

Analytic Conditions for Differentiable Collision Detection in Trajectory Optimization

Jaitly, Akshay; Jha, Devesh K.; Ota, Kei; Shirai, Yuki

TR2025-150 October 23, 2025

Abstract

Optimization-based methods are widely used for computing fast, diverse solutions for complex tasks such as collision-free movement or planning in the presence of contacts. However, most of these methods require enforcing non-penetration constraints between objects, resulting in a non-trivial and computationally expensive problem. This makes the use of optimization-based methods for planning and control challenging. In this paper, we present a method to efficiently enforce non-penetration of sets while performing optimization over their configuration, which is directly applicable to problems like collision-aware trajectory optimization. We introduce novel differentiable conditions with analytic expressions to achieve this. To enforce non-collision between non-smooth bodies using these conditions, we introduce a method to approximate polytopes as smooth semi-algebraic sets. We present several numerical experiments to demonstrate the performance of the proposed method and compare the performance with other baseline methods recently proposed in the literature.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2025

Analytic Conditions for Differentiable Collision Detection in Trajectory Optimization

Akshay Jaitly¹, Devesh K. Jha², Kei Ota³, and Yuki Shirai²

Abstract—Optimization-based methods are widely used for computing fast, diverse solutions for complex tasks such as collision-free movement or planning in the presence of contacts. However, most of these methods require enforcing non-penetration constraints between objects, resulting in a non-trivial and computationally expensive problem. This makes the use of optimization-based methods for planning and control challenging. In this paper, we present a method to efficiently enforce non-penetration of sets while performing optimization over their configuration, which is directly applicable to problems like collision-aware trajectory optimization. We introduce novel differentiable conditions with analytic expressions to achieve this. To enforce non-collision between non-smooth bodies using these conditions, we introduce a method to approximate polytopes as smooth semi-algebraic sets. We present several numerical experiments to demonstrate the performance of the proposed method and compare the performance with other baseline methods recently proposed in the literature.

I. INTRODUCTION

Optimization-based approaches present an effective way to generate rich behavior for robots in the presence of various kinds of constraints [1], [2], [3]. These methods are widely used for planning trajectories of multi-body robotic systems in the presence of obstacles, or in the presence of contact constraints [4]. They are also used in various physics engines (e.g., MuJoCo [5], Bullet [6], Drake [7], Dojo [8], PhysX [9]) where simulation of contact dynamics is performed by first finding the contacts between objects, then solving a constrained optimization problem.

Central to these planning or simulation problems is the ability to compute a signed distance function between bodies which can later be used for downstream tasks like collision-free trajectory planning or simulating contact dynamics. Computation of distance functions tends to be computationally challenging. Oftentimes, evaluating constraints on collision or distance values is non-differentiable, leading to challenges when solving problems in various applications like computer graphics, robotics, video games, etc.

In this paper, we present a formulation that allows us to compute an approximate signed distance function between sets, which we denote as ‘Minimum-Offset-To-Touch’ (MOTT) (see Fig. 1). This can be used to enforce non-penetration constraints during trajectory optimization. Our method embeds computation of the signed distance function

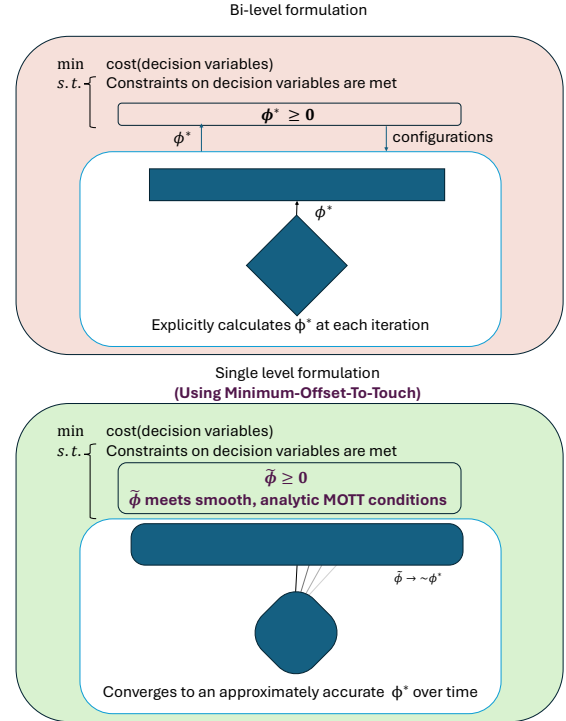


Fig. 1: Optimization-based methods for collision-aware trajectory optimization are traditionally bi-level (shown above in red), where a high level optimization problem calls a lower-level problem to calculate constrained values, like the signed distance between objects. In contrast, we propose Minimum-Offset-To-Touch (MOTT) conditions, which allow us to embed collision avoidance as smooth constraints with analytic form in the higher level problem, resulting in a single level optimization.

alongside the trajectory optimization problem. This is in contrast to the popular bi-level formulations, which require explicit calculation of distance at each solver iteration, as shown in Fig. 1. Other existing single-level optimization methods use a complementarity formulation (see [10], for example) to impose distance function constraints during optimization which are, in general, difficult to solve leading to longer solve times. In contrast, MOTT conditions have continuous, smooth gradients, which allow us to enforce constraints on distances between smooth sets with faster compute efficiency. Through several numerical experiments, we present the computational benefits offered by the proposed method.

Contributions.

- We introduce a ‘Minimum-Offset-To-Touch’ (MOTT) metric for the signed distance between collision bodies.
- MOTT conditions allow us to derive computationally

¹A. Jaitly is with the Robotics Engineering Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA akshayjaitly@hotmail.com

²D. K. Jha and Y. Shirai are with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA 02139 {jha, shirai}@merl.com

³K. Ota is with Mitsubishi Electric, Kanagawa, Japan Ota.Kei@ds.MitsubishiElectric.co.jp

efficient, differentiable nonlinear equality conditions (compared to non-smooth complementarity conditions used in other approaches). These conditions can be embedded in a single-level optimization problem to enforce non-penetration and perform signed distance computation while performing trajectory optimization.

- Additionally, we propose a method for finding smooth semi-algebraic approximations of (non-smooth) polytopic sets, used in conjunction with our MOTT conditions to simplify optimization problems considering non-penetration of polytopic bodies.

II. RELATED WORK

Collision detection and avoidance has been thoroughly studied in robotics literature. There are various methods to compute collisions between bodies [11], [12], [13]. For example, the classic Gilbert–Johnson–Keerthi (GJK) algorithm [11]. It computes the Minkowski difference of the two convex sets and estimates if the origin is inside this difference using iteratively computed simplices. The GJK algorithm and its variants [14] have shown impressively fast collision computation. Some of them are used in the popular Flexible Collision Library (FCL) [15] which is widely used in most physics engines. Another example is the Lin-Canny algorithm [16] which tracks the closest features (e.g., vertices, edges) between two convex polyhedra to check if collision is made. One of the issues for the aforementioned methods is that these solutions are not differentiable, which makes it difficult for gradient-based solvers (e.g., [17]) to effectively use them for planning. Recently, Tracy et al. proposed DCOL [18], which enables differentiable collision computation for convex primitives. For iterative trajectory optimization solvers, all of the above methods require explicit computation of the collision values at each iteration.

Our work is also related to the vast literature on collision-free motion planning in robotics. Sampling-based methods such as RRT [19] can efficiently plan collision-free trajectories. However, these methods do not consider kino-dynamic feasibility of generated motion plans. Model-free methods such as [20] can also design collision-free kinodynamically feasible trajectories, though training can be computationally demanding. Graphs of Convex Sets [21] and Polytopic Action Set And Motion Planning [22] form feasible motion planning as akin to a graph search problem. In these methods, polytopic sets of feasible collision free motions are strung together to form sets of feasible longer-horizon motions. These methods require significant pre-computation to build the polytopic sets, and may require solving Mixed-Integer programs to find motion.

More recently, in [23] (SILICO) the authors introduced a method to solve for the collision values alongside the optimal solution to an optimization problem in a single level. However, this is achieved by introducing a number of non-smooth complementarity constraints, which can present gradient-based solvers with difficulties in converging. In contrast, in this work, we present MOTT conditions, which are differentiable conditions with analytic expressions enforcing

collision detection which do not require enforcing complementarity constraints to design collision-free trajectories, making it much easier for the solvers to converge.

III. BACKGROUND

In this section, we present some relevant notation and background information which is useful in explaining the main results presented in this paper.

A. Optimization-Based Collision Detection

1) *Notation*: We first define some useful notation which is used throughout the paper:

- $\mathbf{a} \leq \mathbf{b}$ (resp. $\frac{\mathbf{a}}{\mathbf{b}}$) denotes an elementwise inequality (resp. elementwise division) between vectors $\mathbf{a}$ and $\mathbf{b}$.
- $\mathbb{R}_-^N$ denotes the non-positive cone, or $x \in \mathbb{R}^N$ such that $x \leq 0$. $\mathbb{R}_+^N$ denotes the nonnegative cone.
- $\mathcal{A}_i = \{\mathbf{x} \in \mathbb{R}^{N_x} | g_i(\mathbf{x}) \leq 0\}$ is a convex set where $g_i(\mathbf{x}) : \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_{g_i}}$ is a pre-defined function.
- The subscript $\cdot_{i,j}$ indicates that a value is associated with collision pair $\{\mathcal{A}_i, \mathcal{A}_j\}$.
- $\phi \in \mathbb{R}$ is a value associated with distance. $\phi_{i,j} \leq 0$ indicates that $\mathcal{A}_i$ and $\mathcal{A}_j$ are intersecting.

2) *Gilbert-Johnson-Keerthi Algorithm*: GJK [11] is a classic algorithm to compute collision between objects. The GJK algorithm effectively solves a ‘closest points problem’, similar to the following,

$$\min_{\mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i} \frac{1}{2} \|\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i\|^2 \quad \text{s.t.} \quad \mathbf{x}_{i,j}^i \in \mathcal{A}_i, \quad \mathbf{x}_{i,j}^j \in \mathcal{A}_j \quad (1)$$

If the distance between the closest points is 0, then $\exists x \in \mathcal{A}_i \cap \mathcal{A}_j$ and the intersection is non-empty, reflecting that they are in penetration. Alternatively, if the minima is positive, the intersection is empty and the objects are not in penetration.

3) *DCOL*: In work considering Differentiable Collision Detection (DCOL [18]), penetration is evaluated by considering the ‘minimum scaling’ amount that bodies must experience in order to be in contact. If they can be shrunk while preserving an intersection, the bodies are considered to be in collision. With α as the scaling factor, where $\alpha > 1$ relates to enlarging the set, $\alpha = 0$ shrinks the set to a point, and $\alpha_{i,j}^*$ is the minimum scaling to touch, the value associated with distance, $\phi_{i,j}$, can be evaluated as $\phi_{i,j} = \alpha_{i,j}^* - 1$.

4) *KKT Conditions*: In this work, we are concerned with conditions that test the validity of a Minimum-Offset-To-Touch solution (introduced later). For this, we will require an understanding of the KKT conditions for optimality.

The KKT conditions are sufficient and necessary conditions for a possible solution to a program like (1) to be optimal. The following are the KKT conditions of the closest points problem. Satisfying these conditions is equivalent to solving GJK.

$$\text{find } \mathbf{x}_{i,j}^i, \mathbf{x}_{i,j}^j, \lambda \quad (2a)$$

$$\text{s.t.} \quad (\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) + \nabla_{\mathbf{x}} \mathbf{g}_i(\mathbf{x}_{i,j}^i) \lambda_i = 0 \quad (2b)$$

$$-(\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) + \nabla_{\mathbf{x}} \mathbf{g}_j(\mathbf{x}_{i,j}^j) \lambda_j = 0 \quad (2c)$$



Fig. 2: Superquadratic approximation of a unit-hypercube in $\mathbb{R}^3$. As ρ increases, the approximation increases in accuracy.

$$g_i(\mathbf{x}_{i,j}^i) \leq 0, \quad g_j(\mathbf{x}_{i,j}^j) \leq 0 \quad (2d)$$

$$\lambda_i \geq 0, \quad \lambda_j \geq 0 \quad (2e)$$

$$\lambda_i^T g_i(\mathbf{x}_{i,j}^i) = 0, \quad \lambda_j^T g_j(\mathbf{x}_{i,j}^j) = 0 \quad (2f)$$

Here (2b) and (2c) are conditions of stationarity, (2d) are conditions of primal feasibility, (2e) are conditions of dual feasibility, and (2f) are conditions of complimentary slackness. λ are the dual variables in the KKT conditions.

B. Superquadrics

Our method utilizes smooth approximations of general polytopic sets. In order to do so, we employ 'superquadrics' (often referred to as superquadrics), seen in Fig. 2. Superquadrics have been used to generate surfaces, largely for use in graphics applications [24][25]. We will define superquadratic sets in N_y dimensions to be $\tilde{R}^{N_y} = \{y \in \mathbb{R}^{N_y} \mid \|y\|_{2\rho}^{2\rho} \leq 1\}$. $\|y\|_{2\rho}$ is the 2ρ norm of y , $(\sum_{k=0}^{N_y} (y_k^{2\rho}))^{\frac{1}{2\rho}}$.

Superquadrics with positive integer values of ρ can be visualized approximately as smooth, rounded hypercubes in $\mathbb{R}^{N_y}$. $\rho = 1$ results in a hyper-sphere of radius 1. As $\rho \rightarrow \infty$, the superquadratic exactly matches $\{-1 \leq y \leq 1\}$.

C. Trajectory Optimization

Trajectory optimization problems considered in this work are solved by iteratively updating a guess for an optimal, feasible, solution. In this strategy, constraints are evaluated at each iteration. These constraints can span considerations of dynamics, kinematics, collision avoidance, etc.

In bi-level strategies, the signed distance (ϕ) is algorithmically evaluated at each iteration. For a collision detection method like DCOL, this may require solving multiple convex programs at each iteration to evaluate the constraints. Single level strategies, on the other hand, may impose analytic constraints (often non-linear) on the decision variables that are only satisfied when ϕ is accurate. These strategies allow a mathematical program solver to converge towards ϕ values alongside the optimal values of other decision variables over the process of solving the optimization problem.

Prior works such as [18],[23] have proposed single-level methods where the KKT conditions of the minimum scaling to touch problem are embedded as constraints in another optimization problem. These require enforcing complimentary slackness, which results in optimization with $N_{g_i} + N_{g_j}$ complementarity constraints for contact pair $\{\mathcal{A}_i, \mathcal{A}_j\}$. In situations like trajectory optimization, where the configuration (q) of the collision objects is variable, $g_i(x, q)$ may

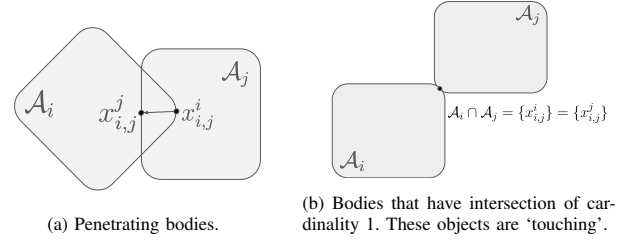


Fig. 3: The 'Minimum-Offset-To-Touch' distance is positive in the case of non-penetration, negative in the case of penetration, and 0 in the case that bodies are touching. Additionally shown are the points (x_i, x_j) that would be touching when the bodies are offset to touch.

be non-convex, resulting in harder to enforce non-convex complementarity constraints for existing single level collision detection methods.

IV. PROBLEM STATEMENT

In this section, we present our problem formulation. Our objective is to find a way to enforce constraints on the signed distance between objects easily.

We assume that the collision objects are closed, bounded, convex shapes $\mathcal{A}_i$ and $\mathcal{A}_j$ in N_x dimensions (similar to previous work such as DCOL [18]).

We define our signed distance metric to be a 'Minimum-Offset-To-Touch' (MOTT), a signed distance equivalent to the minimum amount that a body must be translated (without rotation) in any direction such that shapes are touching, but not overlapping. To touch, the intersection between translated objects must contain only points on the boundary of each set. We will consider non-flat surfaces, where this intersection is necessarily a single point. Examples of touching sets can be seen in Fig. 3b. Offsets required to touch can be seen in Fig. 3a and Fig. 4.

Formally, at an 'Offset-To-Touch',

- $\mathbf{a}$ is a unit vector indicating a 'direction of offset'.
- ϕ is the scalar valued signed offset distance.
- $(\mathcal{A}_i + \{\phi\mathbf{a}\}) \cap \mathcal{A}_j$ contains only points on the boundaries of each set.
- $\mathbf{x}_{i,j}^j \in \mathcal{A}_j$ is in the intersection of $\mathcal{A}_j$ and $\mathcal{A}_i + \{\phi\mathbf{a}\}$. This is a point at which the objects would be touching when $\mathcal{A}_i$ is offset by $\phi\mathbf{a}$.
- $\mathbf{x}_{i,j}^i \in \mathcal{A}_i$ is in the intersection of $\mathcal{A}_i$ and $\mathcal{A}_j - \{\phi\mathbf{a}\}$. This is a point at which the objects would be touching when $\mathcal{A}_j$ is offset by $-\phi\mathbf{a}$.
- $\phi\mathbf{a} = \mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i$

At the MOTT solution, $(\phi, \mathbf{a}, \mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i)$, sets are touching, and ϕ^2 is minimized.

V. METHOD

In this section, we derive sufficient conditions for $(\phi, \mathbf{a}, \mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i)$ to be a valid solution to the MOTT problem for shapes $\mathcal{A}_i$ and $\mathcal{A}_j$. We show that, for bodies with smooth surfaces, these are the smooth and algebraic conditions shown in Fig. 4. We additionally introduce a method for approximating polytopic objects as smooth semi-algebraic sets.

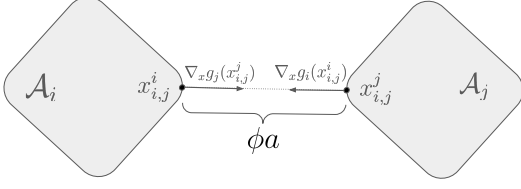


Fig. 4: The ‘Minimum-Offset-To-Touch’ conditions enforce that the surface normals ($\nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_i)$, $\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_j)$) between two convex bodies are in opposite directions, and that the direction of offset ($\mathbf{a}$) is along this surface normal.

A. Collision Detection Between Smooth Bodies

For collision detection, we specifically consider the case where $\mathcal{A}_i$ and $\mathcal{A}_j$ have smooth surfaces. For $\mathcal{A}_k$, we assume that $g_k(\mathbf{x})$ is a smooth, convex, scalar valued ($N_{g_k} = 1$) function.

1) *Conditions for touching:* Recall the definition for touching from Sec. IV. Not every pair $\{\mathbf{x}_{i,j}^i \in \mathcal{A}_i, \mathbf{x}_{i,j}^j \in \mathcal{A}_j\}$ results in a valid offset value $\phi\mathbf{a} = \mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i$ for touching. From our definition $\mathbf{x}_{i,j}^j$ is on the boundary of $\mathcal{A}_j$ and $\mathbf{x}_{i,j}^i = \mathbf{x}_{i,j}^j - \phi\mathbf{a}$ is on the boundary of $\mathcal{A}_i$. For scalar function valued constraints, this is defined as $g_j(\mathbf{x}_{i,j}^j) = 0$ and $g_i(\mathbf{x}_{i,j}^j - \phi\mathbf{a}) = g_i(\mathbf{x}_{i,j}^i) = 0$.

Lemma 1. ($\phi\mathbf{a} = \mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i$) defines a valid offset to touch if and only if

$$g_i(\mathbf{x}_{i,j}^i) = g_j(\mathbf{x}_{i,j}^j) = 0 \quad (3a)$$

$$\frac{\nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i)}{\|\nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i)\|} = -\frac{\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j)}{\|\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j)\|} \quad (3b)$$

As stated prior, under the assumptions we have made the intersection $(\mathcal{A}_i + \{\phi\mathbf{a}\}) \cap \mathcal{A}_j$ contains a single point on the boundaries of both shapes, reflected in condition (3a). Note that, in cases where sets are non-smooth (like polytopes) this intersection may be set valued, instead of a single point. Our assumption of smoothness omits this case.

By the separating hyperplane theorem [26], there necessarily exists at least one hyperplane tangent to both offset bodies at the contact point, which separates the interiors of the convex bodies. For smooth surfaces, this separating hyperplane is uniquely defined by the surface normal at the contacting point. This indicates that $\nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i)$ and $\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j)$, the surface normals of each shape at the contact point, are in opposing directions if and only if the shapes (offset by $\phi\mathbf{a}$, where (3a) is met) are touching, but the interiors are not overlapping. This gives (3b). Thus, we can prove that our conditions are met if $\phi\mathbf{a}$ is an offset to touch.

If the conditions hold, we can state that $\mathbf{x}_{i,j}^j = \mathbf{x}_{i,j}^i + \phi\mathbf{a}$ is in the intersection, and on the boundaries, of $\mathcal{A}_i + \{\phi\mathbf{a}\}$ and $\mathcal{A}_j$ (from (3a)) and that the interiors of the offset shapes are necessarily separated by the tangent hyperplane at $\mathbf{x}_{i,j}^j$.

Thus, any intersection is on the boundary of the shapes, and $\phi\mathbf{a}$ is an offset to touch.

We use $\nabla_{\mathbf{x}}\hat{\mathbf{g}}_k(\mathbf{x})$ to indicate the directionality of the gradient of g_k , which is the surface normal. In the following lemma, we present the main result of this paper.

Lemma 2. At the MOTT solution, $(\phi, \mathbf{a}, \mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i)$, the following conditions hold

$$g_i(\mathbf{x}_{i,j}^j) = 0, \quad g_i(\mathbf{x}_{i,j}^i) = 0 \quad (4a)$$

$$\nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i) = -\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j) \quad (4b)$$

$$(\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) = \phi\mathbf{a} = \phi\nabla_{\mathbf{x}}\hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^i) = -\phi\nabla_{\mathbf{x}}\hat{\mathbf{g}}_j(\mathbf{x}_{i,j}^j) \quad (4c)$$

Proof. Given the conditions for a valid ‘offset-to-touch’ $(\mathbf{x}_{i,j}^i, \mathbf{x}_{i,j}^j)$ the minimum offset to touch problem is the following:

$$\min_{\phi, \mathbf{a}, \mathbf{x}_{i,j}^i, \mathbf{x}_{i,j}^j} \frac{1}{2} \|\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i\|^2 \quad (5a)$$

$$\text{s.t.} \quad \phi\mathbf{a} = (\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) \text{ is a valid offset to touch} \quad (5b)$$

Or, using conditions (3a), (3b),

$$\min_{\mathbf{x}_{i,j}^i, \mathbf{x}_{i,j}^j} \frac{1}{2} \|\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i\|^2 \quad (6a)$$

$$\text{s.t.} \quad g_i(\mathbf{x}_{i,j}^i) = 0, \quad g_j(\mathbf{x}_{i,j}^j) = 0 \quad (6b)$$

$$\nabla_{\mathbf{x}}\hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^i) = -\nabla_{\mathbf{x}}\hat{\mathbf{g}}_j(\mathbf{x}_{i,j}^j) \quad (6c)$$

We wish to show that, if (4a) – (4c) hold for some solution, $(\phi, \mathbf{a}, \mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i)$, then $(\phi, \mathbf{a}, \mathbf{x}_{i,j}^j, \mathbf{x}_{i,j}^i)$ is an optimal solution to (6a) – (6c).

Consider the KKT conditions of (6a) – (6c),

$$\text{find } \mathbf{x}_{i,j}^i, \mathbf{x}_{i,j}^j \text{ s.t.} \quad (7a)$$

$$-(\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) + \lambda_1 \nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i) + \nabla_{\mathbf{x}}^2\mathbf{g}_i(\mathbf{x}_{i,j}^i)\lambda_3 = 0 \quad (7b)$$

$$(\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i) + \lambda_2 \nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j) + \nabla_{\mathbf{x}}^2\mathbf{g}_j(\mathbf{x}_{i,j}^j)\lambda_3 = 0 \quad (7c)$$

$$g_i(\mathbf{x}_{i,j}^i) = 0, \quad g_j(\mathbf{x}_{i,j}^j) = 0 \quad (7d)$$

$$\nabla_{\mathbf{x}}\hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^i) = -\nabla_{\mathbf{x}}\hat{\mathbf{g}}_j(\mathbf{x}_{i,j}^j) \quad (7e)$$

where $\nabla^2 g_k \in \mathbb{R}^{N_x \times N_x}$ indicates the Hessian of g_k . λ_1, λ_2 are arbitrary scalar free variables, and thus $\lambda_1 \nabla_{\mathbf{x}}\mathbf{g}_i(\mathbf{x}_{i,j}^i)$ is functionally equivalent to $\lambda_1 \nabla_{\mathbf{x}}\hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^i)$. Note that we are not required to enforce complementarity.

For $\lambda_3 = 0$, $\lambda_2 = \lambda_1 = \phi$, $\phi \nabla_{\mathbf{x}}\hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^i) = \mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i$, we can see that (7b) and (7c) are equivalent to our condition, (4c). (7d) and (7e) enforce primal feasibility, and directly reflect (4a), (4b). Thus, (4a) – (4c) are sufficient for enforcing the local optimality of a Minimum-Offset-To-Touch solution. $\square$

As indicated in Fig. 4, (4c) indicates that $(\mathbf{x}_{i,j}^j - \mathbf{x}_{i,j}^i)$ is ‘in the direction of’ the surface normal, $\nabla_{\mathbf{x}}\mathbf{g}_j(\mathbf{x}_{i,j}^j)$. Note that, for objects that are not in penetration, the MOTT problem is equivalent to the closest-points problem.

2) *Minimum Offset To Touch for Polytopes:* In the case that we do wish to continue considering polytopic objects, our intuition would still hold. In this case, however, the ‘surface normal’ would not be a unique vector value (as with our smooth body), but rather the normal cone at a point, $\mathbf{x}$. With abuse of notation, we state that (4c), (4b), the surface normal constraints in the MOTT conditions, would be the following

$$\mathbf{a} \in \nabla_{\mathbf{x}} \mathcal{A}_i(\mathbf{x}_{i,j}^i), \quad -\mathbf{a} \in \nabla_{\mathbf{x}} \mathcal{A}_j(\mathbf{x}_{i,j}^j) \quad (8)$$

Where $\nabla_{\mathbf{x}} \mathcal{A}(x)$ indicates the normal cone of $\mathcal{A}$ at x , or a weighted sum of the surface normals of all facets of $\mathcal{A}$ that x is a member of.

Intuitively, stationarity and complimentary slackness constraints in the KKT conditions effectively enforce these conditions. Stationarity ((2b), (2c)) enforces that $-\nabla_{\mathbf{x}} \mathbf{f} = \mathbf{a} = \mathbf{A}^T \lambda$ where $\mathbf{A}^T \lambda$ denotes the weighted sum of the vectors normal to the facets of the polytope. Complimentary slackness, $\lambda^T (\mathbf{A} \mathbf{x} - \mathbf{b}) = 0$ as seen in (2f), enforces that only the relevant subset of normal vectors are considered. Together, the conditions make $\nabla_{\mathbf{x}} \mathcal{A}(\mathbf{x}) = \{\mathbf{a} \mid \mathbf{a} = \mathbf{A}^T \lambda, \lambda^T (\mathbf{A} \mathbf{x} - \mathbf{b}) = 0\}$, the normal cone to the point at x . In considering scalar valued constraints, $g(\mathbf{x})$, we were able to make the assumption that $g(\mathbf{x}) = 0$ at the optimal solution and forgo the considerations of complimentary slackness that would traditionally be present in the KKT conditions of a similar problem. As there is only one ‘face’ of the set (the boundary), we can state that the surface normal is always in a singular direction.

3) *Non-uniqueness:* The Minimum-Scaling-To-Touch problem, however, has non-unique locally optimal solutions, including the Maximum-Offset-To-Touch.

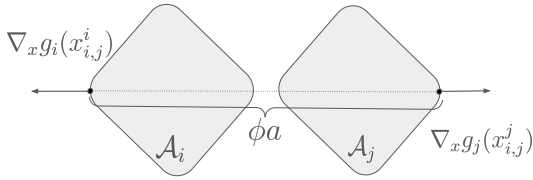


Fig. 5: This is a sub-optimal solution for Minimum-Offset-To-Touch, even though it satisfies the conditions for local optimality.

The example in Fig. 5 highlights an example of a ‘Maximum-Offset-To-Touch’ solution, which finds the furthest points in the collision bodies. This solution is locally optimal to the Minimum-Offset-To-Touch program, and thus satisfies our nonlinear equality MOTT conditions. At such a solution, ϕ is negative, even though the objects are not in penetration. As can be seen in Fig. 5, the surface normals are pointing in opposing directions, away from the interior of each set. For this solution to be a ‘Minimum-Offset-To-Touch’, the surface normals must point towards the interior of the opposing set.

So, we may consider an additional constraint of form

$$(\mathbf{c}_j - \mathbf{c}_i) \cdot \nabla_{\mathbf{x}} \mathbf{g}_i(\mathbf{x}_{i,j}^i) \geq 0 \quad (9)$$

where $\mathbf{c}_j$ is in the interior of $\mathcal{A}_i$ and $\mathbf{c}_i$ is in the interior of $\mathcal{A}_j$. This enforces that $\nabla_{\mathbf{x}} \mathbf{g}_i(\mathbf{x}_{i,j}^i)$ is in the appropriate direction.

This condition may not be strictly necessary in practical applications, however. For instance, if the ‘Minimum-Offset-To-Touch’ conditions are embedded in a trajectory optimization program enforcing non-penetration ($\phi \geq 0$), the ‘Maximum-Offset-To-Touch’ solution – where $\phi < 0$ – is inherently excluded from the set of feasible solutions without the inclusion of (9).

B. Smooth Approximations of Polyhedra

When deriving the conditions for ‘Minimum-Offset-To-Touch’, we made the assumption that each set considered was smooth, and that the constraint $g(\mathbf{x})$ was scalar valued. Non-smooth surfaces, and discontinuous gradients $\nabla_{\mathbf{x}} \hat{\mathbf{g}}(\mathbf{x})$, result in non-smooth (and complementarity) constraints when enforcing validity of an offset value, ϕ . This is not unique to our approach. As stated in V-A.2, approaches like SILICO [23] additionally require the enforcement of non-smooth complementarity constraints for non-smooth collision bodies. This is a restrictive assumption, however, as sets in question are not necessarily smooth. Instead, works have tended to approximate collision bodies with polytopes [27]. In this subsection, we introduce the use of superquadratics to approximate generic polytopic sets as semi-algebraic sets with smooth surfaces and continuous gradients.

1) *Smooth approximations by approximating the non-positive cone:* A linearly constrained set generally takes the form $\mathcal{A} = \{x \in \mathbb{R}^{N_x} \mid \mathbf{y} = \mathbf{A} \mathbf{x} - \mathbf{b}, \mathbf{y} \in \mathbb{R}_{-}^{N_y}\}$. We can understand a polyhedron to be the intersection between an affine subspace of $\mathbb{R}^{N_y}$ and the non-positive cone. Our approximation, $\tilde{\mathcal{A}}$, will depend on creating a smooth approximation of the non-positive cone,

$$\tilde{\mathcal{R}}_{-}^{N_y} = \{\mathbf{y} \mid g(\mathbf{y}) \leq 0\} \subseteq \mathbb{R}_{-}^{N_y}$$

where $g(\mathbf{y}) \in \mathbb{R}^1$. The intersection between the same affine subspace and this smooth approximation (as seen in Fig. 6), will yield a new set in $\mathbb{R}^{N_x}$ with a smooth boundary.

$\tilde{\mathcal{A}} = \{\mathbf{x} \in \mathbb{R}^{N_x} \mid \mathbf{y} = \mathbf{A} \mathbf{x} - \mathbf{b}, \mathbf{y} \in \tilde{\mathcal{R}}_{-}^{N_y}\}$ gives our smooth approximation of the original polytope.

We will create an approximation of the non-positive cone using superquadratics. As mentioned in III-B, we will consider superquadratics of the form $\|\mathbf{y}\|_{2\rho}^{2\rho} \leq 1$. As $\rho \rightarrow \infty$, $\|\mathbf{y}\|_{2\rho}^{2\rho} \leq 1$ becomes equivalent to $-1 \leq \mathbf{y} \leq 1$.

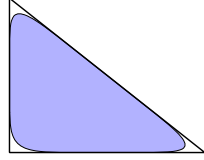
Thus, $\|\mathbf{y}\|_{2\rho}^{2\rho} \leq 1$ approximates a hypercube in the space $\mathbb{R}^{N_y}$, centered at the origin. To approximate the non-positive cone, we will use

$$\tilde{\mathcal{R}}_{-}^{N_y} = \{\mathbf{y} \in \mathbb{R}^{N_y} \mid \left\| 2 \frac{\mathbf{y}}{\bar{\mathbf{y}}} + 1 \right\|_{2\rho}^{2\rho} \leq 1\} \quad (10)$$

where $\bar{\mathbf{y}} \in \mathbb{R}^{N_y}$ and $\frac{\mathbf{y}}{\bar{\mathbf{y}}}$ represents an element-wise division.



(a) Approximation the surface of the non- (b) $\mathbf{y} = \mathbf{Ax} - \mathbf{b}$ gives a hyperplane, noted positive cone, noted in blue. in red.



(c) The approximated semi-algebraic set, in blue, overlaid on the original polytope, outlined in black.

Fig. 6: Superquadratic in the space $\mathbb{R}^3$. $\{\mathbf{y} = \mathbf{Ax} - \mathbf{b}\} \cap \mathbb{R}_-^{N_y}$ represents the original polytope and $\{\mathbf{y} = \mathbf{Ax} - \mathbf{b}\} \cap \tilde{\mathcal{R}}_-^{N_y}$ represents the approximated semi-algebraic set $\tilde{\mathcal{A}}$ (in blue).

2) *Choosing bounds for the superquadratic:* The resulting set approximates a hyper-rectangle within $\mathbb{R}_-^{N_y}$. The faces of the hyper-rectangle are congruent to the faces of the non-positive cone within the bounds $-\bar{\mathbf{y}} \leq \mathbf{y} \leq \mathbf{0}$. Thus,

$$\tilde{\mathcal{A}} = \{\mathbf{y} = \mathbf{Ax} - \mathbf{b}, \left\| 2\frac{\mathbf{y}}{\bar{\mathbf{y}}} + 1 \right\|_{2\rho} \leq 1\} \quad (11)$$

is a valid approximation for the original polytope when $-\bar{\mathbf{y}} \leq \mathbf{Ax} - \mathbf{b}$. $\bar{\mathbf{y}}$ must be chosen to be sufficiently large, such that $-\bar{\mathbf{y}} \leq \mathbf{Ax} - \mathbf{b} \quad \forall \mathbf{x} \in \{\mathbf{x} | \mathbf{Ax} - \mathbf{b} \leq \mathbf{0}\}$.

C. Trajectory Optimization

In this subsection, we present the use of MOTT conditions in a collision avoidance problem. Due to the existence of kinematic considerations when performing trajectory optimization, we consider $\{\mathbf{x}_k | g_k(\mathbf{x}_k, \mathbf{q}) \leq 0\}$ as the approximated collision body in the world frame, where $\mathbf{q}$ is the configuration of the system.

The program is as follows.

$$\min \quad f(\mathbf{q}^0, \mathbf{q}^1, \dots, \mathbf{q}^{T-1}) \quad (12)$$

$$\text{s.t.} \quad \mathbf{a}_{i,j}^t = \nabla_{\mathbf{x}} \hat{\mathbf{g}}_i(\mathbf{x}_{i,j}^{t,i}, \mathbf{q}^t) \quad (13)$$

$$\mathbf{a}_{i,j}^t = -\nabla_{\mathbf{x}} \hat{\mathbf{g}}_j(\mathbf{x}_{i,j}^{t,j}, \mathbf{q}^t) \quad (14)$$

$$\mathbf{x}_{i,j}^{t,j} - \mathbf{x}_{i,j}^{t,i} = \phi_{i,j}^t \mathbf{a}_{i,j}^t \quad (15)$$

$$g_i(\mathbf{x}_{i,j}^{t,i}, \mathbf{q}^t) = g_j(\mathbf{x}_{i,j}^{t,j}, \mathbf{q}^t) = 0 \quad (16)$$

$$\phi_{i,j}^t \geq 0 \quad (17)$$

$\mathbf{q}^t$ indicates the configuration at time step t , and $\mathbf{x}_{i,j}^{t,i}$ indicates the hypothetical contact point on $\mathcal{A}_i$ for pair $\{\mathcal{A}_i, \mathcal{A}_j\}$

in the world frame at time t . We enforce the constraints for each time step, for each contact pair.

(13) – (16) are our Minimum-Offset-To-Touch conditions, while (17) enforces non-penetration of bodies $\mathcal{A}_i$ and $\mathcal{A}_j$. As stated in section V-A.2, enforcing non-penetration between polytopic objects requires enforcing that $\mathbf{a}$ is in the normal cone, which requires enforcing complementarity conditions. As such, we instead consider approximated polytopes, $\tilde{\mathcal{A}}$, as the relevant collision geometry. We calculate the superquadratic based semi-algebraic approximations of the collision bodies a-priori, and utilize them to evaluate $g(\mathbf{x})$ and $\nabla_{\mathbf{x}} g(\mathbf{x})$. Our approximation ($\tilde{\mathcal{A}}_k$) is fully defined by $\bar{\mathbf{y}}_k$, A_k and b_k where A_k, b_k are inherent to the original polytope, $\mathcal{A}_k$. We solve $N_y + 1$ linear programs to find the bounds for $\bar{\mathbf{y}}$ (as detailed in V-B.2).

VI. RESULTS

In this section, we perform experiments to answer the following questions.

- Can enforcing MOTT conditions efficiently find the correct Minimum Offset To Touch?
- Can our trajectory optimizer find collision-free trajectories in cluttered environments efficiently?

A. Experiment Settings

We implement our framework in C++ using SNOPT [17] for solving (12). We implement our custom collision detection constraints within the Drake framework [7]. The computation is tested on a computer with Intel i7-1185G7.

We test the performance of our method in both, a trajectory optimization scenario and for distance computation in a scenario with static objects. We use a Coin-OR Linear Program (CLP) solver to find $\bar{\mathbf{y}}_k$ a-priori.

For trajectory optimization, we consider four distinct tasks with varying levels of complication in which we wish to facilitate collision free motion. We test our method, enforcing the Minimum-Offset-To-Touch conditions, and DCOL, recording the time it takes to solve. For our method, we also collect data on the average error in the solution ‘offset distance’.

The authors in [23] (SILICO) primarily utilize the technique within a scenario where the configuration is fixed, yielding a problem with Linear Complementarity constraints (an LCP). In trajectory optimization, this becomes a Nonlinear Complementarity problem, and is significantly harder to solve. Thus, we do not consider the single level trajectory optimization problem using SILICO to find distance values.

For each of the following, we set the initial guess by linearly interpolating between the initial and desired configurations. We attempt to minimize the distance from the goal at each of T determined knot points, and impose a constraint, limiting the maximum velocity per knot point.

B. Static Scene

First, we implement an experiment solving for the Minimum-Offset-To-Touch with a static scene, where the configuration is pre-determined. This experiment will allow

us to understand the time that a trajectory-optimizer may take to converge to an accurate solution for the Minimum-Offset-To-Touch, the approximation error, and how the two are related to ρ . We measure these values for 120 pairs of bodies randomly placed in space.

For pairs not in penetration, the % error is calculated as $\frac{\phi_{\text{solution}} - \phi_{\text{accurate}}}{\phi_{\text{accurate}}}$ where ϕ_{accurate} is the distance for the polytopes to touch (the GJK solution), and ϕ_{solution} is the offset for the smoothed approximated sets to touch.

The Minimum-Offset-To-Touch solution was used as an initial guess for the values of the offset-to-touch values.

TABLE I: The average solving time and accuracy for a program where we expect to find the Minimum-Offset-To-Touch, but not perform trajectory optimization.

ρ	Solve Time (ms)	average % error
1	24.30	18.34
2	98.37	10.39
3	58.48	6.81
4	76.24	4.93
5	29.30	3.81
6	65.31	3.08
7	32.25	2.80
8	53.45	2.21
9	54.85	1.93

We additionally tested DCOL on the same scene. The average time to solve for the minimum-scaling-to-touch problem was 1.41 ms.

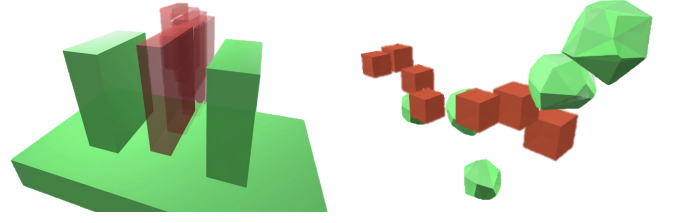
For the static scene, enforcing the Minimum-Offset-To-Touch conditions is not the ideal solution. This requires solving a program with nonlinear equality conditions, while solving for the Minimum Scaling To Touch constitutes a single quadratic program. The advantage of using the MOTT conditions, rather, is when an optimization has to be performed with multiple iterations, where the distance values change in each iteration.

We can also see in Table I, as expected, that as ρ increases the approximation error decreases. This error comes from the fact that our smooth semi-algebraic approximation is an inexact inner approximation of the original smooth collision bodies. As ρ increases, the gap between the approximated body and the original polytope decreases, and thus, as does the gap between the Minimum-Offset-To-Touch solutions. Additionally, the program solves for all considered values of ρ .

C. Trajectory Optimization

1) *Freebody test*: The first trajectory optimization task considers two freely articulated cuboids that fly past each other. This enforces non-penetration for 1 collision pair.

2) *Bookshelf Scene*: The second task is a book placing task. We consider a book to be a freely articulated body, and wish to place it on a shelf with bookends on either side (Fig. 7a). Linearly interpolating between the initial state and the desired final state results in a collision. We do not take into account dynamics considerations. This enforces non-collision for 3 collision pairs, with $\rho = 3$.



(a) a collision-free trajectory for placing a book (red) between two static objects (shown in green). (b) a collision-free trajectory for a drone (shown in green) maneuvering around the static obstacles (shown in red).

Fig. 7: We consider scenarios of varying difficulty. Above are examples of the ‘drone scene’ and the ‘bookshelf scene’.

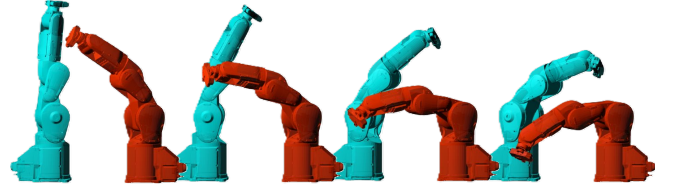


Fig. 8: Snapshots of arm movement obtained using the proposed model in collision-free trajectory optimization. We attempt to find a collision-free trajectory between pre-determined poses.

3) *Drone Scene*: The third task considers a drone (approximated as a cuboid) traversing through a messy scene with polytopic obstacles (Fig. 7b). We consider the drone to be a freely moving body with 6 Degrees Of Freedom. We set the scene such that linearly interpolating between the initial and final states results in collision. This enforces non-penetration for 5 collision pairs, with $\rho = 3$.

4) *Dual manipulator Scene*: In the final task, we consider a scene with two manipulators (Fig. 8), where the initial and final configurations are fixed. This enforces non-penetration for 15 collision pairs, with $\rho = 5$.

TABLE II: Solving times for trajectory optimization problems of varying difficulties. Enforcing the MOTT conditions is more efficient in problems that require more iterations, like the drone and dual manipulator scenes, as a solver converges to accurate distance values early.

	MOTT	DCOL
Freebody Scene	370.55 (ms)	107.52 (ms)
Bookshelf Scene	419.04 (ms)	111.94 (ms)
Drone Scene	2889.44 (ms)	12128.40 (ms)
Dual Manipulator Scene	30.415(s)	355.254(s)

5) *Trajectory Optimization Results*: We see from Table II that, for simpler problems which require less iterations to solve, the solver takes longer to converge to an accurate solution. It struggles to enforce the MOTT conditions while simultaneously taking large steps toward an optimal solution.

For larger problems, on the other hand, enforcing non-penetration using a bi-level formulation takes significantly longer. Using minimum scaling to touch requires the explicit computation of the scaling at each iteration, which can tend to be expensive when all collision pairs, and all iterations, are taken into account. Enforcing the MOTT conditions in

a single level requires that each iteration step of the SQP solver approximately preserve a valid value for ϕ .

VII. CONCLUSION

In this paper, we proposed Minimum-Offset-To-Touch (MOTT) conditions for distance computation between convex bodies. These conditions are differentiable and computationally inexpensive conditions with analytic expressions that can be used to find an accurate metric for signed distance between convex bodies. Assuming that collision bodies are convex and smooth, we derive and verify that the minimum offset for two bodies to touch occurs along the surface normal of each body, and derive MOTT conditions to enforce this property. Using these conditions, a single-level trajectory optimization is able to efficiently and robustly find locally optimal trajectories for rigid bodies. It is noted that our approach does not require enforcing complementarity constraints allowing faster and robust computation. Furthermore, we introduce a method to approximate non-smooth, polytopic, objects using semi-algebraic sets to approximately find distance between non-smooth objects. Our method is extensively evaluated over various planning scenarios and outperforms the baseline method for trajectory optimization problems.

We would like to extend this work to handle contact-rich manipulation tasks such as whole-body manipulation [28], [29] by incorporating contact dynamics as part of constraints. We would like to extend the use of the proposed method to model contact dynamics.

While we utilized this method to enforce non-penetration between physical objects in a trajectory optimization scenario, it can be used to enforce non-intersection of polytopes in higher dimensions as well. This can enable optimization methods to find polytopic inner approximations of non-convex sets, useful for methods like GCS [21] and Polytopic Action-Set And Motion Planning [22]. However, this is also left as a future exercise.

REFERENCES

- [1] "Picking Up Momentum — Boston Dynamics — bostondynamics.com," <https://bostondynamics.com/blog/picking-up-momentum/>, [Accessed 28-02-2025].
- [2] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM review*, vol. 59, no. 4, pp. 849–904, 2017.
- [3] E. Chandler, A. Jaitly, and M. Agheli, "Galileo: A pseudospectral collocation framework for legged robots," 2024. [Online]. Available: <https://arxiv.org/abs/2409.12465>
- [4] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Robotics Research*. Springer, 2020, pp. 985–1000.
- [5] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [6] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, 2015, p. 1.
- [7] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: <https://drake.mit.edu>
- [8] T. A. Howell, S. Le Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable simulator for robotics," *arXiv preprint arXiv:2203.00806*, vol. 9, no. 2, p. 4, 2022.
- [9] "PhysX SDK — developer.nvidia.com," <https://developer.nvidia.com/physx-sdk>, [Accessed 28-02-2025].
- [10] A. U. Raghunathan, D. K. Jha, and D. Romeres, "PyRoboCOP: Python-based robotic control and optimization package for manipulation and collision avoidance," *IEEE Transactions on Automation Science and Engineering*, pp. 1–16, 2024.
- [11] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [12] S. Cameron, "Enhancing gjk: Computing minimum and penetration distances between convex polyhedra," in *Proceedings of international conference on robotics and automation*, vol. 4. IEEE, 1997, pp. 3112–3117.
- [13] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3d game objects," in *Game developers conference*, vol. 170, 2001, p. 209.
- [14] L. Montaut, Q. Le Lidec, V. Petrik, J. Sivic, and J. Carpentier, "Gjk++: Leveraging acceleration methods for faster collision detection," *IEEE Transactions on Robotics*, 2024.
- [15] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3859–3866.
- [16] M. C. Lin and J. F. Canny, "A fast algorithm for incremental distance calculation," in *ICRA*, vol. 91, 1991, pp. 9–12.
- [17] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.
- [18] K. Tracy, T. A. Howell, and Z. Manchester, "Differentiable collision detection for a set of convex primitives," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3663–3670.
- [19] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.
- [20] G. Lin, L. Zhu, J. Li, X. Zou, and Y. Tang, "Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning," *Computers and Electronics in Agriculture*, vol. 188, p. 106350, 2021.
- [21] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science robotics*, vol. 8, no. 84, p. eadf7843, 2023.
- [22] A. Jaitly and S. Farzan, "Paamp: Polytopic action-set and motion planning for long horizon dynamic motion planning via mixed integer linear programming," 2024. [Online]. Available: <https://arxiv.org/abs/2403.10924>
- [23] S. Le Cleac'h, M. Schwager, Z. Manchester, V. Sindhwani, P. Florence, and S. Singh, "Single-level differentiable contact simulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4012–4019, 2023.
- [24] A. Jaklič, A. Leonardis, F. Solina, A. Jaklič, A. Leonardis, and F. Solina, "Superquadrics and their geometric properties," *Segmentation and recovery of superquadrics*, pp. 13–39, 2000.
- [25] P. Ferreira, "Sampling superquadric point clouds with normals," 2018. [Online]. Available: <https://arxiv.org/abs/1802.05176>
- [26] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [27] 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.
- [28] Y. Shirai, T. Zhao, H. Suh, H. Zhu, X. Ni, J. Wang, M. Simchowitz, and T. Pang, "Is linear feedback on smoothed dynamics sufficient for stabilizing contact-rich plans?" *2025 International Conference on Robotics and Automation (ICRA)*, 2025.
- [29] V. Leve, J. Moura, N. Saito, S. Tonneau, and S. Vijayakumar, "Explicit contact optimization in whole-body contact-rich manipulation," *arXiv preprint arXiv:2408.15726*, 2024.