

# A linear and Exact Algorithm for Whole-Body Collision Evaluation via Scale Optimization

Qianhao Wang<sup>†</sup>, Zhepei Wang<sup>†</sup>, and Fei Gao

**Abstract**—Collision evaluation is of vital importance in various applications. However, existing methods are either cumbersome to calculate or have a gap with the actual value. In this paper, we propose a zero-gap whole-body collision evaluation which can be formulated as a low dimensional linear program. This evaluation can be solved analytically in  $O(m)$  computational time, where  $m$  is the total number of the linear inequalities in this linear program. Moreover, the proposed method is efficient in obtaining its gradient, making it easy to apply to optimization-based applications.

## I. INTRODUCTION

Evaluation of collisions is critical in a variety of fields, such as obstacle avoidance in motion planning. In this paper, we propose a novel exact method that evaluates the collisions of two convex objects.

Simplex-based iterative approaches Gilbert-Johnson-Keerthi [1] (GJK) and enhancing GJK [2] have been widely used to calculate the distance between two convex objects. However, these algorithms require pre-processing to compute support functions. When two objects intersect, the computational cost of the GJK has to take the expanding polytope [3] algorithm (EPA) into account. Gilbert et al. [4] propose a growth distance to measure the separation and penetration of two objects. Similar to Gilbert’s work [4], Tracy et al. [5] calculate the minimum scale that both objects have to be enlarged or reduced simultaneously to achieve that there is only one intersection of them. They formulate the minimum scale computation as a conic problem solved by primal-dual interior-point methods, which is complex to solve and get gradient. However, these two methods do not support locality, which means that when we demand to evaluate the collision of a robot with many obstacles, these methods require the robot to perform calculation one by one with all the obstacles, not just with the obstacles within a certain range around it. Furthermore, the methods encounter problems with unstable values when one of the objects is much larger than the other one. Recently, Lutz et al. [6] propose a constructive solid geometry method based on two-layer LogSumExp functions. Although this method is easy to get the distance and distance gradient, it has a gap with the true value.

In this paper, we propose to evaluate collision by calculating the minimum scale at which one of the objects can be scaled to allow it to collide with another object. The proposed

<sup>†</sup> **Equal contribution.**

All authors are with the College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China, and also with the Huzhou Institute of Zhejiang University, Huzhou, 313000, China. Email: {qhwangaa, wangzhepei, fgaoaa}@zju.edu.cn

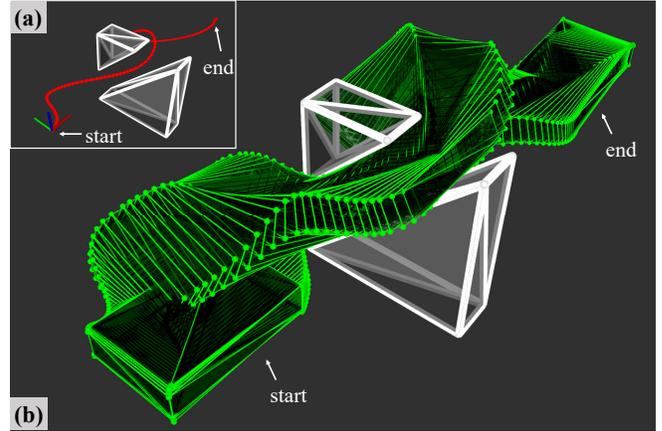


Fig. 1. This figure illustrates the application to a whole-body multicopter trajectory optimization problem which requires the multicopter to fly from the start to the end. (a): The red curve is the trajectory of the multicopter. (b): The white polyhedrons are the obstacles; the green rectangles are the snapshots of the multicopter which flies along the red trajectory in (a) to cross the narrow gap built by the two obstacles.

method has several strengths: **1)** This method is exact, do not have gap with the true value. **2)** In contrast to the above mentioned methods [4], [5] of zooming in or out of the two objects simultaneously, this method support locality. It means that even if there are many obstacles in the environment, this method only needs to focus on a small number of obstacles around the robot body, which can be achieved simply by filtering the obstacles with a bounding-box or ball. **3)** We will formulate the evaluation method into a low dimensional linear program whose computational time is  $O(m)$  [7], where  $m$  is the total number of the linear inequalities in the linear program. With the active constraints of the linear program, we can calculate the gradient of the scale with respect to the ego motion of the scaled object analytically. **4)** Additionally, it offers support conveniently for both objects represented by points and objects represented by surfaces, which then we will abbreviate with V-representation and H-representation. It is worth emphasising that our method can be applied directly, without pre-processing, even for objects represented by redundant points or surfaces.

## II. PROBLEM DEFINITION

We will present the problem definition of calculating the minimum scale with V-representation and H-representation respectively. For ease of presentation, we will refer to the scaled object as the **body** and to the other object as the **obstacle** below.

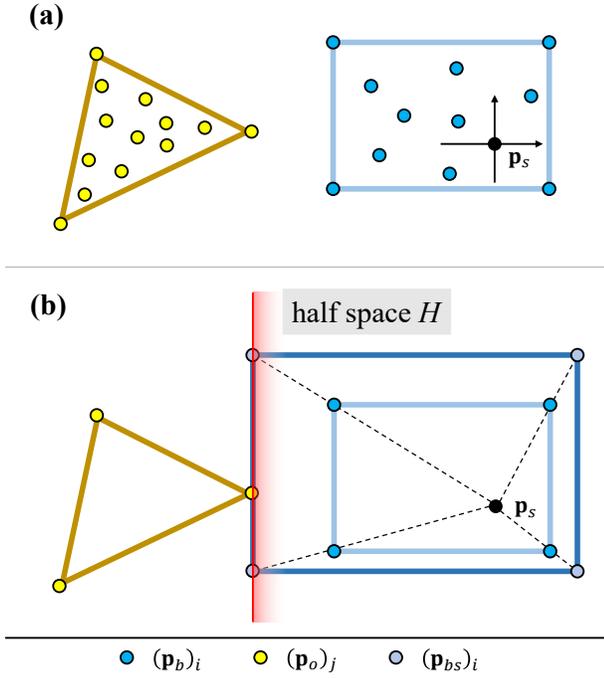


Fig. 2. This figure illustrates the problem definition in V-representation. (a): The body and the obstacle are defined by the yellow and blue convex hull respectively. We use the yellow and blue redundant points to represent the body and the obstacle respectively. (b): To prevent confusion in the visualisation, the redundant points are not illustrated. The light blue points represent the scaled body and the red line denotes the half space containing the scaled body but not the obstacle.

### A. V-representation

In V-representation, as shown in Fig. 2(a), both the yellow and blue objects are defined by a convex hull that can contain the redundant points. However, instead of processing points into convex hull, the proposed method is capable of working directly on the redundant points. It is essential to highlight that for clarity of display, we do not visualise points that are redundant for representing object in Fig. 2(b).

As illustrated in Fig 2(a), we use point sets  $P_{body}^b = \{(\mathbf{p}_b^b)_i | i = 1, 2, \dots, n_b\}$  and  $P_{obs}^b = \{(\mathbf{p}_o^b)_j | j = 1, 2, \dots, n_o\}$  to represent body and obstacle in the body frame. We define a point  $\mathbf{p}_s$  in the body frame as scale seed point, which the body point sets scale about. Then we get obstacle and scaled body point sets in a coordinate system with the scale seed point  $\mathbf{p}_s$  as the origin as

$$\begin{aligned} P_{body}^s(\beta) &= \{(\mathbf{p}_{bs}^s)_i = \beta((\mathbf{p}_b^b)_i - \mathbf{p}_s^b) | i = 1, 2, \dots, n_b\}, \\ P_{obs}^s &= \{(\mathbf{p}_o^s)_j = (\mathbf{p}_o^b)_j - \mathbf{p}_s^b | j = 1, 2, \dots, n_o\}, \end{aligned} \quad (1)$$

where  $\beta \in R^+$  is the scale.

Then we define the problem of calculating the minimum scale in V-representation as to maximize the scale  $\beta$  with the constraints of

$$\begin{cases} \alpha_o^T \beta ((\mathbf{p}_b^b)_i - \mathbf{p}_s^b) \leq 1, & i = 1, 2, \dots, n_b \\ \alpha_o^T ((\mathbf{p}_o^b)_j - \mathbf{p}_s^b) \geq 1, & j = 1, 2, \dots, n_o \end{cases} \quad (2)$$

which means a half space  $H = \{x | \alpha_o^T x \leq 1\}$  is required so

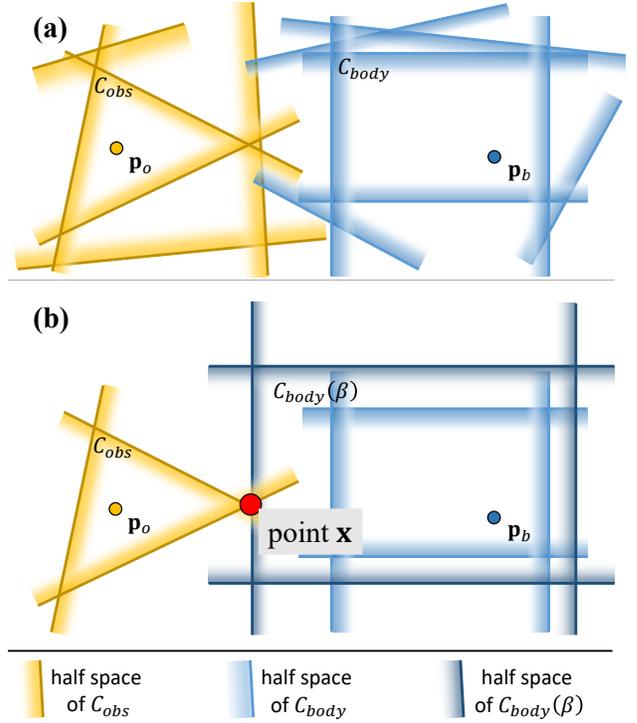


Fig. 3. This figure illustrates the problem definition in H-representation. (a): The body and the obstacle are represented by intersections of several redundant yellow and blue half spaces respectively. (b): For better visualisation, we do not show the half spaces that are redundant for the representation. The dark blue points represent the scaled body and the red point denotes the point that belongs to both the scaled body and the obstacle.

that the scaled body  $P_{body}^s(\beta)$  is inside the half space and the obstacle  $P_{obs}^s$  is outside it, as shown in Fig. 2(b).

We define  $\alpha^T = \beta \alpha_o^T$ . Since  $\beta \in R^+$ , finally, we can formulate the scale calculation problem as a low dimension LP problem:

$$\begin{aligned} \max \quad & \beta \\ \text{s.t.} \quad & \begin{cases} \alpha^T ((\mathbf{p}_b^b)_i - \mathbf{p}_s^b) \leq 1, & i = 1, 2, \dots, n_b \\ \alpha^T ((\mathbf{p}_o^b)_j - \mathbf{p}_s^b) \geq 1, & j = 1, 2, \dots, n_o \end{cases} \end{aligned} \quad (3)$$

### B. H-representation

In H-representation, as illustrated in Fig. 3(a), we use the intersection of several redundant half spaces to represent a convex polyhedron. It needs to be stressed that for clear display, the redundant half spaces are not shown in Fig. 3(b).

As indicated in Fig. 3(a), we use  $C_{body} = \{\mathbf{x} | (\alpha_b)_i^T (\mathbf{x} - \mathbf{p}_b) \leq 1, i = 1, 2, \dots, n_b\}$  and  $C_{obs} = \{\mathbf{x} | (\alpha_o)_j^T (\mathbf{x} - \mathbf{p}_o) \leq 1, j = 1, 2, \dots, n_o\}$  to represent body and obstacle in the world frame.  $\mathbf{p}_b$  and  $\mathbf{p}_o$  are points inside body and obstacle respectively.  $n_b$  and  $n_o$  are the number of half spaces whose intersection define body and obstacle. We make the body scale about  $\mathbf{p}_b$ , then the scaled body can be written as

$$C_{body}(\beta) = \{\mathbf{x} | (\alpha_b)_i^T (\mathbf{x} - \mathbf{p}_b) \leq \beta, i = 1, 2, \dots, n_b\}, \quad (4)$$

where  $\beta \in R^+$  is the scale.

Referring to Eq.2 and Eq.3, we define the problem in H-representation as

$$\begin{aligned} & \max \beta \\ & \text{s.t.} \quad \begin{cases} (\alpha_b)_i^T (\mathbf{x} - \mathbf{p}_b) \leq \beta, & i = 1, 2, \dots, n_b \\ (\alpha_o)_j^T (\mathbf{x} - \mathbf{p}_o) \leq 1, & j = 1, 2, \dots, n_o \end{cases} \end{aligned} \quad (5)$$

which is obviously a low dimension LP problem. And the constraints of Eq.5 means that a point  $\mathbf{x}$  which satisfies  $\mathbf{x} \in C_{body}(\beta) \cap C_{osb}$  is required as illustrated in Fig. 3(b).

### III. GRADIENT COMPUTATION

Since both problems we defined with V-representation and H-representation are in the form of LP, we only derive the sub-gradient in V-representation in this paper and the idea of gradient computation in H-representation is similar.

We use the active constraints of the low dimension LP problem to get the gradient of the scale  $\beta$ . In  $n$ -dimensional environment, based on the LP problem definition in Eq.3, there are two different kinds of active constraints which can be written in the form of linear equations:

$$\begin{aligned} & \left[ \left( (\mathbf{p}_{b.ac}^b)_i - \mathbf{p}_s^b \right)^T \quad 0 \right] \begin{bmatrix} \alpha_{n \times 1} \\ \beta \end{bmatrix} = 1, \quad i = 1, 2, \dots, n_{b.ac} \\ & \left[ \left( (\mathbf{p}_{o.ac}^b)_j - \mathbf{p}_s^b \right)^T \quad -1 \right] \begin{bmatrix} \alpha_{n \times 1} \\ \beta \end{bmatrix} = 0, \quad j = 1, 2, \dots, n_{o.ac} \end{aligned} \quad (6)$$

where  $(\mathbf{p}_{b.ac}^b)_i \in P_{body}^b$ ,  $(\mathbf{p}_{o.ac}^b)_j \in P_{obs}^b$  and  $n_{b.ac} + n_{o.ac} = n + 1$ , which means we should have  $n + 1$  active constraints to solve the LP problem.

We refer to  $(\mathbf{p}_{b.ac}^b)_i$  and  $(\mathbf{p}_{o.ac}^b)_j$  as the active constraint point on the body and the obstacle respectively. In Fig. 4, We show the result of the minimum scale calculation defined in V-representation. As indicated in Fig. 4(b), the red points in the green and white convex hull are the  $(\mathbf{p}_{b.ac}^b)_i$  and  $(\mathbf{p}_{o.ac}^b)_j$  separately. Additionally, the points in the scaled body obtained from the  $(\mathbf{p}_{b.ac}^b)_i$  according to Eq. 1 and  $(\mathbf{p}_{o.ac}^b)_j$  are on the hyperplane  $P = \{x | \alpha^T x \leq \beta\}$  which is represented in Fig. 4(b) by the blue plane.

We combine all the linear equations in Eq. 6 and write them in matrix form as

$$\begin{bmatrix} \left( (\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b \right)^T & 0 \\ \vdots & \vdots \\ \left( (\mathbf{p}_{b.ac}^b)_{n_{b.ac}} - \mathbf{p}_s^b \right)^T & 0 \\ \left( (\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b \right)^T & 1 \\ \vdots & \vdots \\ \left( (\mathbf{p}_{o.ac}^b)_{n_{o.ac}} - \mathbf{p}_s^b \right)^T & 1 \end{bmatrix} \begin{bmatrix} \alpha_{n \times 1} \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (7)$$

Then we block this matrix equation in Eq. 7 in into

$$\begin{bmatrix} \mathbf{A}_{n \times n} & \mathbf{B}_{n \times 1} \\ \mathbf{C}_{1 \times n} & \mathbf{D}_{1 \times 1} \end{bmatrix} \begin{bmatrix} \alpha_{n \times 1} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{E}_{n \times 1} \\ \mathbf{F}_{1 \times 1} \end{bmatrix}, \quad (8)$$

which can be written into equation group form:

$$\begin{cases} \mathbf{A}_{n \times n} \alpha_{n \times 1} + \mathbf{B}_{n \times 1} \beta = \mathbf{E}_{n \times 1} \\ \mathbf{C}_{1 \times n} \alpha_{n \times 1} + \mathbf{D}_{1 \times 1} \beta = \mathbf{F}_{1 \times 1} \end{cases}, \quad (9)$$

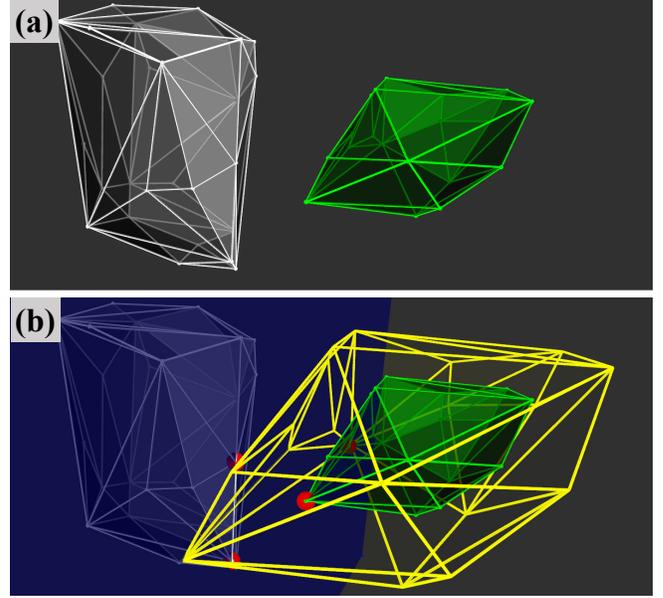


Fig. 4. This figure illustrates the result of calculating the minimum scale defined in V-representation. (a): The white and the green convex hull denote the obstacle and the body respectively. (b): The yellow convex hull is the scaled body obtained by enlarging the body to the minimum scale. The blue plane is the hyperplane splitting the scaled body and the obstacle, and the red points represent the active constraint points which satisfy one of the linear equations in Eq. 6. Although the objects are visualized using convex hull in this figure, the problem defined in V-representation is applied directly to the redundant point sets during the calculation.

where the variable  $\alpha_{n \times 1}$  can be eliminated and get equation which only have one variable  $\beta$ :

$$\mathbf{C}_{1 \times n} \mathbf{A}_{n \times n}^{-1} (\mathbf{E}_{n \times 1} - \mathbf{B}_{n \times 1} \beta) + \mathbf{D}_{1 \times 1} \beta = \mathbf{F}_{1 \times 1}. \quad (10)$$

For a rigid body, the point set  $P_{body}^b$  in body frame is not related to the motion of the body. As for  $P_{obs}^b$ , in practice we can only directly obtain the obstacle's point set  $P_{obs}^w$  in world frame. In order not to lose generality, we define the centre of rotation of the rigid body in the body frame as  $\mathbf{p}_{cen}^b$ , and define  $\mathbf{R}$  and  $\mathbf{t}$  for the rotation and translation of the body in the world frame. Based on the body's motion and  $P_{obs}^w$ , we can get the point in  $P_{obs}^b$  as

$$(\mathbf{p}_o^b)_j = \mathbf{R}^{-1} ((\mathbf{p}_o^w)_j - \mathbf{t} - \mathbf{p}_{cen}^b), \quad (11)$$

Then we can conveniently use the implicit function in Eq. 10, which contains the relationship between the scale and the motion of the rigid body, to obtain the partial derivatives of  $\beta$  with respect to  $\mathbf{R}$  and  $\mathbf{t}$ .

### IV. APPLICATION

To demonstrate the usability of the method proposed in this paper, we apply the method to a whole-body multicopter trajectory optimization problem, based on the differential flatness of multicopter in Wang's work [8]. As shown in Fig. 1, we model the multicopter with a green rectangle and require it to fly through a narrow slit consisting of two white obstacles. We adopt segmented polynomial to represent the flat-output trajectory, of which we use the MINCO [8] to

conduct spatiotemporal deformation. In this SE(3) trajectory optimization, we achieve multicopter whole-body obstacle avoidance by requiring the minimum scale defined in Sec. II greater than 1, which needs the gradient of the scale with respect to the motion of the multicopter.

#### A. Gradient Computation for SE(3) Motion

We derive the calculation of the gradient in detail. As defined in Sec. III, we use  $\mathbf{R}$  and  $\mathbf{t}$  to represent the rotation matrix and translation of the body. For convenience, in optimization we use a normalized quaternion  $\mathbf{q} = [w, x, y, z]^T$  to represent rotation. Referring to [9], the rotation matrix  $\mathbf{R}$  can be expressed by the quaternion  $\mathbf{q}$  as

$$\mathbf{R} = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & 1 - 2(x^2 + z^2) & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & 1 - 2(x^2 + y^2) \end{bmatrix}, \quad (12)$$

based on which, the partial derivatives  $\frac{\partial \mathbf{R}}{\partial \mathbf{q}_*}$ ,  $*$  =  $\{w, x, y, z\}$  can easily be obtained.

In this case where  $n = 3$ , as mentioned in Sec. III, the LP problem which defined in Sec. II-A should have 4 active constraints. The specific situations of  $\mathbf{ac}$  (active constraints) can be divided into three types:

- 3  $\mathbf{ac} \in P_{body}^b$ , 1  $\mathbf{ac} \in P_{obs}^b$ ;
- 2  $\mathbf{ac} \in P_{body}^b$ , 2  $\mathbf{ac} \in P_{obs}^b$ ;
- 1  $\mathbf{ac} \in P_{body}^b$ , 3  $\mathbf{ac} \in P_{obs}^b$ .

We then analyse each type in turn:

1) 3  $\mathbf{ac} \in P_{body}^b$ , 1  $\mathbf{ac} \in P_{obs}^b$ : In this case, the block matrix equation in Eq.8 can be written as

$$\begin{bmatrix} \begin{bmatrix} ((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_2 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_3 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \alpha_{3 \times 1} \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \quad (13)$$

Then based on Eq.10, we can get

$$\beta = ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \begin{bmatrix} ((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_2 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_3 - \mathbf{p}_s^b)^T \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (14)$$

where only  $(\mathbf{p}_{o.ac}^b)_1$  is related to  $\mathbf{R}$  and  $\mathbf{t}$ . Based on Eq.11, for brevity, we write this equation in Eq. 14 as

$$\begin{aligned} \beta &= ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \mathbf{A}^{-1} \mathbf{E}, \\ &= (\mathbf{R}^{-1} ((\mathbf{p}_{o.ac}^w)_1 - \mathbf{t} - \mathbf{p}_{cen}^b) - \mathbf{p}_s^b)^T \mathbf{A}^{-1} \mathbf{E}, \end{aligned} \quad (15)$$

where  $\mathbf{A}$  and  $\mathbf{E}$  are defined in Eq. 8. In this case,  $\mathbf{A}$  and  $\mathbf{E}$  are constant. Then we can get the gradient of  $\beta$  w.r.t.  $\mathbf{t}$  as

$$\frac{\partial \beta}{\partial \mathbf{t}} = -\mathbf{R} \mathbf{M}. \quad (16)$$

And we can get the gradient of  $\beta$  w.r.t.  $\mathbf{q}$  as

$$\frac{\partial \beta}{\partial \mathbf{q}_*} = -(\mathbf{p}_{o.ac}^w)_1^T \frac{\partial \mathbf{R}^T}{\partial \mathbf{q}_*} \mathbf{R} \mathbf{M}. \quad (17)$$

2) 2  $\mathbf{ac} \in P_{body}^b$ , 2  $\mathbf{ac} \in P_{obs}^b$ : In this case, the block matrix equation in Eq.8 can be written as

$$\begin{bmatrix} \begin{bmatrix} ((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_2 - \mathbf{p}_s^b)^T \\ (\Delta \mathbf{p}_{o.ac}^b)^T \\ ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \alpha_{3 \times 1} \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad (18)$$

where the  $\mathbf{ac}$  corresponding to  $\Delta \mathbf{p}_{o.ac}^b$  is

$$\begin{aligned} ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \alpha - \beta &= 0, \\ ((\mathbf{p}_{o.ac}^b)_2 - \mathbf{p}_s^b)^T \alpha - \beta &= 0, \end{aligned} \quad (19)$$

which can be combined to obtain

$$((\mathbf{p}_{o.ac}^b)_1^T - (\mathbf{p}_{o.ac}^b)_2^T) \alpha = (\Delta \mathbf{p}_{o.ac}^b)^T \alpha = 0. \quad (20)$$

Based on Eq.11  $\Delta \mathbf{p}_{o.ac}^b$  can be written as

$$\Delta \mathbf{p}_{o.ac}^b = \mathbf{R}^{-1} ((\mathbf{p}_o^w)_1 - \mathbf{p}_o^w)_2. \quad (21)$$

Then based on Eq.10, we can get

$$\beta = ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \begin{bmatrix} ((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_2 - \mathbf{p}_s^b)^T \\ (\Delta \mathbf{p}_{o.ac}^b)^T \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad (22)$$

which, for brevity, we write as

$$\beta = \mathbf{C} \mathbf{A}^{-1} \mathbf{E}, \quad (23)$$

where  $\mathbf{C}$ ,  $\mathbf{A}$  and  $\mathbf{E}$  are defined in Eq. 8. In this case,  $\mathbf{C}$  is related to  $\mathbf{R}$  and  $\mathbf{t}$ ,  $\mathbf{A}$  is only related to  $\mathbf{R}$ , and  $\mathbf{E}$  is a constant matrix.

Then we can get the gradient of  $\beta$  w.r.t.  $\mathbf{t}$  as

$$\frac{\partial \beta}{\partial \mathbf{t}} = -\mathbf{R} \mathbf{A}^{-1} \mathbf{E}. \quad (24)$$

And we can get the gradient of  $\beta$  w.r.t.  $\mathbf{q}$  as

$$\begin{aligned} \frac{\partial \beta}{\partial \mathbf{q}_*} &= -(\mathbf{p}_{o.ac}^b)_1^T \frac{\partial \mathbf{R}^T}{\partial \mathbf{q}_*} \mathbf{R} \mathbf{A}^{-1} \mathbf{E} \\ &+ \mathbf{C} \mathbf{A}^{-1} \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 3} \\ (\Delta \mathbf{p}_{o.ac}^b)^T \frac{\partial \mathbf{R}^T}{\partial \mathbf{q}_*} \mathbf{R} \end{bmatrix} \mathbf{A}^{-1} \mathbf{E}. \end{aligned} \quad (25)$$

3) 1  $\mathbf{ac} \in P_{body}^b$ , 3  $\mathbf{ac} \in P_{obs}^b$ : In this case, the block matrix equation in Eq.8 can be written as

$$\begin{bmatrix} \begin{bmatrix} ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{o.ac}^b)_2 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{o.ac}^b)_3 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \end{bmatrix} & \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \alpha_{3 \times 1} \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (26)$$

Then based on Eq.10, we can get

$$-((\mathbf{p}_{b.ac}^b)_1 - \mathbf{p}_s^b)^T \begin{bmatrix} ((\mathbf{p}_{o.ac}^b)_1 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{o.ac}^b)_2 - \mathbf{p}_s^b)^T \\ ((\mathbf{p}_{o.ac}^b)_3 - \mathbf{p}_s^b)^T \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \beta = 1, \quad (27)$$

which, for brevity, we write as

$$-\mathbf{C} \mathbf{A}^{-1} \mathbf{B} \beta = 1, \quad (28)$$

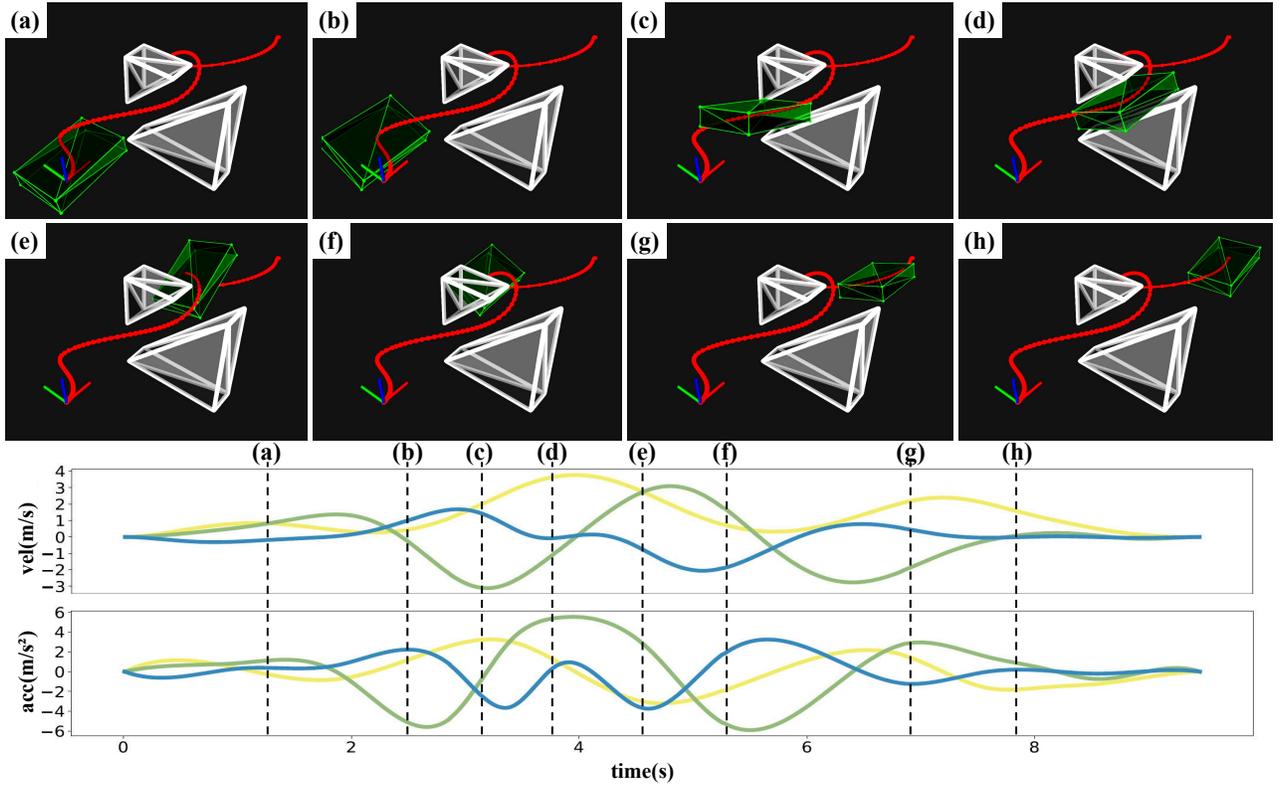


Fig. 5. Experimental validation of our method through a SE(3) whole-body multicopter trajectory optimization, where the multicopter is required to cross a narrow gap made up of two white obstacles. **Top:** The visualization of the experiment. The red curve denotes the trajectory generated by the proposed method. The white objects are the obstacle and the green rectangles in each small figure represent the multicopter which shows the process of the SE(3) flight. **Bottom:** Visualisation of velocity and acceleration of the executed trajectory.

where  $\mathbf{C}$ ,  $\mathbf{A}$  and  $\mathbf{B}$  are defined in Eq. 8.  $\mathbf{C}$  and  $\mathbf{B}$  are constant when  $\mathbf{ac}$  is determined in this case. Only  $\mathbf{A}$  is related to  $\mathbf{R}$  and  $\mathbf{t}$ .

Then we can get the gradient of  $\beta$  w.r.t.  $\mathbf{t}$  as

$$\mathbf{CA}^{-1}\mathbf{B}\frac{\partial\beta}{\partial\mathbf{t}.x} + \mathbf{CA}^{-1}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\mathbf{RA}^{-1}\mathbf{B}\beta = 0. \quad (29)$$

Based Eq.28, the gradient can be written as

$$\frac{1}{\beta}\frac{\partial\beta}{\partial\mathbf{t}.x} + \begin{bmatrix} \frac{1}{\beta} & 0 & 0 \end{bmatrix}\mathbf{RA}^{-1}\mathbf{B}\beta = 0. \quad (30)$$

Finally we obtain the gradient as

$$\frac{\partial\beta}{\partial\mathbf{t}} = -\beta\mathbf{RA}^{-1}\mathbf{B}. \quad (31)$$

And we can get the gradient of  $\beta$  w.r.t.  $\mathbf{q}$  as

$$\frac{1}{\beta}\frac{\partial\beta}{\partial\mathbf{q}.*} - \mathbf{CA}^{-1}\frac{\partial\mathbf{A}}{\partial\mathbf{q}.*}\mathbf{A}^{-1}\mathbf{B}\beta = 0, \quad (32)$$

which can be written as

$$\frac{\partial\beta}{\partial\mathbf{q}.*} = \mathbf{CA}^{-1}\frac{\partial\mathbf{A}}{\partial\mathbf{q}.*}\mathbf{A}^{-1}\mathbf{B}\beta^2, \quad (33)$$

where  $\frac{\partial\mathbf{A}}{\partial\mathbf{q}.*}$  can be get by

$$\frac{\partial\mathbf{A}}{\partial\mathbf{q}.*} = -\begin{bmatrix} (\mathbf{P}_{o.ac}^b)_1^T \\ (\mathbf{P}_{o.ac}^b)_2^T \\ (\mathbf{P}_{o.ac}^b)_3^T \end{bmatrix}\frac{\partial\mathbf{R}^T}{\partial\mathbf{q}.*}\mathbf{R}, \quad (34)$$

## B. Experiment Result

L-BFGS<sup>1</sup> [10] is adopted as an efficient quasi-Newton method to solve the numerical optimization problem. We use Lewis-Overton line search [11] that supports nonsmooth functions to deal with the nonsmoothness of the scale, which sometimes occurs during optimization. As the optimization result shows in Fig. 5, the SE(3) whole-body trajectory generated by our method is collision free and smooth.

## V. CONCLUSION AND FUTURE WORK

In this paper, we propose a exact whole-body collision formulation via linear scale, which can be solved efficiently. Furthermore, we derive the calculation of its gradient and applied it to a 3-D trajectory optimization problem.

In addition to the multicopters mentioned in Sec. IV, the proposed method can be applied to other mobile robots, such as vehicles and robotic arms. Benefiting from its scale-based design, this method can be implemented for deformable robots and swarm formations as well. The above applications of this method will be released in the near future. It is also worth mentioning that we will consider continuous collision formulation in the trajectory optimization to improve the completeness of planning.

<sup>1</sup><https://github.com/ZJU-FAST-Lab/LBFGS-Lite>

## REFERENCES

- [1] E. G. Gilbert, D. W. Johnson, and S. 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.
- [2] 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.
- [3] G. Van Den Bergen, *Collision detection in interactive 3D environments*. CRC Press, 2003.
- [4] E. G. Gilbert and C. J. Ong, "New distances for the separation and penetration of objects," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 579–586.
- [5] K. Tracy, T. A. Howell, and Z. Manchester, "Differentiable collision detection for a set of convex primitives," *arXiv preprint arXiv:2207.00669*, 2022.
- [6] M. Lutz and T. Meurer, "Efficient formulation of collision avoidance constraints in optimization based trajectory planning and control," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2021, pp. 228–233.
- [7] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete & Computational Geometry*, vol. 6, no. 3, pp. 423–434, 1991.
- [8] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, 2022.
- [9] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
- [10] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [11] A. S. Lewis and M. L. Overton, "Nonsmooth optimization via quasi-newton methods," *Mathematical Programming*, vol. 141, no. 1, pp. 135–163, 2013.