigma_{\mathcal{A}}(c),\ \mu_{k\mathcal{A}}(c) = k\mu_{\mathcal{A}}(c)$$

- *Linear Transformation:*

$$\sigma_{R\mathcal{A}}(c) = \sigma_{\mathcal{A}}(R^T c),\ \mu_{R\mathcal{A}}(c) = \mu_{\mathcal{A}}(R^T c)$$

- *Translation:*

$$\sigma_{\mathcal{A}+v}(c) = \sigma_{\mathcal{A}}(c) + c^T v$$

$$\mu_{\mathcal{A}+v}(c) = \mu_{\mathcal{A}}(c) + c^T v$$

- *Minkowski Sum:*

$$\sigma_{\mathcal{A} \oplus \mathcal{B}}(c) = \sigma_{\mathcal{A}}(c) + \sigma_{\mathcal{B}}(c)$$

$$\mu_{\mathcal{A} \oplus \mathcal{B}}(c) = \mu_{\mathcal{A}}(c) + \mu_{\mathcal{B}}(c)$$

- *Convex Hull:*

$$\sigma_{\text{co}(\{\mathcal{A},\mathcal{B}\})}(c) = \sup\{\sigma_{\mathcal{A}}(c), \sigma_{\mathcal{B}}(c)\}$$
$$\mu_{\text{co}(\{\mathcal{A},\mathcal{B}\})}(c) = \inf\{\mu_{\mathcal{A}}(c), \mu_{\mathcal{B}}(c)\}.$$

The following lemmas will prove useful in relating the signed distance between two sets to their respective support and cost functions.

Lemma 5.1. *Let $\alpha, \beta \in \mathbb{R}$ and $c \in \mathbb{R}^n$, $\|c\| = 1$. Given halfspaces $\mathcal{H}^+ = \{x \,|\, c^T x \geq \alpha\}$, $\mathcal{H}^- = \{x \,|\, c^T x \leq \beta\}$, then*

$$\text{sd}(\mathcal{H}^+, \mathcal{H}^-) = \alpha - \beta. \tag{5.6}$$

Lemma 5.2. *Given $\mathcal{V} \subseteq \mathcal{V}^+ \subseteq \mathbb{R}^n$, $\mathcal{O} \subseteq \mathcal{O}^+ \subseteq \mathbb{R}^n$, then*

$$\text{sd}(\mathcal{V}, \mathcal{O}) \geq \text{sd}(\mathcal{V}^+, \mathcal{O}^+). \tag{5.7}$$

5.3 Problem Description

5.3.1 Vehicle Dynamics

Consider a continuous-time model of a vehicle with state $x \in \mathbb{R}^{n_x}$, control input $u \in \mathbb{R}^{n_u}$, and dynamics $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ satisfying

$$\dot{x} = f(x, u). \tag{5.8}$$

In numerical optimal control, it is common to approximate continuous-time dynamics with a discrete-time model. The discrete model is obtained by applying a numerical integration method (e.g. Euler, Runge-Kutta) to the continuous-time dynamics over a fixed time interval ΔT. The state and control values are then represented at indices $k \in \mathbb{Z}_+$ corresponding to their values in continuous time at $t_k = k\Delta T$. Let x_k and u_k denote the state and control respectively at time t_k. The value u_k represents a constant control input applied for $t \in [t_k, t_{k+1})$. Let $\phi(x_i, \bar{u}, t_i, t) := x_i + \int_{t_i}^{t} f(x(s), \bar{u}) ds$ denote the solution of (5.8) at time $t \geq t_i$ with initial state x_i and constant control input $u = \bar{u}$. The resulting discrete-time model is given by[1]

$$\begin{aligned} x_{k+1} &= \phi(x_k, u_k, t_k, t_k + \Delta T) \\ &: = f_{\Delta T}(x_k, u_k). \end{aligned} \tag{5.9}$$

We refer to $f_{\Delta T} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ as the discrete-time model of (5.8) with step-size ΔT.

5.3.2 Vehicle Geometry

Let $\mathcal{A} \subset \mathbb{R}^n$ be a compact convex set describing the shape of the vehicle with dynamics (5.8). Let

$$\mathcal{V}(x) := R(x)\mathcal{A} + p(x) \tag{5.10}$$

denote the space occupied by the vehicle where $R : \mathbb{R}^{n_x} \to SO(n), p : \mathbb{R}^{n_x} \to \mathbb{R}^n$ define the rotation and translation respectively. We refer to $\mathcal{V}(x)$ as the state-dependent geometry of the vehicle. The swept volume is defined as the total space occupied (temporarily) by the vehicle over a time interval $[t_i, t_f]$:

$$\mathrm{sv}_{\mathcal{V},f}(x_i, \bar{u}, t_i, t_f) := \bigcup_{t \in [t_i, t_f]} \mathcal{V}(\phi(x_i, \bar{u}, t_i, t)). \tag{5.11}$$

If the vehicle only undergoes linear translation the swept volume is the convex hull of the start and end poses.

Lemma 5.3. *Let the vehicle have continuous-time dynamics* (5.8) *with associated geometry* (5.10). *Given initial state x_i and control input $\bar{u}$, let $x(t) := \phi(x_i, \bar{u}, t_i, t)$ denote the resulting state trajectory for $t \in [t_i, t_f]$. Let $x_f := \phi(x_i, \bar{u}, t_i, t_f)$. Assume $R(x(t)) = R(x_i) \, \forall \, t \in [t_i, t_f]$. Assume $p(x(t)) = (1 - \xi(t))p(x_i) + \xi(t)p(x_f) \, \forall \, t \in [t_i, t_f]$ where $\xi : [t_i, t_f] \to [0, 1]$ is a continuous function with $\xi(t_i) = 0, \xi(t_f) = 1$. Then*

$$\mathrm{sv}_{\mathcal{V},f}(x_i, \bar{u}, t_i, t_f) = \mathrm{co}(\{\mathcal{V}(x_i), \mathcal{V}(x_f)\}). \tag{5.12}$$

[1] Throughout this work, we assume this relation holds exactly such that the discrete-time model has no integration error.

If the vehicle undergoes rotation or nonlinear translation, the resulting swept volume is, in general, non-convex. Further, we cannot determine the swept volume solely from the start and end poses. This presents a challenge for representing the swept volume within a numerical optimal control problem which only models the vehicle pose at discrete time steps. We assume the existence of a function that allows us to outer approximate the swept volume given the start and end poses and control input applied.

Assumption 5.1 (Swept Volume of Vehicle)**.** *Let the vehicle have continuous-time dynamics* (5.8)*, discrete-time dynamics* (5.9) *and associated geometry* (5.10)*. Let the swept volume be given by* (5.11)*. Assume there exists a $\mathcal{C}^2$ function $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_{\geq 0}$ satisfying*

$$\mathrm{sv}_{\mathcal{V},f}(x_k, u_k, t_k, t_{k+1}) \subseteq \mathrm{co}(\{\mathcal{V}(x_k), \mathcal{V}(x_{k+1})\}) \oplus B_{r(x_k,u_k)} \tag{5.13}$$

where

$$B_{r(x_k,u_k)} := \{y \in \mathbb{R}^n \mid \|y\| \leq r(x_k, u_k)\}. \tag{5.14}$$

Figure 5-1 visualizes this outer approximation. The ball $B_{r(x_k,u_k)}$ accounts for the amount by which the convex hull underapproximates the true swept volume. By making the ball's radius a function of the vehicle state and input, we can minimize the extent to which we overapproximate the swept volume. For example, when the vehicle is moving in a straight line, ideally we would have $r(x_k, u_k) = 0$.

5.3.3 Obstacle Geometry

Let $\mathcal{B} \subset \mathbb{R}^n$ be a closed convex set describing the shape of an obstacle. Let

$$\mathcal{O}(t) := S(t)\mathcal{B} + d(t) \tag{5.15}$$

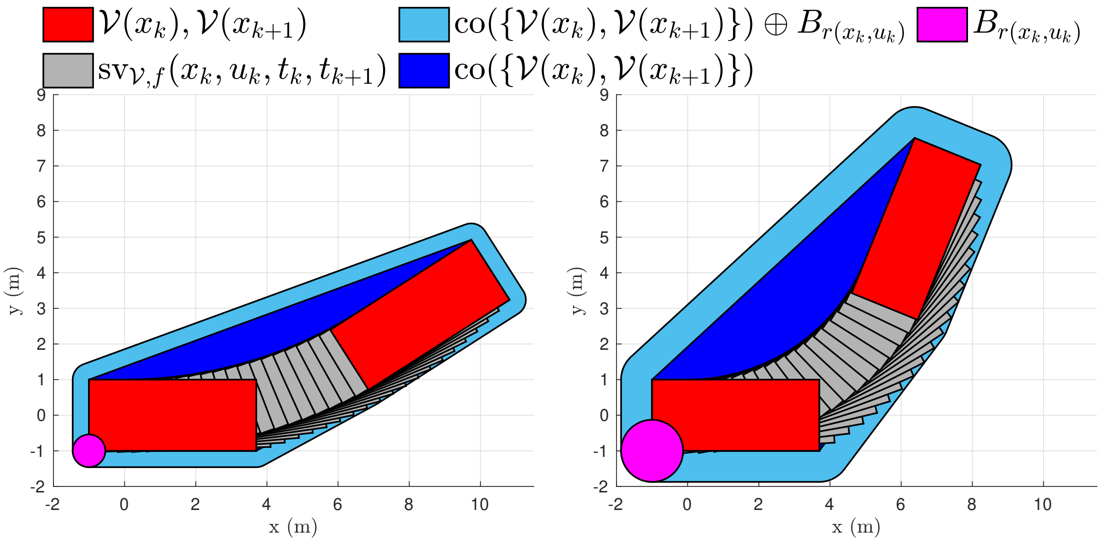


Figure 5-1. Swept volume of vehicle. As the vehicle turns more, the swept volume deviates more from the convex hull of the start and end poses.

denote the space occupied by the obstacle at time t where $S : \mathbb{R} \to SO(n), d : \mathbb{R} \to \mathbb{R}^n$ define the rotation and translation respectively.[2] We refer to $\mathcal{O}(t)$ as the time-dependent geometry of the obstacle. The swept volume of the obstacle is defined as the total space occupied over a time interval $[t_i, t_f]$:

$$\text{sv}_{\mathcal{O}}(t_i, t_f) := \bigcup_{t \in [t_i, t_f]} \mathcal{O}(t). \tag{5.16}$$

In our setting, we are only given the obstacle's pose at $t_k := k\Delta T, k \in \mathbb{Z}_+$, where ΔT is the time step-size. For convenience, let $S_k := S(t_k), d_k := d(t_k)$ and $\mathcal{O}_k := \mathcal{O}(t_k)$ such that the obstacle's pose at time index k is

$$\mathcal{O}_k = S_k \mathcal{B} + d_k. \tag{5.17}$$

We assume that the obstacle's swept volume belongs to the convex hull of the start and end poses inflated by ball B_{w_k}.

Assumption 5.2 (Swept Volume of Obstacle)**.** *Let the obstacle have continuous-time geometry* (5.15) *and discrete-time geometry* (5.17)*. Let the swept volume be given by*

[2] Our notation is chosen to support moving obstacles. For stationary objects we replace $S(t)$ and $d(t)$ with constants.

(5.16). *Let* $w_k \in \mathbb{R}_{\geq 0}$ *satisfy*

$$\mathrm{sv}_{\mathcal{O}}(t_k, t_{k+1}) \subseteq \mathrm{co}(\{\mathcal{O}_k, \mathcal{O}_{k+1}\}) \oplus B_{w_k}. \tag{5.18}$$

5.3.4 Optimization-Based Motion Planning

Consider generating a motion plan over a time horizon $t \in [0,\, T_f]$ for a vehicle with continuous dynamics and geometry given by (5.8) and (5.10) respectively. We use a discrete representation of the dynamics as given by (5.9) with $N\Delta T = T_f$ and $k \in \{0, \ldots, N\}$ for some $N \in \mathbb{Z}_+$. The vehicle starts at state x_S and ends at final state x_F. The vehicle must maintain a minimum signed distance of γ to an obstacle with geometry given by (5.15).[3] Let $X := [x_0^T, \ldots, x_N^T]^T$ and $U := [u_0^T, \ldots, u_{N-1}]^T$ denote the vector of all states and controls respectively. We seek to minimize an objective $l(X, U)$ where $l : X \times U \to \mathbb{R}$. The vehicle is subject to constraints $h(X, U) \leq 0$ where $h : X \times U \to \mathbb{R}^{n_h}$ and the inequality is interpreted element-wise. We assume that $l(X, U)$ and $h(X, U)$ are $\mathcal{C}^2$ functions. The resulting optimization problem is given by

$$\begin{aligned} \min_{X,U} \quad & l(X, U) \\ \text{s.t.} \quad & \\ & x_0 = x_S, \quad x_N = x_F, \\ & x_{k+1} = f_{\Delta T}(x_k, u_k), \;\; k = 0, \ldots, N-1, \\ & h(X, U) \leq 0, \\ & \mathrm{sd}(\mathcal{V}, \mathcal{O}) \geq \gamma. \end{aligned} \tag{5.19}$$

The signed distance function is in general, non-smooth and lacks a closed-form representation. We focus on establishing $\mathcal{C}^2$ conditions that can equivalently represent the signed distance constraints. We first address the case in which the signed distance constraint is imposed at discrete time steps.

[3] We consider a single obstacle to minimize notational clutter. This is without loss of generality as the conditions developed can be repeatedly applied to address the case of multiple obstacles.

Problem 5.1 (Discrete Collision Avoidance)**.** Consider the motion planning problem given by (5.19). Find a set of $\mathcal{C}^2$ constraints that ensure a minimum signed distance of γ at discrete time step t_k using the vehicle state x_k and additional variables $y \in \mathbb{R}^{n_y}$:

$$\operatorname{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma \iff \exists\, x_k, y \,|\, h(x_k, y) \leq 0 \tag{5.20}$$

where $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_c}$ is $\mathcal{C}^2$ and the inequality constraint is interpreted element-wise.

Problem 5.1 only ensures the signed distance constraint is satisfied at time t_k. To ensure the continuous-time trajectory satisfies the signed distance constraint, we evaluate the signed distance using the swept volumes of the vehicle and obstacle over the interval $t \in [t_k, t_{k+1}]$.

Problem 5.2 (Continuous Collision Avoidance)**.** Consider the motion planning problem given by (5.19). Let the vehicle and obstacle swept volumes satisfy Assumptions 5.1 and 5.2. Find a set of $\mathcal{C}^2$ constraints that ensure a minimum signed distance of γ for $t \in [t_k, t_{k+1}]$ using the vehicle state x_k, input u_k and additional variables $y \in \mathbb{R}^{n_y}$:

$$\begin{aligned} &\operatorname{sd}(\operatorname{sv}_{\mathcal{V},f}(x_k, u_k, t_k, t_{k+1}), \operatorname{sv}_{\mathcal{O}}(t_k, t_{k+1})) \geq \gamma \\ &\qquad\qquad \Leftarrow \exists\, x_k, u_k, y \,|\, h(x_k, u_k, y) \leq 0 \end{aligned} \tag{5.21}$$

where $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_c}$ is $\mathcal{C}^2$ and the inequality constraint is interpreted element-wise.

5.4 A Differentiable Signed Distance Representation

We now develop differentiable representations of the signed distance constraints. We focus on establishing this representation for one time step t_k or time interval $[t_k, t_{k+1}]$. This is without loss of generality as these conditions can be repeatedly applied to address multiple time steps or intervals.

5.4.1 Discrete Collision Avoidance

The following lemmas relate the signed distance between two convex sets $\mathcal{C}$ and $\mathcal{D}$ to their cost and support function respectively evaluated for a given vector c.

Lemma 5.4. *Given $\mathcal{C}, \mathcal{D} \subseteq \mathbb{R}^n, c \in \mathbb{R}^n, \|c\| = 1$ then*

$$\operatorname{sd}(\mathcal{C}, \mathcal{D}) \geq \mu_{\mathcal{C}}(c) - \sigma_{\mathcal{D}}(c). \tag{5.22}$$

Lemma 5.5. *Let $\mathcal{C}, \mathcal{D} \subseteq \mathbb{R}^n$ be closed convex sets. Let $\mathcal{C}$ and/or $\mathcal{D}$ be bounded. Then there exists $c \in \mathbb{R}^n, \|c\| = 1$ such that*

$$\operatorname{sd}(\mathcal{C}, \mathcal{D}) = \mu_{\mathcal{C}}(c) - \sigma_{\mathcal{D}}(c). \tag{5.23}$$

Lemma 5.4 suggests a simple method for representing signed distance constraints within a nonlinear program. We introduce a decision variable $c \in \mathbb{R}^n, \|c\| = 1$ along with constraints that make c define a certificate that $\operatorname{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$. Lemma 5.5 guarantees that such a certificate exists.

We will find it convenient to rewrite the cost and support of $\mathcal{V}(x_k)$ and $\mathcal{O}_k$ in terms of the base shape $\mathcal{A}$ and $\mathcal{B}$. Using the properties listed in Proposition 1 yields

$$\begin{aligned}
&\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) \\
&= \mu_{R(x_k)\mathcal{A}+p(x_k)}(c) - \sigma_{S_k\mathcal{B}+d_k}(c) \\
&= (\mu_{R(x_k)\mathcal{A}}(c) + c^T p(x_k)) - (\sigma_{S_k\mathcal{B}}(c) + c^T d_k) \\
&= \mu_{\mathcal{A}}(R(x_k)^T c) - \sigma_{\mathcal{B}}(S_k^T c) + c^T (p(x_k) - d_k).
\end{aligned} \tag{5.24}$$

Lemma 5.6. *Let the vehicle geometry $\mathcal{V}(x_k)$ be given by (5.10). Let the obstacle geometry $\mathcal{O}_k$ be given by (5.17). Then $\operatorname{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$ if and only if there exists $c \in \mathbb{R}^n$ satisfying:*

$$\gamma \leq \mu_{\mathcal{A}}(R(x_k)^T c) - \sigma_{\mathcal{B}}(S_k^T c) + c^T (p(x_k) - d_k), \tag{5.25}$$

$$1 = \|c\|. \tag{5.26}$$

Proof. $\Leftarrow$: From Lemma 5.4, $\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) \geq \gamma \implies \text{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$. $\Rightarrow$: Let $\text{sd}(\mathcal{V}(x_k), \mathcal{O}_k) = \eta$ where $\eta \geq \gamma$. From Lemma 5.5, there exists $c \in \mathbb{R}^n$, $\|c\| = 1$ such that $\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) = \eta$. □

Remark. If $\gamma > 0$ we can relax (5.26) to the convex constraint $\|c\| \leq 1$. To see this, first note that $c = 0$ cannot satisfy (5.25) for $\gamma > 0$ as the right-hand side will evaluate to zero. Now consider a solution c in which $0 < \|c\| < 1$. Multiplying (5.25) by $\frac{1}{\|c\|}$ we obtain

$$\begin{aligned}\frac{1}{\|c\|}\gamma &\leq \frac{1}{\|c\|}(\mu_{\mathcal{A}}(R(x_k)^T c) - \sigma_B(S_k^T c) + c^T(p(x_k) - d_k)\\ &= \mu_A(R(x_k)^T \frac{c}{\|c\|}) - \sigma_B(S_k^T \frac{c}{\|c\|}) + \frac{c^T}{\|c\|}(p(x_k) - d_k).\end{aligned}$$

Let $\tilde{c} = \frac{c}{\|c\|}$. From Lemma 5.4, $\tilde{c}$ provides a certificate that $\text{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \frac{1}{\|c\|}\gamma > \gamma$.

Lemma 5.6 provides conditions for ensuring a minimum signed distance between the vehicle and obstacle. In some applications, it may be desired to also ensure a maximum distance or fixed distance to an obstacle. For example, in surveillance applications we may wish to navigate an environment while remaining within range of a communication tower. Lemma 5.6 can be extended to allow for maximum (or constant) distance constraints. This requires introducing additional variables $o \in \mathcal{O}_k, v \in \mathcal{V}(x_k)$ which provide a certificate of the minimum distance between the objects.

Lemma 5.7. *Let the vehicle geometry $\mathcal{V}(x_k)$ be given by (5.10). Let the obstacle geometry $\mathcal{O}_k$ be given by (5.17). Let $\gamma_{lb}, \gamma_{ub} \in \mathbb{R}$ satisfy $0 \leq \gamma_{lb} \leq \gamma_{ub}$. Then*

$\mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) = \gamma$ *if and only if there exists* $\gamma \in \mathbb{R}, c, o, v \in \mathbb{R}^n$ *satisfying:*

$$\gamma = \mu_{\mathcal{A}}(R(x_k)^T c) - \sigma_{\mathcal{B}}(S_k^T c) + c^T(p(x_k) - d_k), \tag{5.27}$$

$$1 = \|c\|, \tag{5.28}$$

$$o \in \mathcal{O}_k, \tag{5.29}$$

$$v \in \mathcal{V}(x_k), \tag{5.30}$$

$$v = o + \gamma c, \tag{5.31}$$

$$\gamma_{lb} \leq \gamma \leq \gamma_{ub}. \tag{5.32}$$

Proof. $\Leftarrow$: From Lemma 5.4, $\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) = \gamma \implies \mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$. Given $\|c\| = 1$ and $v = o + \gamma c \implies \|o - v\| = \gamma \implies \mathrm{dist}(\mathcal{V}(x_k), \mathcal{O}) \leq \gamma \implies \mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \leq \gamma$. Thus $\mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) = \gamma$. $\Rightarrow$: Let $\mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) = \gamma$ for some $\gamma \geq 0 \implies \mathrm{dist}(\mathcal{V}(x_k), \mathcal{O}_k) = \gamma$. From Lemma 5.5, there exists $c \in \mathbb{R}^n$, $\|c\| = 1$ such that $\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) = \gamma$. Given $\mathrm{dist}(\mathcal{V}(x_k), \mathcal{O}_k) = \gamma \implies \exists\, o \in \mathcal{O}_k, v \in \mathcal{V}(x_k)$ such that $\|o - v\| = \gamma \implies v = o + \gamma d$ for some $d \in \mathbb{R}^n$, $\|d\| = 1$. Assume $d \neq c \implies c^T d < 1 \implies c^T(v - o) = \gamma c^T d < \gamma$. However, $c^T(v - o) \geq \mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) = \gamma$ a contradiction. Thus $d = c$ and there exists c, o, v satisfying $v = o + \gamma c$. □

Lemmas 5.6 and 5.7 provide differentiable representations of signed distance constraints in the case that $\mu_{\mathcal{A}}(c), \sigma_{\mathcal{B}}(c)$ are given by $\mathcal{C}^2$ functions. Points and ellipsoids satisfy this condition. We handle more general convex sets using the dual form of the optimization problems defining the cost and support functions.

Lemma 5.8. *Let the vehicle geometry* $\mathcal{V}(x_k)$ *be given by* (5.10). *Let the obstacle geometry* $\mathcal{O}_k$ *be given by* (5.17). *Let* $\mathcal{A}$ *and* $\mathcal{B}$ *have non-empty interior. Assume the dual form of the cost function of* $\mathcal{A}$ *and support function of* $\mathcal{B}$ *is given by*

$$\mu_{\mathcal{A}}^{\mathrm{dual}}(c) := \sup_{z_\mu}\{f_\mu(c, z_\mu) \,|\, h_\mu(c, z_\mu) \leq 0\}, \tag{5.33}$$

$$\sigma_{\mathcal{B}}^{\mathrm{dual}}(c) := \inf_{z_\sigma}\{f_\sigma(c, z_\sigma) \,|\, h_\sigma(c, z_\sigma) \leq 0\} \tag{5.34}$$

respectively where $z_\mu \in \mathbb{R}^{n_{z_\mu}}, z_\sigma \in \mathbb{R}^{n_{z_\sigma}}, f_\mu : \mathbb{R}^n \times \mathbb{R}^{n_{z_\mu}} \to \mathbb{R}, f_\sigma : \mathbb{R}^n \times \mathbb{R}^{n_{z_\sigma}} \to \mathbb{R}, h_\mu : \mathbb{R}^n \times \mathbb{R}^{n_{z_\mu}} \to \mathbb{R}^{m_\mu}, h_\sigma : \mathbb{R}^n \times \mathbb{R}^{n_{z_\sigma}} \to \mathbb{R}^{m_\sigma}$. Then $\mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$ if and only if there exists $c \in \mathbb{R}^n, z_\mu \in \mathbb{R}^{n_{z_\mu}}, z_\sigma \in \mathbb{R}^{n_{z_\sigma}}$ satisfying:

$$\gamma \leq f_\mu(R(x_k)^T c, z_\mu) - f_\sigma(S_k^T c, z_\sigma) + c^T(p(x_k) - d_k), \tag{5.35}$$

$$0 \geq h_\mu(R(x_k)^T c, z_\mu), \tag{5.36}$$

$$0 \geq h_\sigma(S_k^T c, z_\sigma), \tag{5.37}$$

$$1 = \|c\|. \tag{5.38}$$

Proof. Given $\mathcal{A}$ and $\mathcal{B}$ have non-empty interior, the convex optimization problems defining the cost and support functions satisfy Slater's condition for zero duality gap [34]. Thus there exists z_μ, z_σ such that $\mu_{\mathcal{A}}^{\mathrm{dual}}(c) = \mu_{\mathcal{A}}(c)$ and $\sigma_{\mathcal{B}}^{\mathrm{dual}}(c) = \sigma_{\mathcal{B}}(c)$ respectively. Given zero duality gap, the stated conditions are equivalent to those of Lemma 5.6. □

If the functions $f_\mu, h_\mu, f_\sigma, h_\sigma$ are $\mathcal{C}^2$, then Lemma 5.8 provides a differentiable representation of the signed distance constraints. Lastly, we leverage the convex hull property of the cost and support functions to represent shapes defined by the convex hull of multiple convex sets.

Theorem 5.1. *Let the vehicle geometry $\mathcal{V}(x_k)$ be given by (5.10). Let the obstacle geometry $\mathcal{O}_k$ be given by (5.17). Let $\mathcal{A} = \mathrm{co}(\{\mathcal{A}^{(i)}, i \in [n_{\mathcal{A}}]\})$ and $\mathcal{B} = \mathrm{co}(\{\mathcal{B}^{(j)}, j \in [n_{\mathcal{B}}]\})$ where each $\mathcal{A}^{(i)}, \mathcal{B}^{(j)}$ is a convex set. Then $\mathrm{sd}(\mathcal{V}(x_k), \mathcal{O}_k) \geq \gamma$ if and only if there exists $c \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}$ satisfying*

$$\begin{aligned}
&\alpha \leq \mu_{\mathcal{A}^{(i)}}(R(x_k)^T c), && i \in [n_{\mathcal{A}}],\\
&\beta \geq \sigma_{\mathcal{B}^{(j)}}(S_k^T c), && j \in [n_{\mathcal{B}}],\\
&\gamma \leq \alpha - \beta + c^T(p(x_k) - d_k), &&\\
&1 = \|c\|. &&
\end{aligned}$$

Proof. Note that $\alpha \leq \mu_{\mathcal{A}^{(i)}}(R(x_k)^T c), i \in [n_{\mathcal{A}}] \implies \alpha \leq \mu_{\mathcal{A}}(R(x_k)^T c)$ by the convex hull property of the cost function. Similarly, $\beta \geq \sigma_{\mathcal{B}^{(j)}}(S_k^T c), j \in [n_{\mathcal{B}}] \implies \beta \geq \sigma_{\mathcal{B}}(S_k^T c)$. The remainder of the proof follows the same arguments as Lemma 5.6. □

Remark. If $n_{\mathcal{A}} = 1$ in Theorem 5.1 then we can set $\alpha = \mu_{\mathcal{A}}(R(x_k)^T c)$ without loss of generality. The variable α can be eliminated (replaced with $\mu_{\mathcal{A}}(R(x_k)^T c)$). Similarly, if $n_{\mathcal{B}} = 1$ we can eliminate the variable β. If $n_{\mathcal{A}} = 1$ and $n_{\mathcal{B}} = 1$, Theorem 5.1 reduces to Lemma 5.6.

5.4.2 Examples

We now apply Theorem 5.1 to obtain collision avoidance conditions for polyhedral and ellipsoidal shapes. In doing so, we will see that this formulation introduces fewer variables and constraints than the duality-based formulation of [35]. This can be beneficial for reducing the computational complexity of the nonlinear program. Although our examples are limited to cases in which the vehicle and obstacle shape are of the same class, it is straight-forward to extend these results to cases in which different classes are present (e.g. polyhedral vehicle and ellipsoidal obstacle).

5.4.2.1 Polyhedrons

Consider the case in which both the vehicle shape and obstacle shape are convex, compact polyhedrons in $\mathbb{R}^n$ with $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ vertices respectively:

$$\mathcal{A} = \mathrm{co}(\{a_i \in \mathbb{R}^n, i \in [n_{\mathcal{A}}]\}),$$

$$\mathcal{B} = \mathrm{co}(\{b_j \in \mathbb{R}^n, j \in [n_{\mathcal{B}}]\}).$$

Note that for a single point $q \in \mathbb{R}^n$ we have $\mu_q(c) = \sigma_q(c) = c^T q$. Using Theorem 5.1 we obtain conditions to ensure a minimum signed distance of γ between two polyhedrons:

$$\begin{aligned}
&\alpha \leq c^T R(x_k) a_i, && i \in [n_{\mathcal{A}}] \\
&\beta \geq c^T S_k b_j, && j \in [n_{\mathcal{B}}] \\
&\gamma \leq \alpha - \beta + c^T(p(x_k) - d_k), \\
&1 = \|c\|
\end{aligned}$$

Remark. We contrast this with the method of [35] which assumes a halfspace-representation (vice vertex representation) of a compact polyhedron. Let $m_{\mathcal{A}}, m_{\mathcal{B}}$ be the number of linear constraints necessary to describe $\mathcal{A}, \mathcal{B}$ respectively. The dual approach introduces $(m_{\mathcal{A}} + m_{\mathcal{B}})$ variables and $(2 + n + m_{\mathcal{A}} + m_{\mathcal{B}})$ constraints to represent the signed distance constraint. Note we must have $m_{\mathcal{A}}, m_{\mathcal{B}} \geq n + 1$ for $\mathcal{A}, \mathcal{B}$ to be compact with non-empty interior. Our formulation introduces $(2 + n)$ variables and $(2 + n_{\mathcal{A}} + n_{\mathcal{B}})$ constraints. For the case in which $n_{\mathcal{A}} = m_{\mathcal{A}}, n_{\mathcal{B}} = m_{\mathcal{B}}$, our method introduces fewer variables and fewer constraints.[4]

5.4.2.2 Ellipsoids

Let the vehicle shape and obstacle shape be ellipsoids given by matrices $P, Q \in \mathbb{S}^n_{++}$:

$$\begin{aligned}
\mathcal{A} &= \{x \in \mathbb{R}^n \,|\, x^T P^{-1} x \leq 1\}, \\
\mathcal{B} &= \{x \in \mathbb{R}^n \,|\, x^T Q^{-1} x \leq 1\}.
\end{aligned}$$

Recall that ellipsoids have a closed-form cost and support function given by $\mu_{\mathcal{A}}(c) = -\sqrt{c^T P c}$ and $\sigma_{\mathcal{B}}(c) = \sqrt{c^T Q c}$. As $n_{\mathcal{A}} = 1$ and $n_{\mathcal{B}} = 1$, Theorem 5.1 reduces to Lemma 5.6 yielding:

$$\begin{aligned}
\gamma \leq &-\sqrt{c^T R(x_k) P R(x_k)^T c} - \sqrt{c^T S_k Q S_k^T c} \\
&+ c^T(p(x_k) - d_k), \\
1 = &\|c\|.
\end{aligned}$$

[4] A similar remark applies to the method of [37] which uses Farkas' Lemma. This requires introducing $(m_{\mathcal{A}} + m_{\mathcal{B}})$ variables and $(2(m_{\mathcal{A}} + m_{\mathcal{B}}) + 1)$ constraints.

Remark. The dual formulation in [35] uses second-order cone constraints to represent ellipsoids. Each second-order cone constraint introduces a dual variable pair $\lambda \in \mathbb{R}, u \in \mathbb{R}^n$ and the constraint $\lambda \geq \|u\|$. In total the dual formulation introduces $2(n+1)$ variables and $(4+n)$ constraints. Our formulation introduces n variables and two constraints.

Remark. The support function of an ellipsoid involves the square root, which is not differentiable at the origin. Given $P \succ 0$, the argument only evaluates to zero for $c = 0$ which cannot be a solution. However, we may experience issues if the solver is initialized with $c = 0$. We can add a small smoothing term $\epsilon > 0$ to address this case. Noting that $-\sqrt{c^T P c + \epsilon} < -\sqrt{c^T P c}$ it is seen that this modification is conservative in that satisfying the conditions of Lemma 5.6 means $\mu_{\mathcal{V}(x_k)}(c) - \sigma_{\mathcal{O}_k}(c) > \gamma$. The signed distance constraint is then strictly satisfied.

5.4.2.3 General Convex Sets

Let the vehicle and obstacle geometry be convex sets of the form

$$\mathcal{A} = \{x \in \mathbb{R}^n \,|\, p_i(x) \leq 0, i \in [n_{\mathcal{A}}]\} \tag{5.39}$$

$$\mathcal{B} = \{x \in \mathbb{R}^n \,|\, q_j(x) \leq 0, j \in [n_{\mathcal{B}}]\} \tag{5.40}$$

where $p_i : \mathbb{R}^n \to \mathbb{R}, i \in [n_{\mathcal{A}}]$ and $q_j : \mathbb{R}^n \to \mathbb{R}, j \in [n_{\mathcal{B}}]$ are smooth, convex functions. It can be shown that the cost and support functions in dual form are given by:

$$\mu_{\mathcal{A}}^{\text{dual}}(c) = \max_{x_{\mathcal{A}}, \lambda_{\mathcal{A}_i}} \{c^T x_{\mathcal{A}} + \sum_{i \in [n_{\mathcal{A}}]} \lambda_{\mathcal{A}_i} p_i(x_{\mathcal{A}}) \,|\, c + \sum_{i \in [n_{\mathcal{A}}]} \lambda_{\mathcal{A}_i} \nabla p_i(x_{\mathcal{A}}) = 0, \lambda_{\mathcal{A}_i} \geq 0, i \in [n_{\mathcal{A}}]\} \tag{5.41}$$

$$\sigma_{\mathcal{B}}^{\text{dual}}(c) = \min_{x_{\mathcal{B}}, \lambda_{\mathcal{B}_j}} \{c^T x_{\mathcal{B}} - \sum_{j \in [n_{\mathcal{B}}]} \lambda_{\mathcal{B}_j} q_j(x_{\mathcal{B}}) \,|\, c - \sum_{j \in [n_{\mathcal{B}}]} \lambda_{\mathcal{B}_j} \nabla q_j(x_{\mathcal{B}}) = 0, \lambda_{\mathcal{B}_j} \geq 0, j \in [n_{\mathcal{B}}]\} \tag{5.42}$$

Using Theorem 5.6, we obtain the following conditions which collectively ensure a minimum signed distance of γ.

$$\gamma \le \Big(c^T R(x) x_{\mathcal{A}} + \sum_{i \in [n_{\mathcal{A}}]} \lambda_i p_i(x_{\mathcal{A}})\Big) - \Big(c^T S x_{\mathcal{B}} - \sum_{j \in [n_{\mathcal{B}}]} q_j(x_{\mathcal{B}})\Big) + c^T(p(x) - d) \tag{5.43}$$

$$0 = R(x)^T c + \sum_{i \in [n_{\mathcal{A}}]} \lambda_{\mathcal{A}_i} \nabla p_i(x_{\mathcal{A}}) \tag{5.44}$$

$$0 \le \lambda_{\mathcal{A}_i}, i \in [n_{\mathcal{A}}] \tag{5.45}$$

$$0 = S^T c - \sum_{j \in [n_{\mathcal{B}}]} \lambda_{\mathcal{B}_j} \nabla q_j(x_{\mathcal{B}}) \tag{5.46}$$

$$0 \le \lambda_{\mathcal{B}_j}, j \in [n_{\mathcal{B}}] \tag{5.47}$$

$$1 = \|c\| \tag{5.48}$$

Remark. To our knowledge no other collision avoidance formulations address the case in which the geometries are of the form (5.39), (5.40). This formulation is useful for its generality. Given raw data such as a point cloud representing a shape, methods based on sum-of-squares optimization can be leveraged to turn the data into a model of the form (5.39) [13, 52].

5.4.2.4 Convex Sets with Linear and Second-Order Cone Constraints

Let the vehicle geometry be a convex set given by $n_{\mathcal{A}}$ linear constraints and $m_{\mathcal{A}}$ second-order cone constraints. Let the obstacle geometry be a convex set given by $n_{\mathcal{B}}$ linear constraints and $m_{\mathcal{B}}$ second-order cone constraints. The sets have the form

$$\mathcal{A} = \{x \in \mathbb{R}^n \,|\, 0 \le Cx + d, \|A_i x + b_i\| \le c_i^T x + d_i, i \in [m_{\mathcal{A}}]\}, \tag{5.49}$$

$$\mathcal{B} = \{x \in \mathbb{R}^n \,|\, 0 \le Gx + h, \|E_j x + f_j\| \le g_j^T x + h_j, j \in [m_{\mathcal{B}}]\}. \tag{5.50}$$

where $C \in \mathbb{R}^{n_{\mathcal{A}} \times n}, d \in \mathbb{R}^{n_{\mathcal{A}}}, A_i \in \mathbb{R}^{n \times n}, b_i \in \mathbb{R}^n, c_i \in \mathbb{R}^n, d_i \in \mathbb{R}, i \in [m_{\mathcal{A}}]$ and $G \in \mathbb{R}^{n_{\mathcal{B}} \times n}, h \in \mathbb{R}^{n_{\mathcal{B}}}, E_j \in \mathbb{R}^{n \times n}, f_j \in \mathbb{R}^n, g_j \in \mathbb{R}^n, h_j \in \mathbb{R}, j \in [m_{\mathcal{B}}]$.

For the dual form of $\mu_{\mathcal{A}}(c)$ we introduce dual variables $\lambda \in \mathbb{R}^{n_{\mathcal{A}}}$ for the linear constraints and $(\lambda_i, u_i) \in \mathbb{R} \times \mathbb{R}^n, i \in [m_{\mathcal{A}}]$ for the second-order cone constraints. Similarly, for the dual form of $\sigma_{\mathcal{B}}(c)$ we introduce dual variables $\xi \in \mathbb{R}^{n_{\mathcal{B}}}$ for the linear constraints and $(\xi_j, w_j) \in \mathbb{R} \times \mathbb{R}^n, j \in [m_{\mathcal{B}}]$ for the second-order cone constraints.

$$\mu_{\mathcal{A}}^{dual}(c) = \max_{\lambda,\lambda_i,u_i} \{-d^T\lambda + \sum_{i\in[m_{\mathcal{A}}]} b_i^T u_i - d_i\lambda_i \,|\, c - C^T\lambda + \sum_{i\in[m_{\mathcal{A}}]} A_i^T u_i - c_i\lambda_i = 0, \\ \lambda \geq 0, \lambda_i \geq \|u_i\|, i \in [m_{\mathcal{A}}]\} \tag{5.51}$$

$$\sigma_{\mathcal{B}}^{dual}(c) = \min_{\xi,\xi_j,w_j} \{h^T\xi - \sum_{j\in[m_{\mathcal{B}}]} f_j^T w_j + h_j\xi_j \,|\, c + G^T\xi - \sum_{j\in[m_{\mathcal{B}}]} E_j^T w_j + g_j\xi_j = 0, \\ \xi \geq 0, \xi_j \geq \|w_j\|, j \in [m_{\mathcal{B}}]\} \tag{5.52}$$

Using the dual forms in conjunction with Lemma 5.8 we can readily obtain differentiable signed distance constraints.

5.4.3 Continuous Collision Avoidance

Theorem 5.1 provides differentiable conditions for representing signed distance constraints between a vehicle and obstacle at discrete time steps $t_k, k \in \mathbb{Z}_+$. As the vehicle and obstacle transition between these discrete poses, the signed distance constraint may not be satisfied. This can be resolved by enforcing signed distance constraints with respect to the swept volume of the vehicle and obstacle over the time interval $t \in [t_k, t_{k+1}]$.

Assumptions 5.1 and 5.2 define outer approximations of the swept volume of the vehicle and obstacle respectively. These approximations utilize the convex hull and Minkowski sum operators. To account for the Minkowski sum operator we will make use of the following lemma.

Lemma 5.9. *Let $\mathcal{C}, \mathcal{D} \subset \mathbb{R}^n$ be closed convex sets. Let $\mathcal{C}$ and/or $\mathcal{D}$ be bounded. Let $r_{\mathcal{C}}, r_{\mathcal{D}} \in \mathbb{R}_{\geq 0}$. Then*

$$\mathrm{sd}(\mathcal{C} \oplus B_{r_{\mathcal{C}}}, \mathcal{D} \oplus B_{r_{\mathcal{D}}}) = \mathrm{sd}(\mathcal{C}, \mathcal{D}) - r_{\mathcal{C}} - r_{\mathcal{D}}. \tag{5.53}$$

The following relation results from applying Lemma 5.2 with Assumptions 5.1 and

5.2 followed by Lemma 5.9:

$$\begin{aligned} & \operatorname{sd}(\operatorname{sv}_{\mathcal{V},f}(x_k, u_k, t_k, t_{k+1}), \operatorname{sv}_{\mathcal{O}}(t_k, t_{k+1})) \geq \gamma \\ \Leftarrow & \operatorname{sd}(\operatorname{co}(\{\mathcal{V}(x_k), \mathcal{V}(x_{k+1})\}) \oplus B_{r(x_k,u_k)}, \operatorname{co}(\{\mathcal{O}_k, \mathcal{O}_{k+1}\}) \oplus B_{w_k}) = \gamma \\ \iff & \operatorname{sd}(\operatorname{co}(\{\mathcal{V}(x_k), \mathcal{V}(x_{k+1})\}), \operatorname{co}(\{\mathcal{O}_k, \mathcal{O}_{k+1}\})) = \gamma + r(x_k, u_k) + w_k. \end{aligned} \tag{5.54}$$

From this relation, we can extend Theorem 5.1 to obtain sufficient conditions for continuous collision avoidance.

Theorem 5.2. *Let the vehicle dynamics and geometry satisfy Assumption 5.1. Let the obstacle geometry satisfy Assumption 5.2. Let* $\mathcal{A} = \operatorname{co}(\{\mathcal{A}^{(i)}, i \in [n_{\mathcal{A}}]\})$ *and* $\mathcal{B} = \operatorname{co}(\{\mathcal{B}^{(j)}, j \in [n_{\mathcal{B}}]\})$ *where each* $\mathcal{A}^{(i)}, \mathcal{B}^{(j)}$ *is a convex set.*

Then $\operatorname{sd}(\operatorname{sv}_{\mathcal{V},f}(x_k, u_k, t_k, t_{k+1}), \operatorname{sv}_{\mathcal{O}}(t_k, t_{k+1})) \geq \gamma$ *if there exists* $c \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}$ *satisfying*

$$\begin{aligned} & \alpha \leq \mu_{\mathcal{A}^{(i)}}(R(x_k)^T c), && i \in [n_{\mathcal{A}}] \\ & \alpha \leq \mu_{\mathcal{A}^{(i)}}(R(x_{k+1})^T c) + c^T(p(x_{k+1}) - p(x_k)), && i \in [n_{\mathcal{A}}] \\ & \beta \geq \sigma_{\mathcal{B}^{(j)}}(S_k^T c), && j \in [n_{\mathcal{B}}] \\ & \beta \geq \sigma_{\mathcal{B}^{(j)}}(S_{k+1}^T c) + c^T(d_{k+1} - d_k), && j \in [n_{\mathcal{B}}] \\ & \gamma \leq \alpha - \beta + c^T(p(x_k) - d_k) - r(x_k, u_k) - w_k \\ & 1 = \|c\|. \end{aligned}$$

Proof. The stated conditions arise from applying the necessary and sufficient conditions of Theorem 5.1 to ensure a signed distance of $\gamma + r(x_k, u_k) + w_k$ between the sets $\operatorname{co}(\{\mathcal{V}(x_k), \mathcal{V}(x_{k+1})\})$ and $\operatorname{co}(\{\mathcal{O}_k, \mathcal{O}_{k+1}\})$. From (5.54), this is sufficient for ensuring a signed distance of γ between $\operatorname{sv}_{\mathcal{V},f}(x_k, u_k, t_k, t_{k+1})$ and $\operatorname{sv}_{\mathcal{O}}(t_k, t_{k+1})$. □

Remark. Theorem 5.2 is only sufficient because we are outer-approximating nonconvex swept volumes with convex sets. For example, in Figure 5-1, the right subplot shows an aggressive turn in which our outer approximation introduces noticeable conservatism. Here we are intentionally using a large integration step size ($\Delta T = 0.77s$)

to highlight this aspect. In practical applications, one can reduce the integration step size until this conservatism is acceptable.

5.5 Examples

We demonstrate our method using a car model navigating in $\mathbb{R}^2$ [35]. The vehicle state consists of positions (p_x, p_y), orientation (ψ), velocity (v), and steering angle (δ). The inputs are acceleration (a) and steering rate (s). The parameter $L = 2.7$ is the wheelbase. The continuous-time dynamics are:

$$\begin{aligned} \dot{p}_x &= v \cos \psi \\ \dot{p}_y &= v \sin \psi \\ \dot{\psi} &= v \frac{\tan \delta}{L} \\ \dot{v} &= a \\ \dot{\delta} &= s \end{aligned} \tag{5.55}$$

The vehicle's shape is a polyhedron $\mathcal{A} = \mathrm{co}(\{(\pm 2.5, \pm 1)\})$. The space occupied by the vehicle is given by

$$\mathcal{V}(x) = R(x)\mathcal{A} + p(x) \tag{5.56}$$

where

$$p(x) = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \; R(x) = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix}. \tag{5.57}$$

5.5.1 Swept Volume Approximation Model

The continuous collision avoidance conditions require a $\mathcal{C}^2$ function $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_{\geq 0}$ satisfying Assumption 5.1. We outline out a practical method for doing so. We first simulate the continuous dynamics over a time interval $t \in [0, \Delta T]$ where ΔT is the discrete time step used in the optimal control problem. We do this for various initial states $x_k^{(i)}$ and control inputs $u_k^{(i)}$ within expected ranges. For each sample $(x_k^{(i)}, u_k^{(i)})$

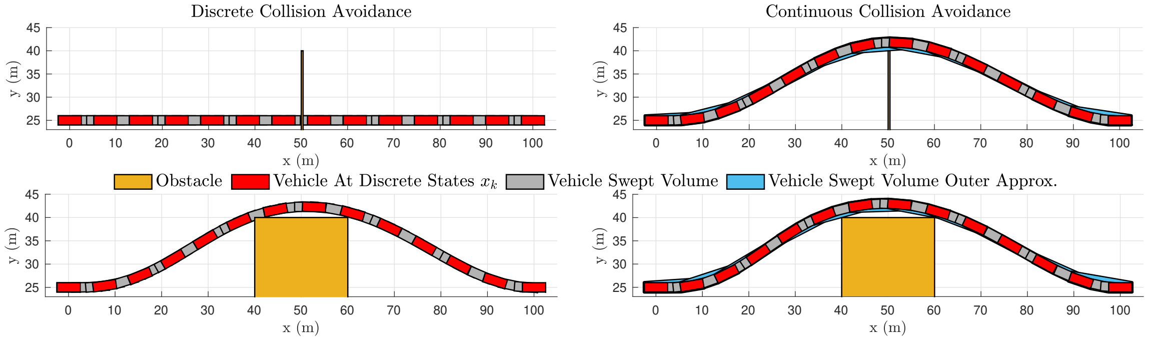


Figure 5-2. Autonomous car navigating obstacles. Discrete collision avoidance incorrectly passes through walls (upper left) and cuts corners (lower left). Continuous collision avoidance prevents these erroneous behaviors by checking collision with respect to an outer approximation of the swept volume.

we compute the convex hull of the resulting swept volume. We then compute the minimum radius $r^{(i)}$ such that $\text{co}(\{\mathcal{V}(x_k^{(i)}), \mathcal{V}(x_{k+1}^{(i)})\}) \oplus B_{r^{(i)}} \supseteq \text{co}(\text{sv}_{\mathcal{V},f}(x_k, u_k, 0, \Delta T))$. Finally we fit a non-negative function to the resulting data samples $\{x_k^{(i)}, u_k^{(i)}, r^{(i)}\}$. This can be done using sum-of-squares (SOS) optimization [16]. In our examples we utilized an 8th-order SOS polynomial $r(v_k, \delta_k)$ to represent the ball radius as a function of vehicle velocity and steering angle.

5.5.2 Results

5.5.2.1 Thin Wall

We first consider navigating around a thin wall and require $\text{sd}(\mathcal{V}(x), \mathcal{O}) \geq 0$. The vehicle begins at $(p_x = 0, p_y = 25, \psi = 0)$ and must end at $(p_x = 100, p_y = 25, \psi = 0)$ while minimizing the squared-norm of the control effort $l(X, U) = \|U\|_2^2$. We set the time horizon to 10s and use $N = 13$ steps, giving a discrete-time step of $\Delta T = \frac{10}{13}$. We use a 4th-order Runge-Kutta method to obtain the discrete dynamic model $x_{k+1} = f_{\Delta T}(x_k, u_k)$. We solve (5.19) using both the discrete collision avoidance conditions (Theorem 5.1) and the continuous collision avoidance conditions (Theorem 5.2). When solving with the discrete collision avoidance conditions, the trajectory passes through the wall in order to minimize the control effort. We note that other methods for optimization-based collision avoidance such as [35] are susceptible to this behavior. By utilizing the continuous collision avoidance conditions, the solver is prevented from exploiting the discrete approximation and returns a trajectory which successfully avoids the wall. Figure 5-2 shows the results in the upper subplots. We note that the outer approximation of the swept volume is only slightly conservative compared to the true swept volume as shown by the blue borders.

5.5.2.2 Corner Cutting

Another issue commonly faced by optimization-based motion planners is corner cutting. To demonstrate this, we replace the thin obstacle with a wide obstacle. Due to the velocity constraints on the vehicle, it is not possible for the discrete-time trajectory to “jump over" the obstacle. Instead, the discrete collision avoidance constraints yield a trajectory that turns to avoid the obstacle. However, it cuts the corner at $(40, 40)$ to minimize the necessary maneuvering. The continuous collision avoidance conditions again prevent this from happening. Figure 5-2 shows the results in the lower subplots.

5.5.2.3 Parallel Parking

We now consider a parallel parking problem from [35]. Figure 5-3 is reproduced from [35] which only considered discrete collision avoidance. Notably, although the discrete poses avoid the obstacles, it is clear that the continuous trajectory would cut the corner at (3,5). We solve this problem using a time-step of $\Delta T = 1.0s$ and time horizon of 40 seconds. Figure 5-4 shows the results with both discrete collision avoidance and continuous collision avoidance. As expected, the discrete collision avoidance trajectory cuts the corner at $(3, 5)$. In contrast, the continuous collision avoidance does not. Figure 5-5 provides a closer view.

Figure 5-6 shows the velocity and steering profiles for the 40-second parallel parking maneuver. The velocity profile of the discrete collision avoidance solution switches

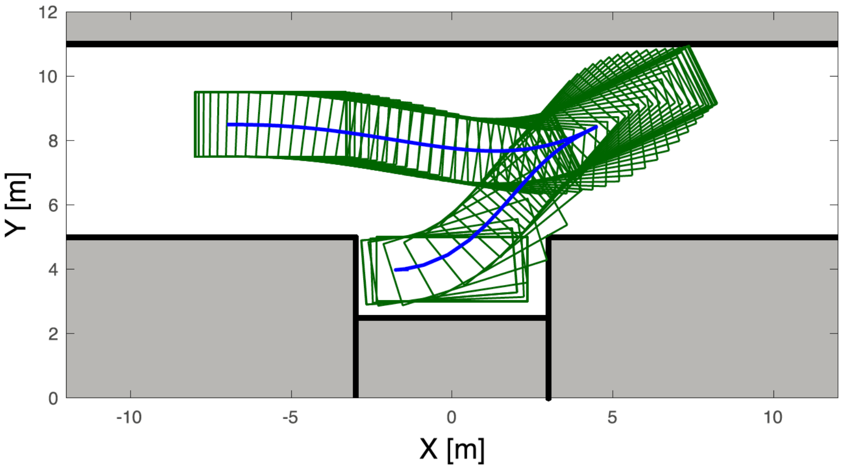


Figure 5-3. Parallel parking example reproduced from [35].

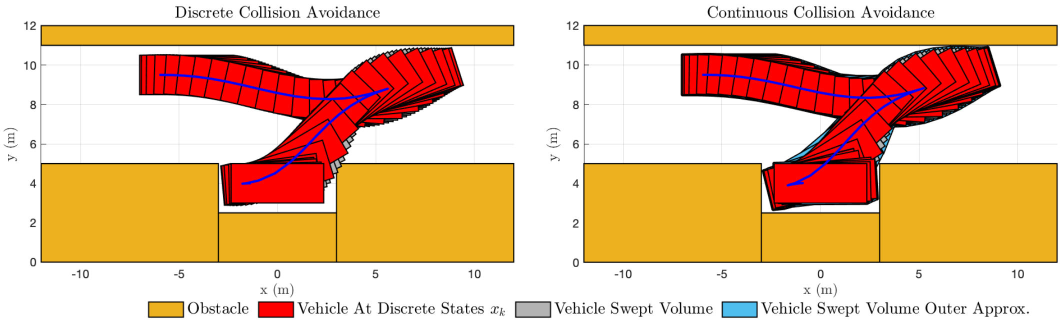


Figure 5-4. Parallel parking with discrete (left) and continuous (right) collision avoidance.

direction two times (i.e. the vehicle initially moves forward, then reverses, then moves forward again). In contrast, the continuous collision avoidance solution switches direction five times in order to avoid the obstacles. The latter is notable for its complexity. In this tightly constrained environment ensuring continuous collision avoidance cannot be achieved by simply turning more aggressively (note that the steering angle is already saturated for portions of the maneuver). Instead, the car must make multiple, small forward-reverse adjustments once it is approximately within the parking spot, similar to what a human driver would naturally do. This demonstrates the power of optimization-based motion planning for tackling difficult maneuvering situations for autonomous systems.

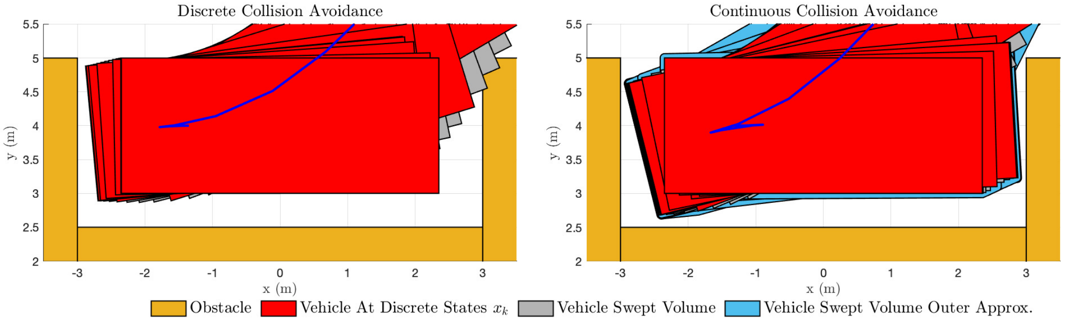


Figure 5-5. Parallel parking with discrete (left) and continuous (right) collision avoidance. Discrete collision avoidance clips the corner.

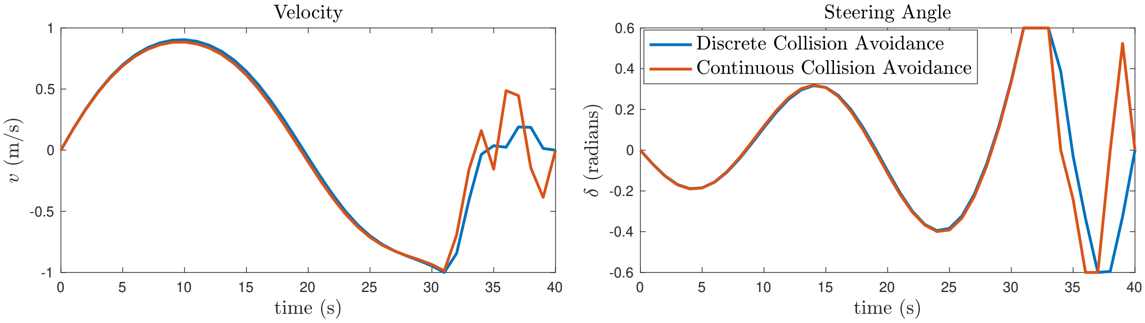


Figure 5-6. Steering and velocity profiles for parallel parking maneuver.

5.5.2.4 Constant Distance

Lastly, we briefly demonstrate the use of Lemma 5.7 for imposing a maximum distance constraint (vice just a lower bound). The vehicle begins at position ($p_x = -10, p_y = 3, \psi = \frac{\pi}{2}$) and must navigate to position ($p_x = 10, p_y = 3, \psi = \frac{3\pi}{2}$) while maintaining a constant distance of 6m from an obstacle. Figure 5-7 shows the resulting profile. The green line segments connect a pair of points (o, v) identified by the solver as a certificate of the minimum distance. We note that this pair is not necessarily unique. The right subplot shows the minimum distance during the 20s profile. As expected, the car maintains a constant distance of 6m to the obstacle.

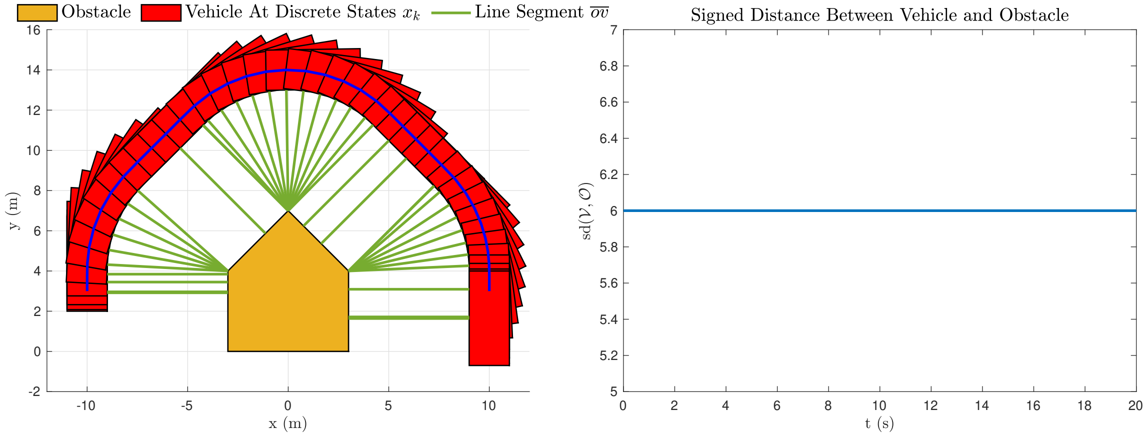


Figure 5-7. Maneuvering while maintaining a constant distance to an obstacle.

5.5.3 Implementation Details

All examples were solved on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 CPU. IPOPT [47] with the MA27 linear solver was used to solve the nonlinear optimization problems with exact gradients and Hessians supplied by CasADi [48].

5.6 Conclusion

A novel formulation of collision avoidance based on signed distance constraints was proposed for convex-shaped vehicles navigating convex obstacles. This formulation is continuously differentiable and therefore suitable for incorporation within optimization-based motion planning algorithms. For the important case of polyhedral and ellipsoidal shapes, this representation is more compact than existing formulations as it introduces fewer additional variables and constraints. Additionally, this formulation can be used to ensure the continuous-time trajectory satisfies the collision avoidance constraints despite being planned in a discrete setting. This provides a rigorous means of preventing “tunneling" and corner-cutting which can occur when collision avoidance is only enforced at discrete time steps. Further we provided a novel extension that allows one to impose upper bounds on the distance between a vehicle and an obstacle.

Chapter 6

Future Work

Chapter 5 developed a novel approach for representing signed distance constraints as a set of differentiable conditions. The method relied on using the support function representation of convex sets. Beyond signed distance constraints, variations of this approach can be used to address a number of challenging problems in optimization-based motion planning. We conclude this dissertation by briefly outlining some of these extensions.

6.1 Robust Motion Planning

In the optimization-based motion planning algorithms considered so far, the dynamics are assumed to be known exactly. However, there is typically a mismatch between the assumed model and that of reality. The dynamics model may be an approximation and/or there may be uncontrolled inputs arising from the environment (e.g. wind blowing on a quadcopter). We generically account for model mismatch by introducing a disturbance term w into our standard dynamic model

$$\dot{x} = f(x, u, w) \tag{6.1}$$

along with the constraint $w \in \mathcal{W}$ where $\mathcal{W}$ is the set of all possible disturbances. Without loss of generality, we can assume this set contains the origin and we refer to the case $w = 0$ as the nominal (undisturbed) model.

Traditional optimization-based planning algorithms do not account for the impact of these disturbances. Instead, it is assumed that closed-loop feedback control is used to compensate for disturbances. Heuristics are used in the planning stage to obtain solutions that provide enough operating margin for the controllers to successfully reject disturbances. For example, it is common to use objectives that penalize the control effort utilized. This indirectly keeps the control commands away from their limits so the lower-level feedback control can make small adjustments to account for disturbances.

Robust motion planning algorithms explicitly account for disturbances. One popular method is DIRTREL (DIRect TRanscription with Ellipsoidal disturbances and Linear feedback) [53]. DIRTREL accounts for the impact of disturbances by propagating them through linearized approximations of the system dynamics in a manner similar to an extended Kalman filter. This yields ellisoidal approximations of the system state distribution at a given time index k,

$$x_k \in \bar{x}_k + \{\delta x \,|\, \delta x^T E^{-1} \delta x \leq 1\} \tag{6.2}$$

where $\bar{x}_k \in \mathbb{R}^n$ is the nominal state and $E \in \mathbb{S}^n_{++}$ defines the uncertainty. The main contribution of DIRTREL is a method for generating the uncertainty matrices E within a motion planning algorithm.

DIRTREL does not provide a means to ensure that the constraints are satisfied for all possible realizations of the uncertainty. Instead, the authors propose sampling from this uncertainty by extracting the extreme values of the ellipsoid which are given by the columns of the matrix square root

$$x_k \in \bar{x}_k \pm (\text{col } E^{\frac{1}{2}}). \tag{6.3}$$

Robust motion planning is then approximately achieved by ensuring that the constraints are satisfied at the given sample points. This approach has two main drawbacks. First, other realizations of the uncertainty (beyond the extreme points) may still violate the constraints. Second, extracting these extreme points requires embedding a differentiable representation of the matrix square root operation within a nonlinear program. This is computationally intensive and the available methods (e.g. Denman-Beavers, Babylonian) are not guaranteed to converge.

In motion planning problems, most state constraints of practical interest can be posed as ensuring $x \in \mathcal{G}$ or $x \notin \mathcal{G}$ where $\mathcal{G}$ is a convex set. Under uncertainty, the point x becomes a set $\mathcal{X}$ and robust motion planning requires ensuring $\mathcal{X} \subseteq \mathcal{G}$ or $\mathcal{X} \cap \mathcal{G} = \emptyset$. If $\mathcal{X}$ and $\mathcal{G}$ are convex sets, we can use the methods of chapter 5 to represent these set constraints in a differentiable manner. For example, we can exactly represents the ellipsoidal uncertainty of DIRTREL instead of approximating it with samples obtained from the matrix square root operation.

To briefly demonstrate this we consider motion planning of a Dubin's car model navigating around two circular obstacles. The vehicle is subject to disturbances which push it off the nominal trajectory planned. Figure 6-1 shows the trajectory obtained with DIRTREL. The blue line shows the nominal trajectory. The red ellipsoids represented the uncertainty bounds at each discrete time index. DIRTREL ensures the extreme points (green dots) of these ellipsoid do not touch the obstacle.[1] However, other realizations of the uncertainty may still violate the constraints. We simulate the system with 100 different realizations of the uncertainty and plot the continuous time trajectories in gray. Of these samples, 10 instances violate the constraints (touch the yellow obstacles).

Figure 6-2 shows the same example but exactly representing the ellipsoidal uncertainty sets using the methods of chapter 5. Similar to our method for continuous collision avoidance, we approximately ensure robustness between discrete time points by enforcing constraints using the convex hull of neighboring ellipsoids. This yields the connected tube shown in red. We simulate the system with the same 100 realizations

[1] The system has 3 states (x, y, θ) so our 2D plots of (x, y) show 6 extreme points per ellipsoid.

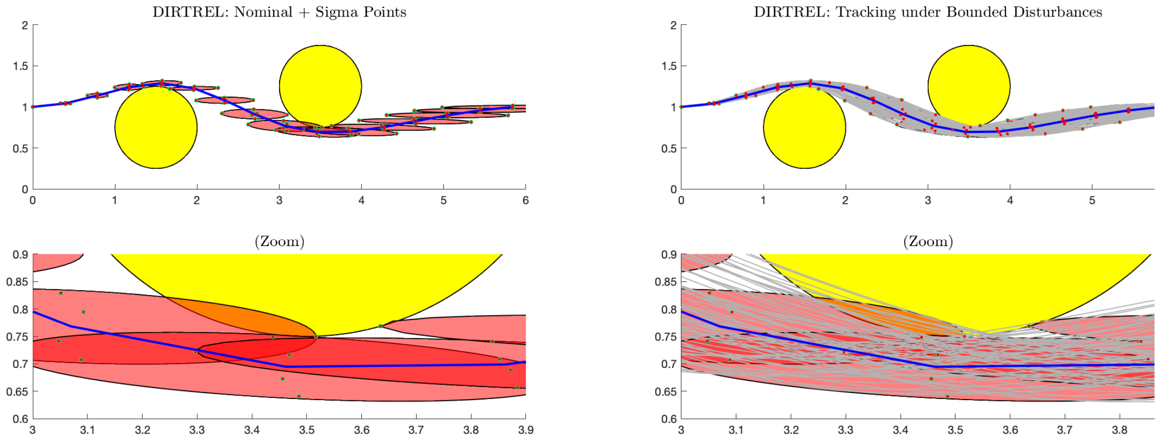


Figure 6-1. DIRTREL with sampled ellipsoid approximation.

of the uncertainty. All samples satisfy the constraints (no trajectories touch the yellow obstacles). Further, by eliminating the problematic Denman-Beavers algorithm we obtain a more efficient nonlinear program. The original DIRTREL formulation took 5.3 seconds to solve. With the revised formulation this is reduced to 0.7s using the same solver settings. We note that Denman-Beavers requires matrix inversion operations which prevent us from using the fast scalar symbolic math (SX variables) within the automatic differentiation tool CasADi [48]. Instead we must use the slower, matrix symbolic math (MX variables). Our formulation allows us to use either. Using SX variables, the computation time is further reduced from 0.7s down to 130ms, a 40x-speedup relative to the original DIRTREL formulation.

6.2 Bilevel Optimization with Convex Subproblems

Bilevel optimization refers to hierarchical optimization problems consisting of two levels. An upper-level optimization problem (the leader) determines the value of decision variable x. Based on the value of x a lower-level optimization problem (the follower) determines the value of decision variable y. These variables are linked through the objectives $F(x, y)$, $f(x, y)$ and “coupling constraints” $G(x, y)$ and $g(x, y)$.

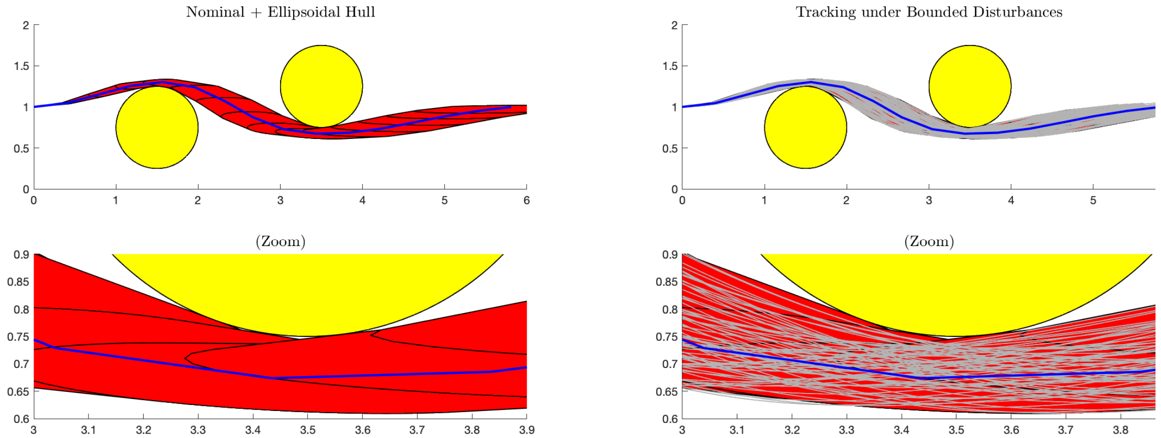


Figure 6-2. DIRTREL with exact ellipsoid representation.

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \quad & F(x,y) \\ \text{s.t.} \quad & \\ & G(x,y) \leq 0 \\ & y \in \arg\min\{f(x,y) : g(x,y) \leq 0\} \end{aligned} \tag{6.4}$$

Many subtle variations exist in the bilevel literature. Here we are showing the most common formulation. If the minimizing argument of the lower-level problem is non-unique, the upper-level optimization problem is free to choose the preferred value of y. For this reason, this formulation is sometimes referred to as "optimistic" bilevel optimization.

Solving bilevel problems typically requires reformulating the lower-level problem to obtain a "single-level" problem which is accepted by optimization solvers. In many applications, the lower-level problem is convex. Under appropriate constraint qualifications (e.g. Slater's), the implicitly-defined lower-level problem can be equivalently represented by the Karush-Kuhn-Tucker (KKT) conditions. This requires introducing additional decision variables representing the Lagrange multipliers. This is by far the most common reformulation.

Besides the algebraic conditions of optimality given by the KKT conditions, we can also define optimal points geometrically in terms of a supporting hyperplane. Let f be a differentiable convex function and $\mathcal{Z} = \{z \,|\, g(z) \leq 0\}$ be a convex set. Then

$$z^* \in \arg\min\{f(z) : g(z) \leq 0\} \iff z^* \in \mathcal{Z}, \nabla f(z^*)^T(z - z^*) \geq 0 \, \forall z \in \mathcal{Z}. \tag{6.5}$$

Here $-\nabla f(z^*)$ defines a supporting hyperplane to the feasible set at z^* [34]. Note the following trivial relation

$$\nabla f(z^*)^T(z - z^*) \geq 0 \, \forall z \in \mathcal{Z} \iff \nabla f(z^*)^T z^* \leq \min_{z \in \mathcal{Z}} \nabla f(z^*)^T z. \tag{6.6}$$

The right-hand-side of this expression involves the cost function that was introduced in chapter 5

$$\mu_{\mathcal{Z}}(c) := \min_{z \in \mathcal{Z}} c^T z. \tag{6.7}$$

There we developed differentiable representations of this cost function for a variety of convex sets. We can utilize this same machinery to reformulate our convex lower-level problem into algebraic conditions. Letting $\mathcal{Y}(x) := \{y \,|\, g(x, y) \leq 0\}$ we obtain

$$y^* \in \arg\min\{f(x, y) : g(x, y) \leq 0\} \iff y^* \in \mathcal{Y}(x), \nabla f(x, y^*)^T y^* \leq \mu_{\mathcal{Y}(x)}(\nabla f(x, y^*)). \tag{6.8}$$

The latter can be represented using the methods of chapter 5. The resulting expressions are distinct from those obtained from the KKT conditions. Appealingly, for simple convex sets such as ellipsoids and polyhedrons, no dual variables are required leading to compact representations. Surprisingly, a recent survey of methods for solving bilevel problems with convex lower-level problems makes no mention of this formulation [54].

Bilevel problems arise naturally in economics and operations research. More recently their use has become popular in planning and control of robotics [55]. Challenging physical properties such as friction and other non-smooth mechanics can be represented implicitly by an optimization problem. Similarly, the actions of other intelligent agents in a scenario may be defined by an optimization problem. Modeling these implicitly-defined systems within a motion planning problem leads to a bilevel formulation. We briefly demonstrate these formulations using the optimality conditions given by (6.8).

6.2.1 Modeling of Pursuit-Evasion

Consider the classic problem of pursuit-evasion between two agents. The evader wishes to reach a target while ensuring a pursuer does not intercept them prior to arriving at the target. One simple model of the pursuer's behavior would be to minimize the distance of its position $p_k^P \in \mathbb{R}^n$ to the evader's position $p_k^E \in \mathbb{R}^n$ at time index k. We assume the pursuer can only move a finite distance between discrete time steps, $p_k^P - p_{k-1}^P \in \mathcal{R}$, where $\mathcal{R} \subseteq \mathbb{R}^n$ is a convex set. This leads to a simple projection representation of the pursuer's behavior

$$p_k^P \in \arg\min\{\|p_k^P - p_k^E\| : p_k^P - p_{k-1}^P \in \mathcal{R}\}. \tag{6.9}$$

For example, assume that $\mathcal{R} = \{y \,|\, ||y|| \leq r\}$. Using the results of chapter 5 along with (6.8), we can model the pursuer's behavior at time index k as satisfying

$$\|p_k^P - p_{k-1}^P\| \leq r \tag{6.10}$$

$$(p_k^P - p_k^E)^T p_k^P \leq -r\sqrt{(p_k^P - p_k^E)^T (p_k^P - p_k^E)}. \tag{6.11}$$

We utilize this formulation to solve a basic pursuit-evasion problem. The evader is modeled as a Dubin's car which must travel from $(0,0)$ to $(10,0)$ over a 2-second time horizon. It must ensure a given minimum distance from a pursuer that is initially located at $(10,0)$ and moves according to the guidance law (6.9). We model the dynamics using a time step of $0.02s$. The pursuer is able to move 0.1m per time step such that $\mathcal{R} = \{y \,|\, \|y\| \leq 0.1\}$ in (6.9). The evader's objective is to minimize the control effort utilized to evade the pursuer while still reaching the target. We pose this as a numerical optimal control problem using the constraints (6.10) and (6.11) to model the pursuer's implicitly-defined guidance law. Figure 6-3 shows the resulting evasive maneuvers that ensure minimum distances of 1 and 3 to the pursuer.

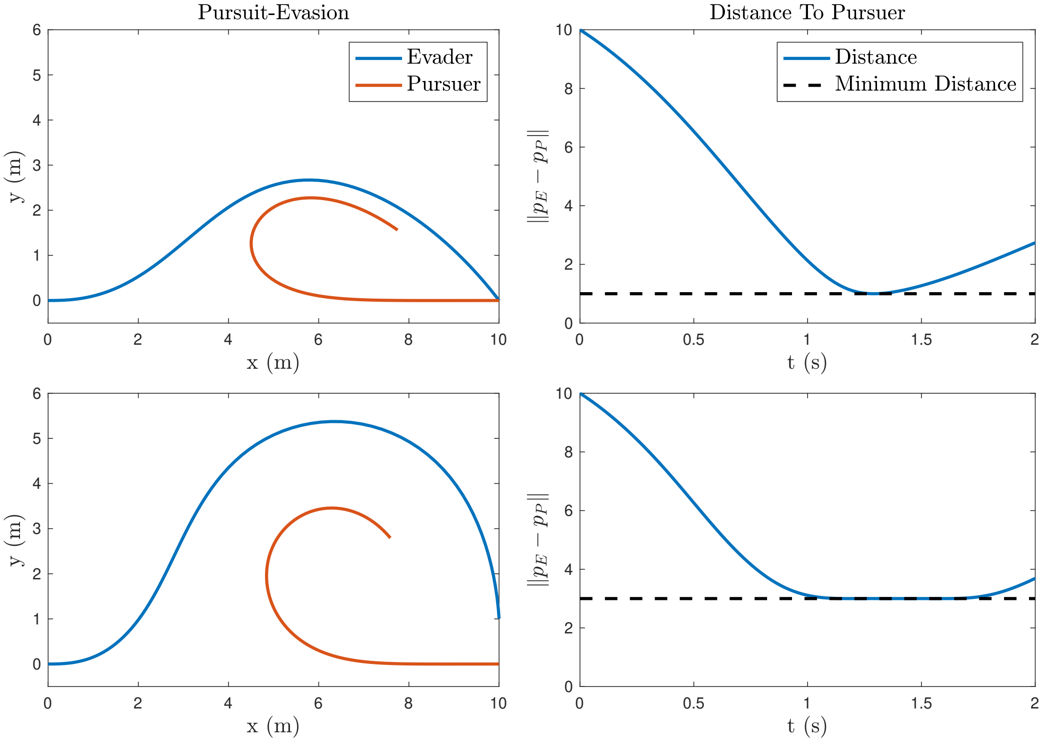


Figure 6-3. Minimum effort evasion for ensuring distance of 1 (upper) and 3 (lower) to pursuer.

6.2.2 Modeling the Pusher-Slider System

The pusher-slider system is a popular model for studying planar pushing of objects through frictional contacts. The system consists of a planar object with position (x, y) and orientation θ. A pusher makes contact with one edge of the object along which it can slide. The position of the pusher along its sliding axis is given by p_y. The pusher can apply a non-negative normal force f_n in the body-frame of the object along with a tangential force f_t that is constrained by a friction cone

$$-\mu_p f_n \leq f_t \leq \mu_p f_n \tag{6.12}$$

where $\mu_p > 0$ is the coefficient of friction. The pusher can slide ($\dot{p}_y \neq 0$) only when the tangential force is at the boundary of the friction cone as shown by Figure 6-4. For further details of the model we refer the reader to [56].

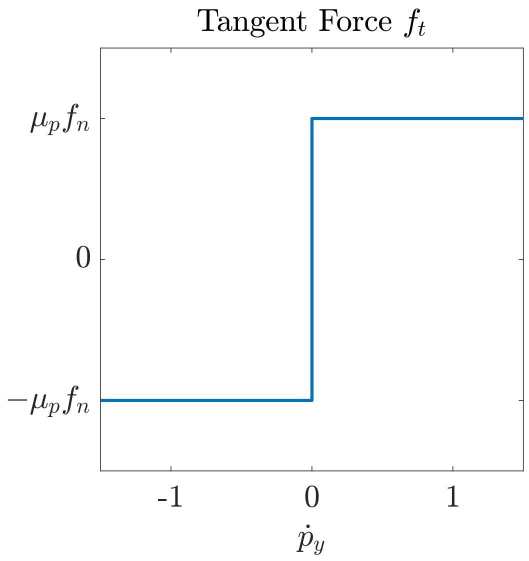


Figure 6-4. Friction model of pusher-slider.

The friction model in Figure 6-4 can be represented as

$$f_t \in \arg\min\{-\dot{p}_y f_t : -\mu_p f_n \leq f_t \leq \mu_p f_n\}. \tag{6.13}$$

This is a linear program parameterized by f_n and $\dot{p}_y$. Using the results of chapter 5 along with (6.8), we can represent f_t equivalently as belonging to the set of points satisfying

$$-\dot{p}_y f_t \leq -\dot{p}_y \mu_p f_n \tag{6.14a}$$

$$-\dot{p}_y f_t \leq \dot{p}_y \mu_p f_n \tag{6.14b}$$

$$-\mu_p f_n \leq f_t \leq \mu_p f_n. \tag{6.14c}$$

Alternatively we could equivalently represent f_t in terms of the KKT conditions for the linear program. However this yields a larger set of constraints along with introducing dual variables associated with the inequalities.

We pose a problem of moving an object from $(0, 0)$ to $(0, 0.4)$ while avoiding three obstacles. Obstacle avoidance conditions are enforced using the method of Chapter 5. We use a time step of 0.1s and a time horizon of 5s. With the friction model represented by (6.14) we obtain a solution in 3.3s using IPOPT. With the

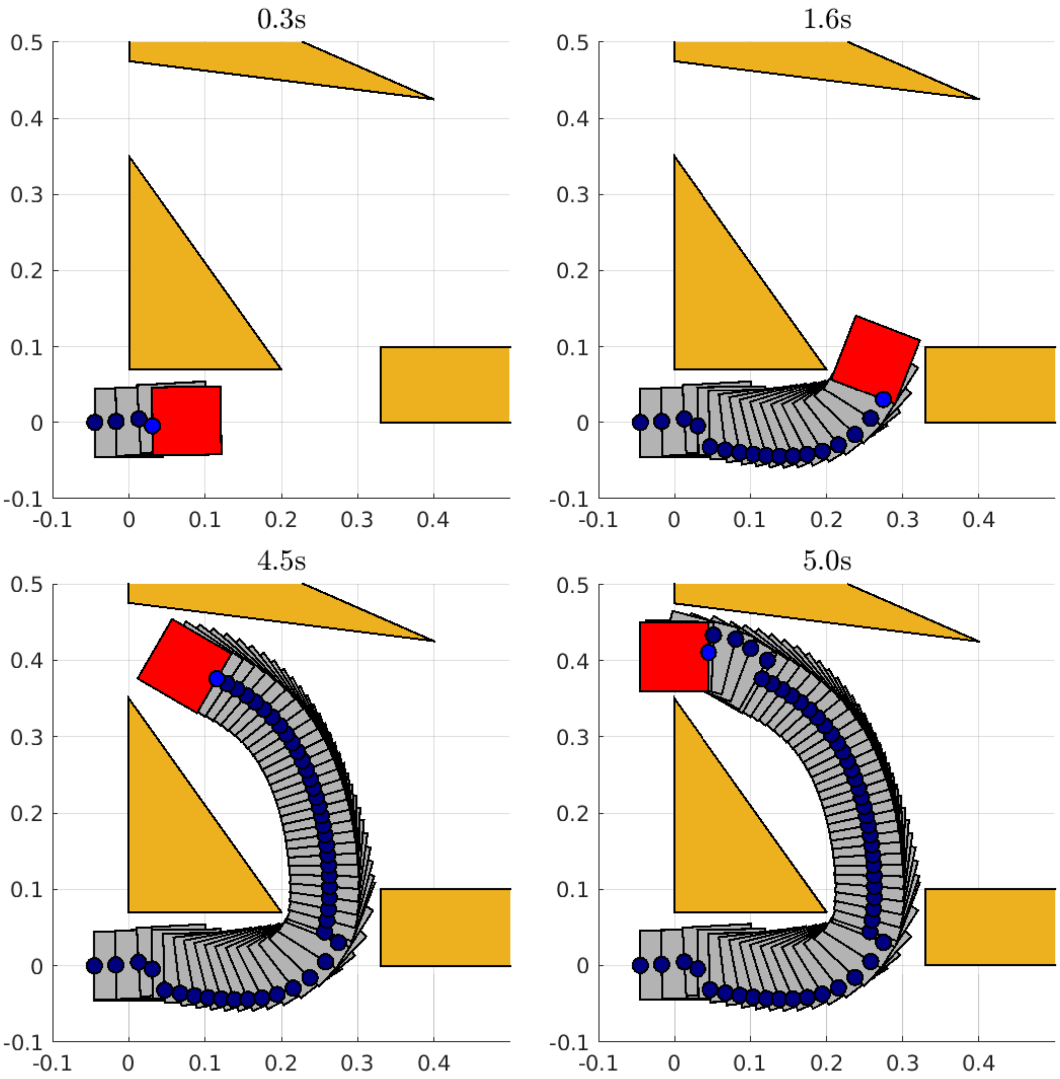


Figure 6-5. Pusher-slider system moving block from $(0, 0)$ to $(0, 0.4)$ over 5 seconds.

friction model represented in terms of the KKT conditions IPOPT failed to converge. Figure 6-5 shows the resulting trajectory with the friction modeled based on the proposed reformulation. Figure 6-6 shows the forces applied and slider position. For the tangential force we also plot the bounds given by the friction cone. During time intervals in which the pusher position changes the tangential force is on the boundary of the friction cone as expected.

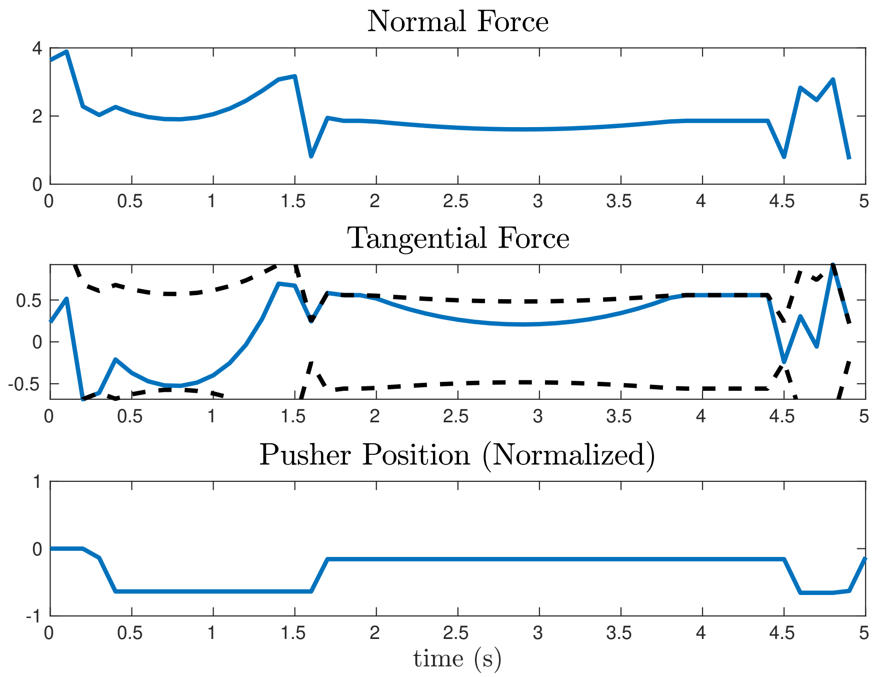


Figure 6-6. Pusher-slider force and position profile for moving block from $(0, 0)$ to $(0, 0.4)$ over 5 seconds.

6.3 Conclusion

This dissertation developed new methods for representing various semialgebraic sets that arise in planning and control of autonomous systems. Chapters 2-4 relied on sum-of-squares optimization to find inner or outer approximations of sets of interest. In contrast, Chapter 5 utilized the support function representation of sets to obtain exact reformulations of signed distance constraints between convex sets. In this closing chapter it was shown how this method can be extended to address motion planning when uncertainty is present or the dynamics are defined implicitly by an optimization problem. We plan to further pursue these directions in future work.

Appendix A

Proofs

A.1 Proof of Lemma 2.3

Proof. Assume $1 < s^\star < s$ satisfies $\mathcal{F} \subseteq \mathcal{X} \subseteq s^\star \mathcal{F}$. Let $x, sx \in \mathcal{X}, x \neq 0$ such that $tx \notin \mathcal{X} \, \forall \, t \in (1, s)$. Given $sx \in \mathcal{X} \implies sx \in s^\star \mathcal{F} \implies \frac{s}{s^\star} x \in \mathcal{F}$. However, $1 < \frac{s}{s^\star} < s \implies \frac{s}{s^\star} x \notin \mathcal{F}$, a contradiction. Thus $s^\star \geq s$.

□

A.2 Proof of Lemma 2.4

Proof. Recall the Hausdorff distance between two compact, convex sets can be written in terms of their support functions.

$$\begin{aligned} d_H(s\mathcal{F}, \mathcal{F}) &= \max_{c \in S^{n-1}} |\sigma_{s\mathcal{F}}(c) - \sigma_{\mathcal{F}}(c)| && \text{(A.1)} \\ &= \max_{c \in S^{n-1}} |s\sigma_{\mathcal{F}}(c) - \sigma_{\mathcal{F}}(c)| && \text{(A.2)} \\ &= (s-1) \cdot \max_{c \in S^{n-1}} \sigma_{\mathcal{F}}(c) && \text{(A.3)} \\ &= (s-1) \cdot \max_{x \in \mathcal{F}} \|x\|_2. && \text{(A.4)} \end{aligned}$$

□

A.3 Proof of Lemma 2.5

Proof. For convenience, define the following:

$$\mathcal{H} := \{p \,|\, \nabla g_i(q)^T(p-q) \leq 0 \,\forall\, q \in \partial\mathcal{X}_i, i \in [m]\}.$$

We show that $\ker\mathcal{X} \subseteq \mathcal{H}$ and $\ker\mathcal{X} \supseteq \mathcal{H}$ and therefore $\ker\mathcal{X} = \mathcal{H}$.

$\Rightarrow$ ($\ker\mathcal{X} \subseteq \mathcal{H}$): Assume $p \in \ker\mathcal{X}$ but there exists a point $q \in \partial\mathcal{X}_i$ for some $i \in [m]$ such that $\nabla g_i(q)^T(p-q) > 0$. Recall the definition of the directional derivative:

$$\lim_{t\to 0} \frac{g_i(tp + (1-t)q) - g_i(q)}{t} = \nabla g_i(q)^T(p-q).$$

Given $g_i(q) = 1$ and $\nabla g_i(q)^T(p-q) > 0$ implies there exists an open interval $t \in (0,\alpha), \alpha > 0$ in which $g_i(tp + (1-t)q) > 1$. The line segment over this open interval does not belong to $\mathcal{X}$. Thus $p \notin \ker\mathcal{X}$, a contradiction.

$\Leftarrow$ ($\ker\mathcal{X} \supseteq \mathcal{H}$): Let $p \in \mathcal{H}$. Assume $p \notin \ker\mathcal{X} \implies \exists\, q \in \mathcal{X}$ such that $l(t) \notin \mathcal{X}$ for some $t \in (0,1]$ where $l(t) := tp + (1-t)q$.[1] As $\mathcal{X}$ is compact, $l(t) \notin \mathcal{X} \implies g_i(l(t)) > 1$ for some $i \in [m]$ and open interval $t \in (a,b)$ satisfying $0 \leq a < b$ with $a < 1$. Without loss of generality, let $a = 0$ such that $q \in \partial\mathcal{X}_i$ and $g_i(l(0)) = 1$. Applying the definition of the directional derivative yields:

$$\lim_{t\to 0} \frac{g_i(l(t)) - g_i(l(0))}{t} = \nabla g_i(q)^T(p-q).$$

The left-hand side of this relation is non-negative. The right-hand side is non-positive per the definition of $\mathcal{H}$. Thus both sides must equal zero. As $\nabla g_i(q) \neq 0$, this implies

$$(p-q) \perp \nabla g_i(q). \tag{A.5}$$

Assume w.l.o.g. that $\nabla g_i(q)$ is aligned with coordinate n:

$$\nabla g_i(q) = \begin{bmatrix} 0_{n-1}^T & r \end{bmatrix}^T, r > 0. \tag{A.6}$$

[1] We have not yet shown that $\mathcal{H} \subseteq \mathcal{X}$ so we are not assuming $p \in \mathcal{X}$.

If this does not hold we can introduce an appropriate change of variables. Together, (A.5) and (A.6) $\implies (p_n - q_n)r = 0 \implies l_n(t) = q_n$. From this we have

$$l(t) = \begin{bmatrix} tp_{[n-1]} + (1-t)q_{[n-1]} \\ q_n \end{bmatrix}. \tag{A.7}$$

Define the following parameterized curve $\phi : \mathbb{R} \to \mathbb{R}^n$ which moves along the boundary $g_i(x) = 1$, starting from p:

$$\phi(t) = \begin{bmatrix} tp_{[n-1]} + (1-t)q_{[n-1]} \\ h(tp_{[n-1]} + (1-t)q_{[n-1]}) \end{bmatrix}. \tag{A.8}$$

Given $\frac{\partial g_i}{\partial x_n}(q) \neq 0$, from the implicit function theorem there exists an open set $U \subset \mathbb{R}^{n-1}$ with $q_{[n-1]} \in U$ and C^1 function $h : U \to \mathbb{R}$ such that $h(q_{[n-1]}) = q_n$ and $g_i(x_{[n-1]}, h(x_{[n-1]})) = 1$ for all $x_{[n-1]} \in U$. Here we are restricting coordinates $x_{[n-1]}$ to the line segment parameterized by t. Thus $g_i(\phi(t)) = 1$ for all t such that $\phi_{[n-1]}(t) \in U$. Let $t \in (-c,\, d), c > 0, d > 0$ denote this interval.

The line $l(t)$ and curve $\phi(t)$ only differ in coordinate n. Given $g_i(l(t)) > 1, t \in (0, b)$ and $g_i(\phi(t)) = 1, t \in (-c, d) \implies q_n \neq \phi_n(t) \, \forall \, t \in (0, \min(b, d))$.

Given $\frac{\partial g_i}{\partial x_n}(q) > 0 \implies \frac{\partial g_i}{\partial x_n} > 0$ for some open ball around q as g_i is smooth. Assuming $\phi_n(t) > q_n \implies g_i(\phi(t)) > g_i(l(t)) > 1$ for points sufficiently close to q, a contradiction. Thus $\phi_n(t) < q_n$ for some interval $t \in (0, e), e > 0$. From this we have

$$\frac{\partial g_i(\phi(t))}{\partial x_n}(q_n - \phi_n(t)) > 0, \forall t \in (0, e). \tag{A.9}$$

Given $q_n = \phi_n(0) > \phi_n(t)$ for some interval $t \in (0, e)$, by the mean value theorem there exists $t_\star \in (0, e)$ such that $\frac{d\phi_n}{dt}(t_\star) < 0$. This yields the following relation:

$$\frac{\partial g_i(\phi(t_\star))}{\partial x_n} \frac{d\phi_n(t_\star)}{dt} < 0. \tag{A.10}$$

Given $g_i(\phi(t)) = 1 \, \forall \, t \in (-c, d) \implies \frac{dg_i}{dt}(\phi(t)) = 0$. We expand this at the point $t_\star$ obtaining

$$\begin{aligned} 0 &= \frac{\partial g_i(\phi(t_\star))}{\partial x_{[n-1]}}^T \frac{d\phi_{[n-1]}(t_\star)}{dt} + \frac{\partial g_i(\phi(t_\star))}{\partial x_n} \frac{d\phi_n(t_\star)}{dt} \\ &= \frac{\partial g_i(\phi(t_\star))}{\partial x_{[n-1]}}^T (p_{[n-1]} - q_{[n-1]}) + \frac{\partial g_i(\phi(t_\star))}{\partial x_n} \frac{d\phi_n(t_\star)}{dt}. \end{aligned} \tag{A.11}$$

From equations (A.10) and (A.11) we obtain

$$\frac{\partial g_i(\phi(t_\star))}{\partial x_{[n-1]}}^T (p_{[n-1]} - q_{[n-1]}) > 0. \tag{A.12}$$

Finally, we evaluate the stated constraint on $p \in \mathcal{H}$ at the boundary point $\phi(t_\star)$ giving

$$\begin{aligned} \nabla g_i(\phi(t_\star))^T(p - \phi(t_\star)) = \frac{\partial g_i(\phi(t_\star))}{\partial x_n}(p_n - \phi_n(t_\star)) + \\ \frac{\partial g_i(\phi(t_\star))}{\partial x_{[n-1]}}^T (p_{[n-1]} - q_{[n-1]})(1 - t_\star). \end{aligned} \tag{A.13}$$

From (A.9) and (A.11) and noting that $(1 - t_\star) > 0$ and $q_n = p_n$ gives

$$\nabla g_i(\phi(t_\star))^T(p - \phi(t_\star)) > 0. \tag{A.14}$$

Thus $p \notin \mathcal{H}$, a contradiction.

□

A.4 Proof of Lemma 2.6

$\ker(\mathcal{A} \cap \mathcal{B}) \supseteq \ker\mathcal{A} \cap \ker\mathcal{B}$: Let $l(x, y) = \{\lambda x + (1 - \lambda)y \,|\, \lambda \in [0, 1]\}$ for some $x \in \ker\mathcal{A} \cap \ker\mathcal{B}$ and $y \in \mathcal{A} \cap \mathcal{B}$. As $x \in \ker\mathcal{A}, y \in \mathcal{A} \implies l(x, y) \subseteq \mathcal{A}$ and similarly, $x \in \ker\mathcal{B}, y \in \mathcal{B} \implies l(x, y) \subseteq \mathcal{B}$, we see that $x \in \ker(\mathcal{A} \cap \mathcal{B})$.□

$\ker(\mathcal{A} \cup \mathcal{B}) \supseteq \ker\mathcal{A} \cap \ker\mathcal{B}$: Let $l(x, y) = \{\lambda x + (1 - \lambda)y \,|\, \lambda \in [0, 1]\}$ for some $x \in \ker\mathcal{A} \cap \ker\mathcal{B}$ and $y \in \mathcal{A} \cup \mathcal{B}$. For the case when $y \in \mathcal{A}$, then $x \in \ker A \implies l(x, y) \subseteq \mathcal{A} \implies l(x, y) \subseteq \mathcal{A} \cup \mathcal{B}$. Similarly, for the case when $y \in \mathcal{B}$, then $x \in \ker B \implies l(x, y) \in \mathcal{B} \implies l(x, y) \subseteq \mathcal{A} \cup \mathcal{B}$. Therefore $x \in \ker(\mathcal{A} \cup \mathcal{B})$. □

Remark. Note that there is no relation between $\ker(\mathcal{A} \cap \mathcal{B})$ and $\ker(\mathcal{A} \cup \mathcal{B})$ in general. We gives examples in which one set is a subset of the other.

$\ker(\mathcal{A} \cup \mathcal{B}) \supset \ker(\mathcal{A} \cap \mathcal{B})$: Let $\mathcal{A} \setminus \mathcal{B} \neq \emptyset$ and $\mathcal{B} \setminus \mathcal{A} \neq \emptyset$. Let $\mathcal{A} \cup \mathcal{B}$ be a convex set. Then $\ker(\mathcal{A} \cup \mathcal{B}) = \mathcal{A} \cup \mathcal{B} \supset (\mathcal{A} \cap \mathcal{B}) \supseteq \ker(\mathcal{A} \cap \mathcal{B})$.

$\ker(\mathcal{A} \cup \mathcal{B}) \subset \ker(\mathcal{A} \cap \mathcal{B})$: Let $\mathcal{A}$ be a compact set that is not star-convex with non-empty interior. Let $\mathcal{B}$ be a non-empty convex set satisfying $\mathcal{B} \subset \mathcal{A}$. Then $\ker(\mathcal{A} \cap \mathcal{B}) = \mathcal{B} \supset \emptyset = \ker(\mathcal{A} \cup \mathcal{B})$.

A.5 Proof of Lemma 5.1

Proof. We prove this for the $\alpha < \beta$ case. The $\alpha > \beta$ and $\alpha = \beta$ cases can be shown using similar arguments. Let $t = (\beta - \alpha)c$ and $x \in \mathcal{H}^+ \implies c^T(x+t) \geq \alpha + (\beta - \alpha) = \beta \implies x + t \notin (\mathcal{H}^- \setminus \partial\mathcal{H}^-) \implies \text{pen}(\mathcal{H}^+, \mathcal{H}^-) \leq (\beta - \alpha) \implies \text{sd}(\mathcal{H}^+, \mathcal{H}^-) \geq (\alpha - \beta)$. Assume $\text{sd}(\mathcal{H}^+, \mathcal{H}^-) > \alpha - \beta \implies \text{pen}(\mathcal{H}^+, \mathcal{H}^-) < \beta - \alpha \implies \exists\, t \in \mathbb{R}^n, \|t\| < \beta - \alpha$ such that $(\mathcal{H}^+ + t) \cap (\mathcal{H}^- \setminus \partial\mathcal{H}^-) = \emptyset$. Let $x \in \mathcal{H}^+$ satisfy $c^T x = \alpha \implies c^T(x+t) = \alpha + c^T t \leq \alpha + \|c\|\|t\| < \beta \implies x + t \in (\mathcal{H}^- \setminus \partial\mathcal{H}^-)$ a contradiction. Therefore $\text{sd}(\mathcal{H}^+, \mathcal{H}^-) = \alpha - \beta$.

□

A.6 Proof of Lemma 5.2

Let $\text{sd}(\mathcal{V}^+, \mathcal{O}^+) = -a \leq 0 \implies \text{pen}(\mathcal{V}^+, \mathcal{O}^+) = a \implies \exists\, t \in \mathbb{R}^n, \|t\| = a$ such that $(\mathcal{V}^+ + t) \cap (\mathcal{O}^+ \setminus \partial\mathcal{O}^+) = \emptyset \implies (\mathcal{V} + t) \cap (\mathcal{O} \setminus \partial\mathcal{O}) = \emptyset \implies \text{pen}(\mathcal{V}, \mathcal{O}) \leq a \implies \text{sd}(\mathcal{V}, \mathcal{O}) \geq -a$. The $\text{sd}(\mathcal{V}^+, \mathcal{O}^+) > 0$ case can be shown using similar arguments.

A.7 Proof of Lemma 5.3

Proof. Note that any $\mathcal{V}(x(t)) \in \text{sv}_{\mathcal{V},f}(x_i, \bar{u}, t_i, t_f)$ satisfies

$$\begin{aligned} \mathcal{V}(x(t)) &= R(x_i)\mathcal{A} + (1 - \xi(t))p(x_i) + \xi(t)p(x_f) \\ &= (1 - \xi(t))\mathcal{V}(x_i) + \xi(t)\mathcal{V}(x_f). \end{aligned} \tag{A.15}$$

Given $\xi(t)$ is continuous with $\xi(t_i) = 0, \xi(t_f) = 1 \implies \forall\, \lambda \in [0,1]\, \exists\, t \in [t_i, t_f] \,|\, \xi(t) = \lambda$. It follows that

$$\text{sv}_{\mathcal{V},f}(x_i, \bar{u}, t_i, t_f) = \text{co}(\{\mathcal{V}(x_i), \mathcal{V}(x_f)\}). \tag{A.16}$$

□

A.8 Proof of Lemma 5.4

Proof. Define the following halfspaces:

$$\mathcal{H}^+ = \{x \,|\, c^T x \geq \mu_\mathcal{C}(c)\} \tag{A.17}$$

$$\mathcal{H}^- = \{x \,|\, c^T x \leq \sigma_\mathcal{D}(c)\} \tag{A.18}$$

From Lemma 5.1 $\mathrm{sd}(\mathcal{H}^+, \mathcal{H}^-) = \mu_\mathcal{C}(c) - \sigma_\mathcal{D}(c)$. Noting that $\mathcal{C} \subset \mathcal{H}^+$ and $\mathcal{D} \subset \mathcal{H}^-$ yields the stated inequality.

□

A.9 Proof of Lemma 5.5

We prove this for the case in which $\mathrm{sd}(\mathcal{C}, \mathcal{D}) < 0$. The $\mathrm{sd}(\mathcal{C}, \mathcal{D}) \geq 0$ case can be shown using similar arguments.

$\mathrm{sd}(\mathcal{C}, \mathcal{D}) = \gamma < 0 \implies \mathrm{pen}(\mathcal{C}, \mathcal{D}) = |\gamma| \implies \exists t \in \mathbb{R}^n, \|t\| = |\gamma|$ such that $(\mathcal{C} + t) \cap (\mathcal{D} \setminus \partial\mathcal{D}) = \emptyset$. As these are disjoint convex sets there exists a separating hyperplane $\mu_\mathcal{C}(c) \geq \sigma_{\mathcal{D}\setminus\partial\mathcal{D}}(c)$ for some $c \in \mathbb{R}^n \setminus 0$. By the scaling properties of the cost and support function we can take $\|c\| = 1$ w.l.o.g. Noting that $\sigma_{\mathcal{D}\setminus\partial\mathcal{D}}(c) = \sigma_\mathcal{D}(c)$ we obtain $\mu_\mathcal{C}(c) + c^T t \geq \sigma_\mathcal{D}(c)$ for some $\|c\| = 1$. Assume $c \neq \frac{t}{\|t\|} \implies c^T t < \|t\|$. Let $\kappa = (c^T t) \implies \mu_\mathcal{C}(c) + c^T(\kappa c) \geq \sigma_\mathcal{D}(c) \implies \mu_{\mathcal{C}+\kappa c}(c) \geq \sigma_\mathcal{D}(c) \implies (\mathcal{C} + \kappa c) \cap (\mathcal{D} \setminus \partial\mathcal{D}) = \emptyset \implies \mathrm{pen}(\mathcal{C}, \mathcal{D}) \leq \kappa < \|t\| = |\gamma|$ a contradiction. Thus $c = \frac{t}{\|t\|}$. Assume $\mu_\mathcal{C}(c) + c^T t > \sigma_\mathcal{D}(c) \implies \mu_\mathcal{C}(c) + |\gamma| > \sigma_\mathcal{D}(c) \implies \mathrm{sd}(\mathcal{C}, \mathcal{D}) > -|\gamma|$ by Lemma 5.4 a contradiction. Thus $\mu_\mathcal{C}(c) + c^T t = \sigma_\mathcal{D}(c)$ for some $\|c\| = \frac{t}{\|t\|}$ where $(\mathcal{C} + t) \cap (\mathcal{D} \setminus \partial\mathcal{D}) = \emptyset$.

A.10 Proof of Lemma 5.9

Proof. From Lemma 5.5 there exists $c \in \mathbb{R}^n, \|c\| = 1$ such that $\mu_\mathcal{C}(c) - \sigma_\mathcal{D}(c) = \mathrm{sd}(\mathcal{C}, \mathcal{D}) \implies \mu_\mathcal{C}(c) - r_\mathcal{C}\sqrt{c^T c} - \sigma_\mathcal{D}(c) - r_\mathcal{D}\sqrt{c^T c} = \mathrm{sd}(\mathcal{C}, \mathcal{D}) - r_\mathcal{C} - r_\mathcal{D} \implies \mu_{\mathcal{C}\oplus B_{r_\mathcal{C}}}(c) -$

$\sigma_{\mathcal{D}\oplus B_{r_\mathcal{D}}}(c) = \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D}$. From Lemma 5.4, this implies $\mathrm{sd}(\mathcal{C}\oplus B_{r_\mathcal{C}}, \mathcal{D}\oplus B_{r_\mathcal{D}}) \geq \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D}$. Assume $\mathrm{sd}(\mathcal{C}\oplus B_{r_\mathcal{C}}, \mathcal{D}\oplus B_{r_\mathcal{D}}) > \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D} \implies \exists\, c \in \mathbb{R}^n, \|c\| = 1$ such that $\mu_{\mathcal{C}\oplus B_{r_\mathcal{C}}}(c) - \sigma_{\mathcal{D}\oplus B_{r_\mathcal{D}}}(c) > \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D} \implies \mu_\mathcal{C}(c) - r_\mathcal{C}\sqrt{c^Tc} - \sigma_\mathcal{D}(c) - r_\mathcal{D}\sqrt{c^Tc} > \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D} \implies \mu_\mathcal{C}(c) - \sigma_\mathcal{D}(c) > \mathrm{sd}(\mathcal{C},\mathcal{D}) \implies \mathrm{sd}(\mathcal{C},\mathcal{D}) > \mathrm{sd}(\mathcal{C},\mathcal{D})$ a contradiction. Thus $\mathrm{sd}(\mathcal{C}\oplus B_{r_\mathcal{C}}, \mathcal{D}\oplus B_{r_\mathcal{D}}) = \mathrm{sd}(\mathcal{C},\mathcal{D}) - r_\mathcal{C} - r_\mathcal{D}$. □

References

1. Polyak, B., Scherbakov, P. & Shmulyian, S. Construction of value set for robustness analysis via circular arithmetic. *International Journal of Robust and Nonlinear Control* **4,** 371–385 (May 1994).

2. Farouki, R. & Pottmann, H. Exact Minkowski Products of N Complex Disks. *Reliable Computing* **8,** 43–66 (Feb. 2002).

3. Huang, X., Ryu, E. & Yin, W. Tight Coefficients of Averaged Operators via Scaled Relative Graph. *Journal of Mathematical Analysis and Applications* **490,** 124211 (May 2020).

4. Guthrie, J. *Large-Signal Stability Analysis of Pulsed Constant Power Loads via Sum-of-Squares Optimization* in *2019 IEEE Electric Ship Technologies Symposium (ESTS)* (2019), 127–133.

5. Guthrie, J. & Mallada, E. *Adversarial Model Predictive Control via Second-Order Cone Programming* in *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), 1403–1409.

6. Guthrie, J. & Mallada, E. *Minimum-Time Charging of Energy Storage in Microgrids via Approximate Conic Relaxation* in *2020 European Control Conference (ECC)* (2020), 1713–1718.

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

8. Magnani, A., Lall, S. & Boyd, S. *Tractable fitting with convex polynomials via sum-of-squares* in *Proceedings of the 44th IEEE Conference on Decision and Control* (2005), 1672–1677.

9. Henrion, D. & Lasserre, J.-B. Inner Approximations for Polynomial Matrix Inequalities and Robust Stability Regions. *IEEE Transactions on Automatic Control* **57,** 1456–1467 (2012).

10. Ahmadi, A. A., Hall, G., Makadia, A. & Sindhwani, V. *Geometry of 3D Environments and Sum of Squares Polynomials* in *Robotics: Science and Systems* (2017).

11. Cerone, V., Piga, D. & Regruto, D. *Polytopic outer approximations of semialgebraic sets* in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (2012), 7793–7798.

12. Guthrie, J. & Mallada, E. *Outer Approximations of Minkowski Operations on Complex Sets via Sum-of-Squares Optimization* in *2021 American Control Conference (ACC)* (2021), 2367–2373.

13. Guthrie, J., Kobilarov, M. & Mallada, E. *Closed-Form Minkowski Sum Approximations for Efficient Optimization-Based Collision Avoidance* in *2022 American Control Conference (ACC)* (2022).

14. Jones, M. & Peet, M. M. *Using SOS for Optimal Semialgebraic Representation of Sets: Finding Minimal Representations of Limit Cycles, Chaotic Attractors and Unions* in *2019 American Control Conference (ACC)* (2019), 2084–2091.

15. Lasserre, J.-B. A generalization of Löwner-John's ellipsoid theorem. *Mathematical Programming* **152** (Aug. 2014).

16. Parrilo, P. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization* PhD thesis (California Institute of Technology, 2000).

17. Brunn, H. Über Kerneigebiete. *Math. Ann.* **93,** 436–440 (1913).

18. Henrion, D., Lasserre, J.-B. & Savorgnan, C. Approximate Volume and Integration for Basic Semialgebraic Sets. *SIAM Review* **51,** 722–743 (Nov. 2009).

19. Korda, M. & Henrion, D. Convergence rates of moment-sum-of-squares hierarchies for volume approximation of semialgebraic sets. *Optimization Letters* **12,** 435–442 (2018).

20. Putinar, M. Positive Polynomials on Compact Semi-algebraic Sets. *Indiana University Mathematics Journal* **42,** 969–984 (1993).

21. ApS, M. *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.* (2017).

22. Lofberg, J. *YALMIP : a toolbox for modeling and optimization in MATLAB* in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)* (Sept. 2004), 284–289.

23. Delanoue, N., Jaulin, L. & Cottenceau, B. Using interval arithmetic to prove that a set is path-connected. *Theoretical Computer Science* **351,** 119–128 (2006).

24. Polyak, B. & Vishnyakov, A. Multiplying Disks: Robust Stability of a Cascade Connection. *European Journal of Control* **2,** 101–111 (Dec. 1996).

25. Skogestad, S. & Postlethwaite, I. *Multivariable feedback control: Analysis and Design* (John Wiley, Hoboken, US-NJ, 2005).

26. Farouki, R., Moon, H. & Ravani, B. Minkowski Geometric Algebra of Complex Sets. *Geometriae Dedicata* **85** (July 2000).

27. Lysenko, M., Shapiro, V. & Nelaturi, S. Non-commutative morphology: Shapes, filters, and convolutions. *Computer Aided Geometric Design* **28,** 497–522 (Nov. 2011).

28. Gargantini, I. & Henrici, P. Circular Arithmetic and the Determination of Polynomial Zeros. *Numer. Math.* **18,** 305–320 (Jan. 1971).

29. Permenter, F. & Parrilo, P. A. *Selecting a monomial basis for sums of squares programming over a quotient ring* in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (2012), 1871–1876.

30. Bock, H. & Plitt, K. A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*. *IFAC Proceedings Volumes* **17.** 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984, 1603–1608 (1984).

31. Biegler, L. T. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering* **8,** 243–247 (1984).

32. Liniger, A., Domahidi, A. & Morari, M. Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods* **36,** 628–647. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2123 (2015).

33. LaValle, S. M. *Planning Algorithms* (Cambridge University Press, 2006).

34. Boyd, S. & Vandenberghe, L. *Convex Optimization* (Cambridge University Press, Mar. 2004).

35. Zhang, X., Liniger, A. & Borrelli, F. Optimization-Based Collision Avoidance. *IEEE Transactions on Control Systems Technology* **29,** 972–983 (2021).

36. Zhang, X., Liniger, A., Sakai, A. & Borrelli, F. *Autonomous Parking Using Optimization-Based Collision Avoidance* in *2018 IEEE Conference on Decision and Control (CDC)* (2018), 4327–4332.

37. Gerdts, M., Henrion, R., Hömberg, D. & Landry, C. Path planning and collision avoidance for robots. *Numerical Algebra, Control and Optimization* **2,** 437–463 (2012).

38. Patel, R. B. & Goulart, P. J. Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization. *Journal of Guidance, Control, and Dynamics* **34,** 218–230. eprint: https://doi.org/10.2514/1.49518 (2011).

39. Ruan, S. & Chirikjian, G. Closed-form Minkowski sums of convex bodies with smooth positively curved boundaries. *Computer-Aided Design* **143** (Feb. 2022).

40. Ruan, S. *et al. Efficient Exact Collision Detection between Ellipsoids and Superquadrics via Closed-form Minkowski Sums* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 1765–1771.

41. Hoffmann, C. M. *Conversion Methods Between Parametric and Implicit Curves and Surfaces* in (1990).

42. Guthrie, J. Inner and Outer Approximations of Star-Convex Semialgebraic Sets - Supplementary Material (2022).

43. Schneider, R. *Convex Bodies: The Brunn–Minkowski Theory* 2nd ed. (Cambridge University Press, 2013).

44. Berg, de, M., Cheong, O., Kreveld, van, M. & Overmars, M. *Computational geometry : algorithms and applications* 3rd. English (Springer, Germany, 2008).

45. Hart, P. E., Nilsson, N. J. & Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* **4,** 100–107 (1968).

46. Mellinger, D., Michael, N. & Kumar, V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research* **31,** 664–674. eprint: https://doi.org/10.1177/0278364911434236 (2012).

47. Wächter, A. & Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* **106,** 25–57 (2006).

48. Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B. & Diehl, M. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation* **11,** 1–36 (2019).

49. Choi, Y.-K., Chang, J.-W., Wang, W., Kim, M.-S. & Elber, G. Continuous Collision Detection for Ellipsoids. *IEEE Transactions on Visualization and Computer Graphics* **15,** 311–325 (2009).

50. Dueri, D., Mao, Y., Mian, Z., Ding, J. & Açikmeşe, B. *Trajectory optimization with inter-sample obstacle avoidance via successive convexification* in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017), 1150–1156.

51. Schulman, J. *et al.* Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* **33,** 1251–1270 (2014).

52. Guthrie, J. Inner and Outer Approximations of Star-Convex Semialgebraic Sets. *IEEE Control Systems Letters* **7,** 61–66 (2023).

53. Manchester, Z. & Kuindersma, S. Robust direct trajectory optimization using approximate invariant funnels. *Autonomous Robots* **43,** 375–387 (2019).

54. Kim, Y., Leyffer, S. & Munson, T. in, 335–360 (Nov. 2020).

55. Howell, T. *et al.* Trajectory Optimization with Optimization-Based Dynamics. *IEEE Robotics and Automation Letters* **7,** 6750–6757 (July 2022).

56. Hogan, F. *Reactive Manipulation with Contact Models and Tactile Feedback* PhD thesis (Massachusetts Institute of Technology, 2020).