

Differentiable Collision-Free Parametric Corridors

Jon Arrizabalaga^{1,2}, Zachary Manchester², Markus Ryll¹

Abstract— This paper presents a method to compute differentiable collision-free parametric corridors. In contrast to existing solutions that decompose the obstacle-free space into multiple convex sets, the continuous corridors computed by our method are smooth and differentiable, making them compatible with existing numerical techniques for learning and optimization. To achieve this, we represent the collision-free corridors as a path-parametric off-centered ellipse with a polynomial basis. We show that the problem of maximizing the volume of such corridors is convex, and can be efficiently solved. To assess the effectiveness of the proposed method, we examine its performance in a synthetic case study and subsequently evaluate its applicability in a real-world scenario from the KITTI dataset.

Code: <https://github.com/jonarriza96/corrgen>

Video: <https://youtu.be/MvC7bPodXz8>

I. INTRODUCTION

Path-parametric methods have gained popularity in the formulation of navigation algorithms, such as high-level planners [1]–[3], reinforcement learning (RL) policies [4]–[6] or low-level model predictive controllers (MPC) [7]–[9]. The fundamental concept behind these parametric methods is to either introduce the path-parameter as an additional degree of freedom, enabling the system to regulate its progress along the path [10], [11], or to conduct a change of coordinates that project the *Euclidean states* to the *spatial states*, i.e., the progress along the path and the orthogonal distance to it [12]–[14]. These parametric formulations have proven successful for two primary reasons: firstly, they inherently capture the notion of advancement along the path, and secondly, spatial bounds manifest as convex constraints in the orthogonal terms of the spatial states.

Despite the increasing interest in path-parametric methods, existing collision-free space representation techniques proposed in the literature are not compatible with such formulations. The current solutions to this problem, often referred to as *corridor generation*, involve decomposing the free space into a collection of convex polyhedra [15]–[18]. However, applying these methods to the aforementioned parametric formulations presents two significant drawbacks: Firstly, discretizing the free space into multiple convex sets introduces the *polyhedron allocation problem*, wherein the assignment of a polyhedron to a particular state must be pre-determined [19]. Essentially, this discretization disrupts the differentiability of the corridors with respect to the progress variable, making it challenging to integrate the collision-free corridors into optimization and learning routines. Secondly,

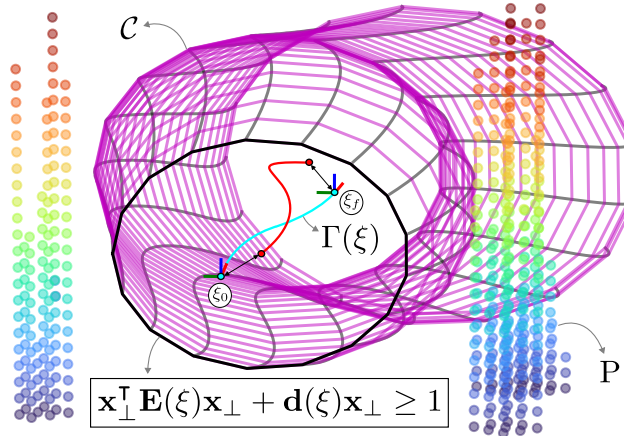


Fig. 1: Given a reference path Γ parameterized by path-parameter ξ and a point cloud P , we present a method that computes differentiable and smooth parametric corridors \mathcal{C} . The reference path Γ is depicted by the blue line with a moving frame, the colored dots refer to the point cloud P and the corridor \mathcal{C} is given in magenta. To generate the corridor, we optimize over a parametric ellipse $\{\mathbf{E}(\xi), \mathbf{d}(\xi) \mid \xi \in [\xi_0, \xi_f]\}$, where besides searching for its dimensions, we also compute the offset from the reference Γ . In other words, the center of the ellipse, shown in red, is allowed to deviate from the reference Γ . Our method can run real-time and is applicable to both, planar (2D) and spatial (3D) cases.

while representing the free space with a set of hyperplanes aligns with the *Euclidean* perspective, it neglects the ability to enforce convex constraints in the orthogonal components.

In this context, where conventional collision-free generation algorithms cannot be directly applied to parametric formulations, and where obstacle avoidance primarily involves constraining the transverse states, the development of parametric collision-free corridors has been overlooked. Consequently, the corridors generated by studies utilizing parametric formulations are often tailored to specific cases and do not generalize across different scenarios [9], [20], [21]. Due to the lack of a cross-compatible parametric corridor generation approach, most of the literature on path-parameterized methods prioritizes performance over safety, dropping the ability to impose spatial constraints, and consequently, leading to the loss of formal safety guarantees. In the MPC framework, this translates into an extremely fragile tuning process [22], heavily reliant on the sophistication and patience of an expert. Similarly, in the RL scenario this entails highly penalizing excursions out of the admissible corridors, forcing the policy to learn a trade off between optimality and safety [23], [24].

In light of these shortcomings, there is a need to properly address the generation of collision-free parametric corridors. Multiple research questions arise naturally, including how to parameterize the obstacle-free space in a differentiable way, and how to construct a lightweight program capable of computing the respective corridors in real-time.

¹Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM)

²Robotics Institute (RI), Carnegie Mellon University (CMU)

In this paper, we address this need by presenting a method that, given a point cloud and a reference path, computes a collision-free parametric corridor around it. More specifically, the generated corridors are differentiable with respect to the path-parameter, making them compatible with existing optimization and learning pipelines; they can be larger than the corridors generated by existing tools; and they rely on a well-structured convex optimization formulation that can be solved in real-time. To the best of the authors' knowledge this is the first generic method that enables the computation of differentiable parametric corridors in real-time that are compatible with path-parametric formulations. To achieve this, our method relies on three main ingredients:

- 1) We represent the obstacle-free space as a parametric off-centered ellipse with Chebyshev polynomials.
- 2) We formulate a convex optimization problem – a Linear Program (LP) for 2D planar cases and a Semidefinite Program (SDP) for 3D spatial cases – capable of maximizing the volume of the corridor within the collision-free space.
- 3) We show that approximating the parametric ellipse by a diagonally-dominant matrix reduces the original SDP problem to an LP, resulting in real-time computational tractability without compromising the corridor's volumetric capacity.

The remainder of this paper is structured as follows: Section II formally introduces the parametric corridor-generation problem tackled in this work, Section III presents our solution by delving deeper into all three ingredients, Section IV shows the simulated and experimental results, and lastly, Section V presents the conclusions.

II. PROBLEM STATEMENT

In this section, we introduce the foundational concepts of our method, namely the reference path, the point cloud and the corridor. While the notation and problem description are presented in a generic spatial 3D form, we note that they also apply to the planar 2D case.

Let the occupied space be represented by a *point cloud* $P \in \mathbb{R}^{m \times 3}$ with m points

$$P = [\mathbf{p}_1, \dots, \mathbf{p}_m] \quad \text{where} \quad \mathbf{p}_{1, \dots, m} \in \mathbb{R}^3, \quad (1)$$

and assume Γ to be a *reference path* within the free space $P \notin \Gamma$. Without loss of generalization, we define Γ as a parametric path with a moving frame attached to it. Its respective position and orientation are given by two functions, $\gamma : \mathbb{R} \mapsto \mathbb{R}^3$ and $R_\Gamma : \mathbb{R} \mapsto \mathbb{R}^{3 \times 3} \subseteq \text{SO}(3)$, dependant on path-parameter ξ :

$$\Gamma = \{\xi \in [\xi_0, \xi_f] \subseteq \mathbb{R} \mapsto \gamma(\xi) \in \mathbb{R}^3, R_\Gamma(\xi) \in \mathbb{R}^{3 \times 3}\}. \quad (2)$$

Additionally, we define the transverse plane $\perp_\Gamma(\xi) \in \mathbb{R}^2$ as the plane orthogonal to the path, i.e. its basis is given by the second and third column of R_Γ .

The collision-free space associated to point cloud P around reference Γ is denoted as a *corridor* \mathcal{C}_Γ^P and encoded as an

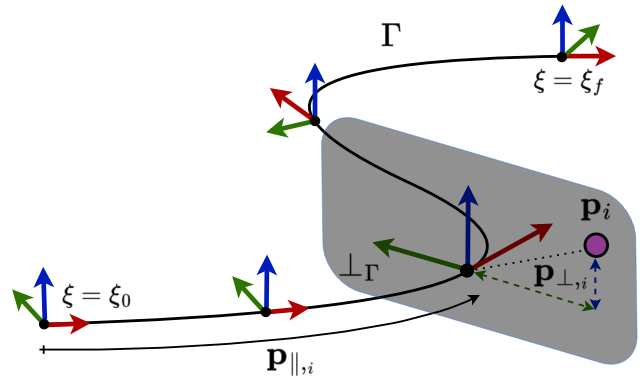


Fig. 2: Projection of a point \mathbf{p}_i , represented by the pink dot, from its Euclidean coordinates onto the reference path Γ which is parameterized by ξ and has a moving frame $R_\Gamma(\xi)$ attached to it. The orthogonal plane \perp_Γ defined by the second (green) and third (blue) components of the frame is depicted in gray. The coordinates resulting from the projection are decoupled into a tangent $\mathbf{p}_{\parallel,i}$ and an orthogonal $\mathbf{p}_{\perp,i}$ component.

inequality on a function $c_\Gamma^P : \mathbb{R} \mapsto \mathbb{R}^c$, such that,

$$\mathcal{C}_\Gamma^P = \{\xi \in [\xi_0, \xi_f] \subseteq \mathbb{R}, \mathbf{x}_\perp \in \perp_\Gamma(\xi) \subseteq \mathbb{R}^2 \mapsto c_\Gamma^P(\xi, \mathbf{x}_\perp) \leq \mathbf{0} \subseteq \mathbb{R}^c\}, \quad (3)$$

where c depends on the selected free space representation, and thus, will be specified in the upcoming Section III. For a visual illustration of the point cloud P in (1), the reference path Γ in (2) and the collision-free corridor \mathcal{C}_Γ^P , please see Fig. 1.

Having defined the preliminaries, we can now formally state the problem addressed in this paper:

Problem 1 (Collision-free corridor generation): Given the point cloud P in (1) and the parametric reference path Γ in (2), compute a parametric corridor \mathcal{C}_Γ^P , as defined in (3), that is:

- P1.1 **Collision-free:** The corridor is contained within the obstacle-free space, i.e., $P \notin \mathcal{C}_\Gamma^P$.
- P1.2 **Differentiable:** The corridor is (at least) two times differentiable and continuous with respect to the path-parameter, i.e., $c_\Gamma^P(\xi)$ is C_ξ^2 .
- P1.3 **Volume-maximizing:** The corridor encompasses the entire available space around the reference path Γ without intersecting with the point cloud P .

Fulfilment of P1.1 and P1.3 results from the desire of maximizing navigation performance while maintaining safety. Besides that, P1.2 guarantees the existence of continuous first and second-order derivatives, and thereby, ensures the corridor's compatibility with existing numerical methods for learning and optimization. In the remainder of this manuscript, we present a method that solves this problem.

III. SOLUTION APPROACH

The corridor-generation problem discussed in this paper is built upon (i) a projection of the point cloud P to the reference path Γ , (ii) a representation of the corridor as a parametric ellipse whose components are polynomials and (iii) a convex optimization problem that allows for maximizing the corridor's volume. These three elements serve as the fundamental building blocks of the proposed methodology. In

the following section, we provide a more in-depth discussion of each of these components.

A. Projecting the point cloud to the reference path

Prior to selecting a specific corridor representation for (3), a preliminary step is necessary: projecting the point cloud from Euclidean space onto the reference path Γ , i.e., $P \rightarrow P_\Gamma = [P_\parallel, P_\perp]$. This process involves decomposing the point cloud into its tangential component P_\parallel , which corresponds to the path-parameter associated with each point in the cloud, and its orthogonal component P_\perp , representing the projection of the distance to the reference Γ onto the transverse plane \perp_Γ . Fig. (2) provides a visual depiction of this point projection onto the reference path.

From a mathematical standpoint, the projection is conducted in two steps. First, we find the path-parameters associated to the closest point in the reference path by

$$P_\parallel = \arg \min_{\xi} \|P - \gamma(\xi)\|_2, \quad (4a)$$

and second, we compute the orthogonal term by projecting the distance vector to the transverse plane of the reference path:

$$P_\perp = (P - \gamma(P_\parallel)) R_\Gamma(P_\parallel). \quad (4b)$$

Notice that the first (tangential) term of P_\perp is 0, while the remaining two decompose the distance into the second and third components of the moving frame R_Γ . Besides that, the projection operation, as outlined in eqs. (4a) and (4b), is fully parallelizable, and thus, it is particularly well-suited for implementation on a GPU, enabling efficient projections regardless of the size of the point cloud.

B. Selecting a parametric cross section

Motivated by the need to formulate a corridor-generation method that is compatible with path-parameterized navigation methods and existing learning and optimization routines, we consider *parametric* corridors of the form (3). In other words, the corridors consist of a predefined cross section, whose dimensions evolve according to path-parameter ξ .

The choice of the cross section holds significant importance as it directly impacts the expressiveness and flexibility of the corridor. A cross section with fewer degrees of freedom limits the corridor's ability to accurately approximate the true free space. Hence, our objective is to select a cross section that offers maximum adaptability while ensuring differentiability¹ and computational feasibility.

The simplest cross section consists of a circle, which offers a single degree of freedom (the radius). An ellipse adds another degree of freedom by enabling different width and height parameters. Letting the ellipse rotate further increases the decision variables to 3. Additionally, allowing it to have an offset from the center raises the total degrees of freedom to 5. Ultimately, by expanding the ellipse's basis with

¹The naive way to approach this problem would be to solve a planar convex decomposition problem at the desired transverse plane. Since the number of sides associated with a polygon obtained from convex decomposition is unknown, the resulting corridor would not comply with P1.2 in Problem 1, i.e., it would be neither differentiable nor continuous.

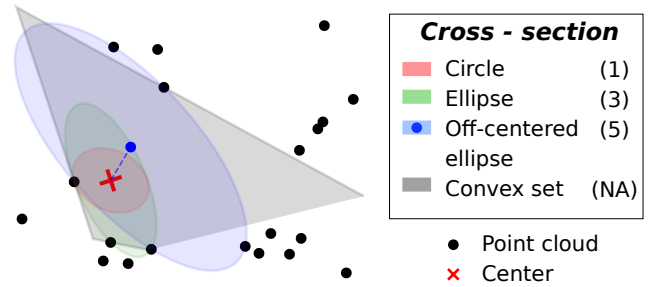


Fig. 3: Area maximization for different cross section representations centered in the red cross within the transverse point cloud P_\perp represented by the black dots. The depicted cross sections are: a circle with 1 degree of freedom (red), an ellipse with 3 (green) and an off-centered ellipse with a total 5 (blue). The gray area represents the maximum convex polygon for the given point cloud encompassing the center. The offset for the off-centered ellipse is given by the blue dashed line. From this comparison, it is apparent that the higher the number the degrees of freedom, the bigger the area of the cross section.

additional polynomial terms, we can create a semialgebraic set [25], and thus, provide a high degree of expressiveness to the cross section. This concept is illustrated in Fig. 3, which compares the cross sections obtained by solving an area maximization problem. As can be seen, a higher number of degrees of freedom corresponds to a larger cross-sectional area.

The richness of semialgebraic sets comes at the expense of difficulties in guaranteeing the closeness and convexity of the cross section [26]. This choice makes the overall corridor generation pipeline more complicated, both from a formulation and a numerical standpoint. For this reason, we pick the next most favorable option: a rotating off-centered ellipse. Notice that, within the navigation scenario, allowing the ellipse to have an offset with respect to the reference path Γ is of major importance, since the reference might be very close to the point cloud, specially if it is obtained from search or sampling-based methods² [31].

The aforementioned off-centered rotating ellipse is given by

$$\mathcal{E}(E, \mathbf{p}_E) = \{\mathbf{x}_\perp \in \mathbb{R}^2 \mid (\mathbf{x}_\perp - \mathbf{p}_E)^\top E (\mathbf{x}_\perp - \mathbf{p}_E) \leq 1\}, \quad (5)$$

where $E \in S_{++}^2$ is a symmetric positive-definite matrix and $\mathbf{p}_E \in \mathbb{R}^2$ is the center of the ellipse. Once the cross section is defined as a rotating off-centered ellipse in (5), the next step is to path-parameterize it to sweep along the reference path from the initial point ξ_0 to the final point ξ_f . To achieve this, we employ polynomials of degree n that govern the evolution of both the ellipse and its center with respect to the path-parameter ξ . This allows us to fully define the parametric function of the corridor introduced in (3) as:

$$\mathcal{c}_\Gamma^P(\xi, \mathbf{x}_\perp, \mathbf{c}_E, \mathbf{c}_P) = (\mathbf{x}_\perp - \mathbf{p}_E(\xi, \mathbf{c}_P))^\top E(\xi, \mathbf{c}_E) (\mathbf{x}_\perp - \mathbf{p}_E(\xi, \mathbf{c}_P)) - 1, \quad (6)$$

where $\mathbf{c}_E \in \mathbb{R}^{2 \times 2 \times n+1}$ and $\mathbf{c}_P \in \mathbb{R}^{2 \times n+1}$ are the coefficients of the polynomials associated with the ellipse matrix E

²This also provides an alternative view on the corridor-generation problem, where corridors whose cross sections are allowed to be off-centered and contain a rich-enough representation are an appealing alternative to other hierarchical and compute-heavy mapping methods, such as Euclidean Signed Distance Functions (ESDFs) [27]–[30].

and its centerpoint \mathbf{p}_E , respectively. From (6), it is apparent that the dimension of the inequality in (3) is $c = 1$.

Intuitively, increasing the degree n of the polynomial enhances the corridor's ability to adapt to variations along the reference path. The utilization of polynomials offers dual advantages: Firstly, it ensures that the resulting corridors exhibit inherent smoothness and differentiability with respect to the path-parameter ξ . Secondly, this formulation aligns with the *sums of squares* framework [32], a critical aspect for formulating the volume maximization problem discussed in the subsequent subsection.

C. Corridor volume maximization

To encapsulate the entire free space, we aim to maximize the volume of the corridor by formulating an optimization problem that finds the polynomial coefficients \mathbf{c}_E and \mathbf{c}_P in (6). Given that the area of the parametric ellipse \mathcal{E} in (5) is $A_{\mathcal{E}} = \pi/\sqrt{\det E}$, the optimization problem we seek to solve is as follows:

$$\max_{\mathbf{c}_E, \mathbf{c}_P} V_{\mathcal{E}} = \int_{\xi_0}^{\xi_f} -\det E(\xi, \mathbf{c}_E) d\xi \quad (7a)$$

$$\text{s.t. } E(\xi, \mathbf{c}_E) \succ 0, \quad \xi \in [\xi_0, \xi_f] \quad (7b)$$

$$\mathbf{c}_{\Gamma}^P(\mathbf{P}_{\parallel}, \mathbf{P}_{\perp}, \mathbf{c}_E, \mathbf{c}_P) \geq 0, \quad (7c)$$

where (7b) ensures that the ellipse matrix $E(\cdot, \mathbf{c}_E)$ remains positive-definite throughout the entire corridor and (7c) guarantees that all the points in the cloud lie outside the corridor.

The current form of the optimization problem (7) is defined in the continuous domain, rendering it infeasible for direct implementation on a computer. Moreover, it is characterized by nonlinearity and nonconvexity. To make it computationally tractable, we reformulate the problem and introduce several modifications aimed at convexifying it. In particular, we address four key modifications:

Firstly, we discretize the continuous parts of the optimization problem, namely the cost function (7a) and the positive-definite constraint (7b), into N evaluations. This discretization is valid when N is significantly larger than the order of the polynomials, i.e., $N \gg n$. Secondly, to avoid the nonlinearities associated with the determinant of the ellipse in (7a), we approximate it with the trace of the matrix, thus yielding $\det E \approx \text{tr}(E)$. Thirdly, we prevent unbounded growth in obstacle-free areas by introducing upper bounds on the dimensions of the parametric ellipse. To this end, we modify the point cloud \mathbf{P} by introducing a wrapper around the path Γ , which not only sets an upper limit but also allows us to exclude points located further away than the wrapper itself, thereby reducing the total number of constraints in the optimization problem³. As a consequence, the optimization problem is always bounded and feasible. The reduced point cloud, obtained after introducing the wrapper and removing the points outside of it, is denoted as ${}_w\mathbf{P} \in \mathbb{R}^{w \times 3}$, where w is the number of remaining points. Fourth, to address

³An alternative method is to decouple constraint (7b) into $E - \lambda \mathbf{I} \succ 0$ and $-E + \bar{\lambda} \mathbf{I} \succ 0$, where λ and $\bar{\lambda}$ represent the maximum and minimum eigenvalues of the ellipse matrix E , respectively. These eigenvalues correspond to the maximum \bar{l} and minimum l axis sizes of the ellipse, related by $\lambda = 2/\sqrt{\bar{l}}$ and $\bar{\lambda} = 2/\sqrt{l}$.

the nonlinearities in (7c), we reformulate the corridor's parametric definition as

$$\partial_{\Gamma}^P(\xi, \mathbf{x}_{\perp}, \mathbf{c}_E, \mathbf{c}_D) = \mathbf{x}_{\perp}^T E(\xi, \mathbf{c}_E) \mathbf{x}_{\perp} + \mathbf{d}(\xi, \mathbf{c}_D) \mathbf{x}_{\perp} - 1, \quad (8)$$

where $\mathbf{d}(\xi, \mathbf{c}_D) \in \mathbb{R}^2$ is a polynomial on ξ with coefficients $\mathbf{c}_D \in \mathbb{R}^{2 \times n+1}$. This term allows for the aforementioned offset from the reference path Γ , while removing the nonlinearities of (6).

By incorporating all four modifications, the approximation to the original problem (7) that we actually solve is given by:

$$\min_{\mathbf{c}_E, \mathbf{c}_D} -V_{\mathcal{E}} \approx \sum_{i=1}^N \text{tr}(E(\xi_i, \mathbf{c}_E)) \quad (9a)$$

$$\text{s.t. } E(\xi_i, \mathbf{c}_E) \succ 0, \quad i = 1, \dots, N \quad (9b)$$

$$\partial_{\Gamma}^P({}_w\mathbf{P}_{\parallel}, {}_w\mathbf{P}_{\perp}, \mathbf{c}_E, \mathbf{c}_D) \geq 0. \quad (9c)$$

In its present state, (9) is a convex optimization problem with $5n$ decision variables and Nw constraints. More specifically, it is a Semidefinite-Program (SDP) [33], a type of problem that can be efficiently solved with off-the-shelf open-source solvers [34]–[38]. It is worth highlighting that, for high polynomial degrees n , we avoid the problem becoming ill-conditioned by choosing a Chebyshev polynomial basis [39].

Since the positive-definite constraint in (9b) is a two-dimensional Linear Matrix Inequality (LMI), further relaxations allow for approximating SDP (9) into a more restricted and lightweight formulation, such as a Second-Order-Cone-Program (SOCP) or a Linear-Program (LP) [40]. Aiming to maximize computational efficiency, we choose the latter option, which replaces the positive-definite constraint on E with diagonal dominance, i.e., both diagonal terms E_{11} and E_{22} need to be bigger than the off-diagonal one E_{12} . Given that all diagonally dominant matrices are positive-definite [40], we replace constraint (9b) by $E_{11} > E_{12}$ and $E_{22} > E_{12}$, and thus, reduce the original SDP problem (9) into an LP.

D. The planar 2D case

The computation of planar corridors is a subset of the problem introduced in the previous subsection. In this case, the corridor is defined by the area enclosed between two polynomials $\mathbf{b}_{+}(\xi, \mathbf{c}_{B_{+}}) \in \mathbb{R}$ and $\mathbf{b}_{-}(\xi, \mathbf{c}_{B_{-}}) \in \mathbb{R}$, situated on the positive and negative sides of the orthogonal plane ${}_{\perp}\Gamma$. Here, $\mathbf{c}_{B_{+}}$ and $\mathbf{c}_{B_{-}}$ represent the corresponding coefficients. The optimization problem seeks to maximize the area, which involves tailoring problem (9) as follows:

$$\min_{\mathbf{c}_{B_{+}}, \mathbf{c}_{B_{-}}} -A_b \approx -\sum_{i=1}^N \mathbf{b}_{+}(\xi_i, \mathbf{c}_{B_{+}}) + \mathbf{b}_{-}(\xi_i, \mathbf{c}_{B_{-}}) \quad (10a)$$

$$\text{s.t. } \mathbf{b}_{+}(\xi_i, \mathbf{c}_{B_{+}}) > 0, \quad i = 1, \dots, N \quad (10b)$$

$$\mathbf{b}_{-}(\xi_i, \mathbf{c}_{B_{-}}) < 0, \quad i = 1, \dots, N \quad (10c)$$

$$\mathbf{b}_{+}({}_w\mathbf{P}_{\parallel}^{+}, \mathbf{c}_{B_{+}}) \leq {}_w\mathbf{P}_{\perp}^{+}, \quad (10d)$$

$$\mathbf{b}_{-}({}_w\mathbf{P}_{\parallel}^{-}, \mathbf{c}_{B_{-}}) \leq {}_w\mathbf{P}_{\perp}^{-}, \quad (10e)$$

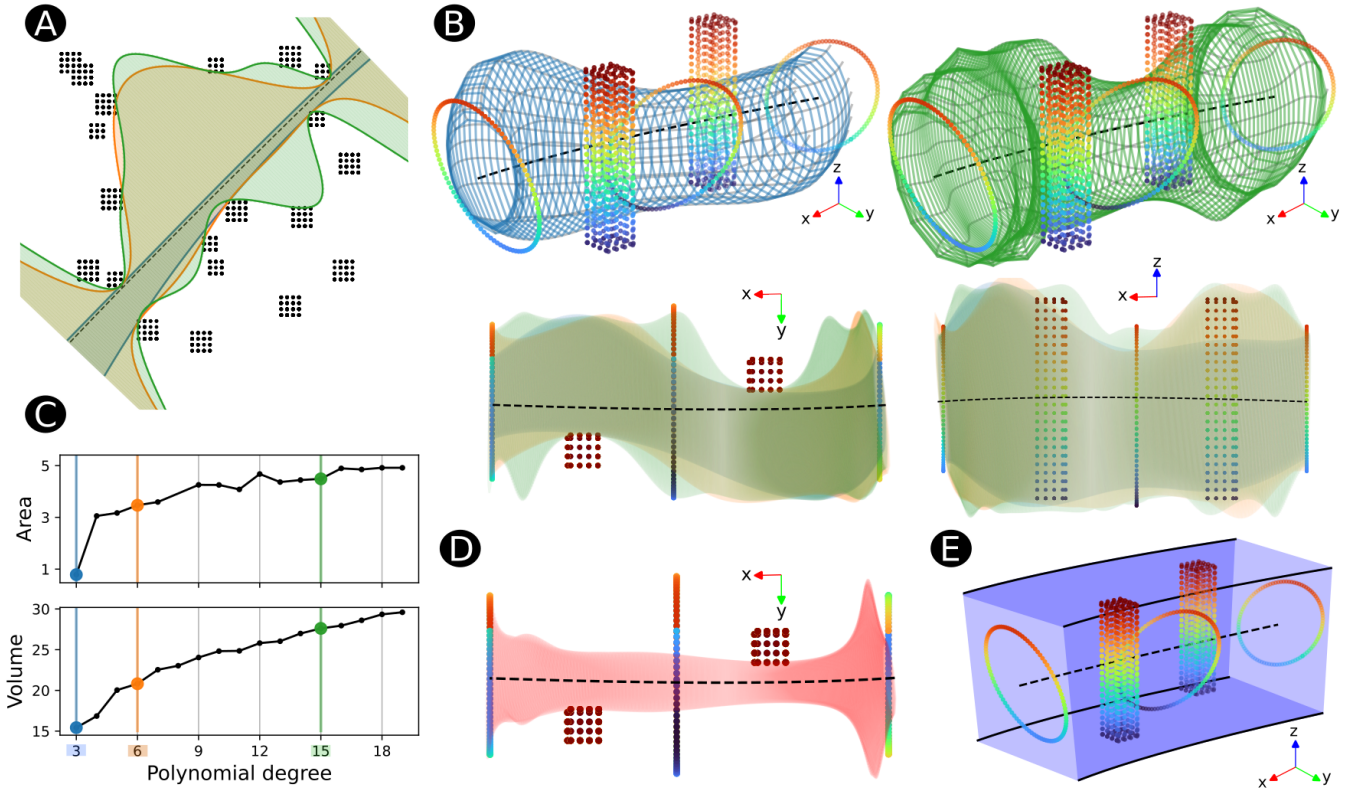


Fig. 4: 2D and 3D corridors generated by our method for different polynomial degrees: 3 in blue, 6 in orange, and 15 in green. In the planar case (A) the point cloud is depicted in black, while in the spatial case (B) it is colored according to the height. The black dashed line represents the reference path Γ . The correlation between the corridor’s area – applicable to (A) – and volume – applicable to (B) – with respect to the polynomial degree is plotted in (C). As the degree increases, the respective parametric areas and volumes increase. This trend is confirmed in (A) and (B), where the expressiveness of the sampled corridors augments with the degree. Panel (D) shows the corridor associated with a parametric ellipse centered on reference Γ , making apparent the benefits of optimizing over the offset. Panel (E) depicts the wrapper that prevents the 3D corridors in (B) and (D) from becoming unbounded.

where wP^+ and wP^- separate the point cloud in the positive and negative sides of the orthogonal plane \perp_{Γ} . The resulting problem (10) is an LP with $2n$ decision variables and $2Nw$ constraints.

IV. EXPERIMENTS

To evaluate our approach, we split the experimental analysis into two parts: First, we get a better understanding of the method’s performance by focusing on two illustrative synthetic case studies. Second, we analyze its behavior in a real-world scenario obtained from the KITTI Vision Benchmark Suite [41].

Numerical implementation

In all evaluations, we compute the reference Γ by first determining a collision-free path using kinodynamic path-planning [42] and then parameterizing it with an interpolation algorithm based on Pythagorean hodograph curves [43]. It is important to note that these choices are entirely modular and independent of the presented corridor-generation method. When benchmarking against convex decomposition, we compare against [16], the implementation of which is available online⁴. Other than that, we discretize the problem into $N = 100$ evaluations and use Clarabel [36] and ProxQP [44] to solve the underlying SDPs and LPs, respectively.

⁴<https://github.com/sikang/DecompUtil>

A. An illustrative toy-example

To better comprehend our corridor-generation method, we examine its performance on two different 2D and 3D exemplary case studies created by randomly positioning a series of columns and rings. In particular, we compare the corridors obtained for various polynomial degrees n and we showcase the importance of optimizing over the offset with respect to reference path Γ .

The resulting solutions are illustrated in Fig. 4. In panels (A) and (B), the corridors obtained for degrees 3, 6, and 15 are presented in blue, orange, and green, respectively, for both 2D and 3D cases. From these corridors, it becomes evident that increasing the degree n enhances the corridor’s expressiveness: the blue corridor is the smallest, while its capacity to adapt to the environment increases with the orange, resulting in the largest corridor for green. In panel (C), we validate this observation by demonstrating the correlation between area/volume and polynomial degree for multiple degrees. Panel (D) shows the top-down view of the corridor obtained from a centered (no offset) parametric ellipse, while panel (E) depicts the wrapper used to bound the computation of the three-dimensional corridors in (B) and (D).

The analysis of our results yields two significant conclusions: Firstly, we observe that increasing the polynomial degree leads to an expansion in the area/volume of the corridor. This augmentation is particularly notable for lower

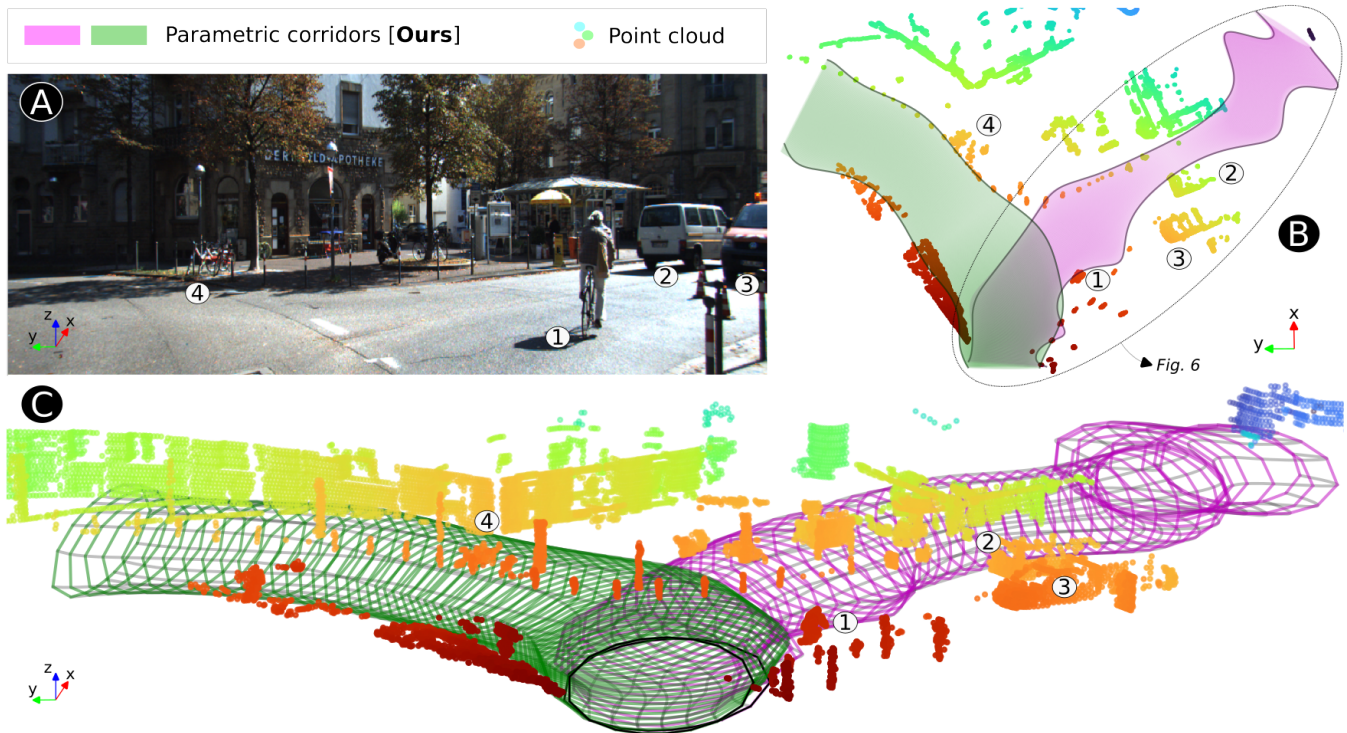


Fig. 5: Two different 3D corridors for polynomial degree $n = 9$ generated by our method in a real-world challenging driving scenario, where the road bifurcates into two narrow lanes, with other cars and cyclists. The case study is part of the KITTI Vision Benchmark Suite [41]. Panel (A) shows an RGB camera-based overview of the scene, while the respective point cloud and corridors are given in panels (B) –top view– and (C) –isometric view–. The point cloud is color-coded based on proximity to the camera’s location. To facilitate the comprehension of the environment, numbered markers in white (from 1 to 4), associated with relevant features in the scene, have been added to all panels. For a detailed analysis of the corridor by the dashed line in Panel (B), please refer to Fig. 6.

degrees; however, it becomes less pronounced for higher degrees. Secondly, the ability to optimize over the ellipse’s center point emerges as a crucial factor. As illustrated in panel (D) of Fig. 4, constraining the ellipse to be centered on the reference path Γ limits the achievable maximum volume, which is specially detrimental when the reference path is close to obstacles. However, allowing for deviations in the center point makes the corridor generation method agnostic to the underlying reference path.

B. Real-world case study

Having gained a deeper understanding on the corridor generation method, we proceed to assess its applicability to a real-world scenario taken from the KITTI dataset [41]. In particular, we focus on a roundabout intersection where a bifurcation into two lanes allows for the computation of two corridors. This intersection presents numerous challenges due to its various features, including cyclists, cars, and narrow roads.

The resulting corridors, along with the lidar-based point cloud and a raw RGB color camera view, are shown in Fig. 5. The numbers added to the image in panel (A) and the point clouds in panels (B) and (C) help to understand the landscape and locate the aforementioned features. From these visualizations it is apparent that the computed corridors encapsulate the free space, extending across the entire width of the road, and adeptly accommodating other vehicles, bicycles, and road boundaries.

To gain insight into how our method compares to the well-established convex decomposition [16], we refer to Fig. 6-A, where the top views of the corridors obtained from sequentially increasing the polynomial degree n are depicted alongside the corridors resulting from decomposing the free space into p convex sets. The results show that both methods encompass very similar spaces in all four cases. However, the hyperparameters differ significantly. In convex decomposition, the quantity and distribution of polyhedra are determined based on the number of segments within a precomputed linear path, which may compromise the corridor’s volume if the reference path has few segments. In contrast, our method employs the polynomial degree n as its unique hyperparameter, while remaining agnostic to the underlying reference. Additionally, Fig. 6-A also depicts how a parametric cross section with a polynomial basis result in a continuous and smooth space representation, contrasting with the discreteness intrinsic to convex decomposition.

Besides the comparison with convex decomposition, Fig. 6-A further confirms the tendency observed in the previous synthetic case studies, namely that increasing the polynomial degree results in larger corridors. In particular, the corridor associated with $n = 24$ shows to encompass the entirety of the available space.

To reflect on the trade off between the volume gains obtained from increasing the polynomial degree n and the corresponding computational burden, we refer to Fig. 6-B. Here, we show the volumes and solve-times obtained from

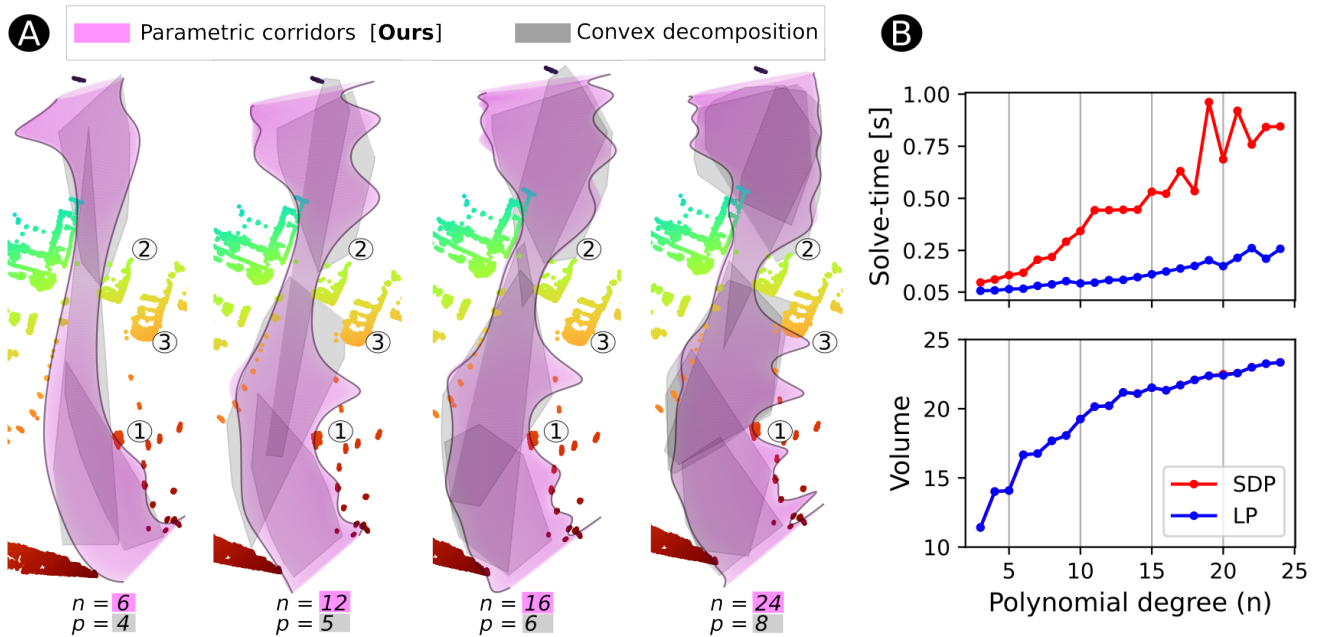


Fig. 6: An in-depth study of the real-world scenario depicted in Fig. 5. Panel (A) illustrates the top view of corridors obtained by progressively increasing the polynomial degree n , alongside the corridors resulting from decomposing the free space into p convex sets, depicted in gray. Panel (B) shows the evolution of corridor’s volume and computation times with respect to the polynomial degree for both SDP and LP problems.

computing the pink corridor in Fig. 5 for various polynomial degrees, ranging from $n = 3$ up to $n = 25$, for both the original SDP and the approximated LP problems. From these results our conclusions are twofold: First, similar to the observations in the previous case studies, the increase in volume is significant for lower degrees and it becomes marginal for higher ones. Second, the approximated LP yields exactly the same corridor volumes as the original SDP, while showing major computational speedups.

We attribute this to two reasons: On the one hand, the cost function (9a) maximizes the trace, practically implying that the matrix of the resulting corridors are diagonally dominant. On the other hand, the freedom to optimize over the offset with respect to the reference Γ overcomes the limitations inherited from requiring matrix E to be diagonally dominant, resulting in corridors capable of encapsulating the available space, while maintaining diagonally dominant matrices.

These results suggest that our method is capable of computing differentiable, continuous and smooth parametric corridors at approximately 5 to 20 Hz, rendering the suggested method real-time deployable.

V. CONCLUSIONS

In this work, we proposed a corridor-generation method that allows the collision-free space to be represented in a continuous and differentiable manner. For this purpose, we predefine the cross section of the corridor as an off-centered ellipse whose components are parameterized by Chebyshev polynomials. Seeking a complete representation of the obstacle-free space, we show that the corridor volume maximization problem can be formulated as a convex optimization problem over the polynomial’s coefficients. More specifically, the problem is a Linear Program (LP) in the 2D planar case and a Semidefinite Program (SDP) in the 3D spatial case. For the latter, we show that approximating

the parametric ellipse by a diagonally dominant matrix also reduces it to an LP, resulting in significant computational speed improvements without compromising the corridor’s volumetric capacity.

The performance of our method has been evaluated in synthetic case studies and further validated in a real-world scenario from the KITTI dataset. These results demonstrate that the corridors generated by our method achieve comparable volumetric capacity to those generated by the discrete convex decomposition method, eliminating the requirement to predefine the number of convex sets. Finally, the continuous, smooth, and differentiable nature of the obtained corridors, coupled with their real-time deployability, underscores the method’s suitability for optimization and learning-based path-parametric navigation algorithms.

REFERENCES

- [1] R. Verschuere, N. van Duijkeren, J. Swevers, and M. Diehl, “Time-optimal motion planning for n-dof robot manipulators using a path-parametric system reformulation,” in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 2092–2097.
- [2] S. Spedicato and G. Notarstefano, “Minimum-time trajectory generation for quadrotors in constrained environments,” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1335–1344, 2017.
- [3] J. Arrizabalaga and M. Ryll, “Sctomp: Spatially constrained time-optimal motion planning,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 4827–4834.
- [4] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, “Autonomous drone racing with deep reinforcement learning,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1205–1212.
- [5] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devic, F. Eckert, F. Fuchs *et al.*, “Outracing champion gran turismo drivers with deep reinforcement learning,” *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.
- [6] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

- [7] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.
- [8] T. Oelerich, F. Beck, C. Hartl-Nesic, and A. Kugi, "Boundmpc: Cartesian trajectory planning with error bounds based on model predictive control in the joint space," *arXiv preprint arXiv:2401.05057*, 2024.
- [9] J. Arrizabalaga and M. Ryll, "Towards time-optimal tunnel-following for quadrotors," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4044–4050.
- [10] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 6137–6142.
- [11] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1026–1039, 2015.
- [12] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [13] N. van Duijkeren, R. Verschuere, G. Pipeleers, M. Diehl, and J. Swevers, "Path-following mmpc for serial-link robot manipulators using a path-parametric system reformulation," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 477–482.
- [14] J. Arrizabalaga and M. Ryll, "Spatial motion planning with pythagorean hodograph curves," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 2047–2053.
- [15] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2015, pp. 109–124.
- [16] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [17] X. Zhong, Y. Wu, D. Wang, Q. Wang, C. Xu, and F. Gao, "Generating large convex polytopes directly on point clouds," *arXiv preprint arXiv:2010.08744*, 2020.
- [18] C. Toumeh and A. Lambert, "Voxel-grid based convex decomposition of 3d space for safe corridor generation," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 4, p. 87, 2022.
- [19] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1934–1940.
- [20] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison, and M. Diehl, "Nmpc for racing using a singularity-free path-parametric model with obstacle avoidance," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14324–14329, 2020.
- [21] A. Romero, S. Govil, G. Yilmaz, Y. Song, and D. Scaramuzza, "Weighted maximum likelihood for controller tuning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1334–1341.
- [22] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.
- [23] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, p. eadg1462, 2023.
- [24] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, 2022.
- [25] M. Shiota, *Geometry of subanalytic and semialgebraic sets*. Springer Science & Business Media, 2012, vol. 150.
- [26] J. B. Lasserre, "Convexity in semialgebraic geometry and polynomial optimization," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1995–2014, 2009.
- [27] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.
- [28] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.
- [29] Y. Pan, Y. Kompis, L. Bartolomei, R. Mascaro, C. Stachniss, and M. Chli, "Voxfield: Non-projective signed distance fields for online planning and 3d reconstruction," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5331–5338.
- [30] J. Wang, T. Zhang, Q. Zhang, C. Zeng, J. Yu, C. Xu, L. Xu, and F. Gao, "Implicit swept volume sdf: Enabling continuous collision-free trajectory generation for arbitrary shapes," *arXiv preprint arXiv:2405.00362*, 2024.
- [31] S. LaValle, "Planning algorithms," *Cambridge University Press google scholar*, vol. 2, pp. 3671–3678, 2006.
- [32] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.
- [33] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.
- [34] M. Yamashita, K. Fujisawa, and M. Kojima, "Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0)," *Optimization Methods and Software*, vol. 18, no. 4, pp. 491–505, 2003.
- [35] M. Andersen, J. Dahl, and L. Vandenberghe, "Cvxopt: Convex optimization," *Astrophysics Source Code Library*, pp. ascl-2008, 2020.
- [36] P. Goulart, "Interior-point conic optimization with clarabel.jl," <https://oxfordcontrol.github.io/ClarabelDocs/stable/>, accessed: 2024-02-29.
- [37] B. O'Donoghue, "Operator splitting for a homogeneous embedding of the linear complementarity problem," *SIAM Journal on Optimization*, vol. 31, pp. 1999–2023, August 2021.
- [38] M. Garstka, M. Cannon, and P. Goulart, "Cosmo: A conic operator splitting method for convex conic problems," *Journal of Optimization Theory and Applications*, vol. 190, no. 3, pp. 779–810, 2021.
- [39] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. CRC press, 2002.
- [40] A. A. Ahmadi and A. Majumdar, "Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization," *SIAM Journal on Applied Algebra and Geometry*, vol. 3, no. 2, pp. 193–230, 2019.
- [41] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.
- [42] D. J. Webb and J. Van Den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 5054–5061.
- [43] Z. Šir and B. Jüttler, " c^2 hermite interpolation by pythagorean hodograph space curves," *Mathematics of Computation*, vol. 76, no. 259, pp. 1373–1391, 2007.
- [44] A. Bambade, F. Schramm, S. El Kazdadi, S. Caron, A. Taylor, and J. Carpentier, "Proxqp: an efficient and versatile quadratic programming solver for real-time robotics applications and beyond," 2023.