IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

# Motion Planning for Aerial Pick-and-Place With Geometric Feasibility Constraints

Huazi Cao, Jiahao Shen, Cunjia Liu<sup>®</sup>, *Member, IEEE*, Bo Zhu<sup>®</sup>, *Member, IEEE*, and Shiyu Zhao<sup>®</sup>, *Member, IEEE* 

Abstract—This paper studies the motion planning problem of the pick-and-place of an aerial manipulator that consists of a quadcopter flying base and a Delta arm. We propose a novel partially decoupled motion planning framework to solve this problem. Compared to the state-of-the-art approaches, the proposed one has two novel features. First, it does not suffer from increased computation in high-dimensional configuration spaces. That is because it calculates the trajectories of the quadcopter base and the end-effector separately in Cartesian space based on proposed geometric feasibility constraints. The geometric feasibility constraints can ensure the resulting trajectories satisfy the aerial manipulator's geometry. Second, collision avoidance for the Delta arm is achieved through an iterative approach based on a pinhole mapping method, so that the feasible trajectory can be found in an efficient manner. The proposed approach is verified by five experiments on a real aerial manipulation platform. The experimental results show the effectiveness of the proposed method for the aerial pick-and-place task.

Note to Practitioners—Aerial manipulators have attracted increasing research interest in recent years due to their potential applications in various domains. In this paper, we particularly focus on the motion planning problem of the pick-and-place of aerial manipulators. We propose a novel partially decoupled motion planning framework, which calculates the trajectories of the quadcopter base and the end-effector in Cartesian space, respectively. Geometric feasibility constraints are proposed to coordinate the trajectories to ensure successful execution. Five experiments on a real aerial manipulator platform demonstrate

Manuscript received 29 November 2023; revised 2 March 2024; accepted 6 March 2024. This article was recommended for publication by Associate Editor X. Li and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported in part by the Research Center for Industries of the Future, Westlake University, under Grant WU2022C027; in part by the Dean's Special Projects, School of Engineering, Westlake University, under Grant WU2023B013; in part by the National Natural Science Foundation of China under Grant 62373386; and in part by the Industry-University-Research Fund Project of the Ministry of Education of China under Grant 2021ZYA02017. (*Corresponding author: Shiyu Zhao.*)

Huazi Cao and Jiahao Shen are with the Intelligent Unmanned Systems Laboratory, School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: caohuazi@westlake.edu.cn; shenjiahao@westlake.edu.cn).

Cunjia Liu is with the Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU Loughborough, U.K. (e-mail: c.liu5@lboro.ac.uk).

Bo Zhu is with the Center for Advanced Control and Smart Operations (CACSO), Nanjing University, Suzhou 215163, China (e-mail: zhubo5@mail. sysu.edu.cn).

Shiyu Zhao is with the Research Center for Industries of the Future, the School of Engineering, and the Westlake Institute for Advanced Study, Westlake University, Hangzhou 310024, China (e-mail: zhaoshiyu@ westlake.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2024.3382296.

Digital Object Identifier 10.1109/TASE.2024.3382296

the effectiveness of the approach. In future research, we will address the motion planning problem of aerial manipulators in complex environments.

*Index Terms*— Aerial manipulator, Delta arm, aerial pick-andplace, motion planning, collision avoidance.

# I. INTRODUCTION

N AERIAL manipulator is a novel type of flying robot that consists of a multirotor and a robotic arm. Due to their ability to move quickly and operate precisely in high-altitude and complex workspaces, the aerial manipulator has potential applications in various domains, including transportation, inspection, and maintenance (see [1], [2], [3], [4] for recent surveys).

Aerial manipulation has been studied from various aspects such as platform design [5], [6], [7], [8], motion control [9], [10], [11], [12], motion planning [13], [14], [15] and visual servoing [16], [17], [18], [19] up to now. Our work focuses on the motion planning problem of aerial pick-and-place tasks, where the aerial manipulator is required to grasp and move objects in the environment (see Fig. 1). It is noted that safety and high efficiency are important to the aerial pick-and-place task since collisions between the aerial manipulator and the obstacles in the environment may occur. A computationally efficient motion planning scheme is necessary for the aerial pick-and-place task since it can diminish the demand for computing resources and contribute to mission success by enabling rapid re-planning and adaptation. This motivates our study to focus on an effective motion planning scheme that ensures collision-free trajectories.

Motion planning of multirotors is generally solved in Cartesian space [20]. The existing methods for the motion planning of multirotors can be classified into two categories. The first is the sampling-based motion planning method that focuses on global solutions of problems by exploration and exploitation [21]. The second is the optimization-based motion planning method that focuses on local solutions by using high-order information on the problems [22]. Different from the motion planning of multirotors, the motion planning of manipulators is generally solved in configuration space. The approaches proposed in the early stages aimed to obtain predefined paths and lacked the capability to adapt to changing environments [23]. More recently, researchers have proposed effective planning methods that consider the dynamics of the

1545-5955 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See https://www.ieee.org/publications/rights/index.html for more information.



Fig. 1. Aerial pick-and-place by an aerial manipulator. The experimental video is available at https://youtu.be/q7O9v7l2Oho.

manipulator, enabling the calculation of trajectories in complex workspaces [24], [25], [26].

Different from the motion planning of multirotors, the motion planning of an aerial manipulator is more challenging for two reasons. First, the aerial manipulator has more degrees of freedom (DoFs) since it combines the DoFs of the quadcopter and the robotic arm. Second, the aerial manipulator is required to manipulate the objects, which makes the constraints of the planning problem more complex. Different from the motion planning of a ground mobile manipulator, the motion planning of an aerial manipulator is more challenging since the aerial manipulator flies in a 3D environment rather than a 2D environment. In addition, the robotic arm and the multirotor base are dynamically coupled, which means their movements mutually affect each other. Existing approaches for motion planning for aerial manipulation can be classified into two categories based on the space in which planners calculate trajectories.

The first category is to plan the motion of the aerial manipulator in configuration space. In early works, the RRT\* method has been used to plan the path of the aerial manipulator in the configuration space without considering the dynamics [13], [14]. As a consequence, the resulting trajectory may not be executable for the aerial manipulator when its movement is fast. To address this issue, the dynamics of the aerial manipulator must be considered in motion planning. The existing methods that consider the dynamics in motion planning can be classified into three types.

The first type uses a kinematics controller as a local planner in the sampling-based global planner [15]. It guarantees the feasibility of the trajectory for the real system and also enables searching for a solution directly in the reduced and more relevant task space. However, collision avoidance is not inherently embedded in local planning, which may cause its result to be not collision-free. The second type uses the differential flatness principle to ensure the dynamical feasibility [27]. In particular, motion planning methods for a special long-reach aerial manipulator have been proposed in [28] and [29] based on this point of view. The platform in these works consists of a multirotor with a long bar extension that incorporates a lightweight dual arm in the tip. Since the dynamical feasibility constraints represented by the differential flatness are nonlinear, this type of method may be computationally expensive. The third type uses trajectory generation to ensure the dynamical feasibility [30], [31]. In the trajectory generation, the trajectories are obtained by solving optimization problems. The dynamical feasibility constraints are treated as constraints of the optimization problems. Planning in configuration space, however, suffers high computation costs when the dimension of the space is high and the workspace is large [32]. Unfortunately, aerial manipulators generally have high DoFs and large workspaces, which therefore motivates researchers to study other approaches to solve the motion planning problem.

The second category of approaches directly plans the trajectory of the end-effector in Cartesian space. The motion planning of the whole aerial manipulator is often solved practically by decoupling the flying base and the manipulator [33]. Firstly, the trajectory of the flying base approaching the manipulation position is calculated. Then, the motion of the end-effector is planned by assuming that the flying base stays in the same pose during manipulation. However, this method is conservative and inefficient in terms of energy and execution time [1]. To address this issue, the dynamic feasibility constraint must be considered in the trajectory planning of the end-effector. Therefore, a dynamically feasible task space planning method for underactuated aerial manipulators based on the differential flatness principle has been proposed in [34]. However, this method does not consider obstacle avoidance which is generally required in real scenarios.

The above analysis reveals the limitations of the existing motion planning approaches for aerial manipulators. Planning in configuration space incurs high computational costs due to the high DoF of aerial manipulators, while the existing methods of planning in Cartesian space do not consider obstacle avoidance, a crucial factor in real-world scenarios. To address these limitations, this paper proposes a novel framework that integrates the motion planning of both the flying base and the manipulator in a constrained workspace. The proposed algorithm is designed for an aerial manipulator consisting of a quadcopter and a Delta arm. Although there are existing optimization-based planning methods (e.g., [35], [36]) available for motion planning of quadrupeds and humanoid robots, they are all planned in configuration space. In contrast, our goal is to achieve an efficient motion planning method for aerial manipulators in Cartesian space. The novelty of our approach is outlined below:

1) We propose a novel partially decoupled motion planning framework for the aerial pick-and-place task. This framework calculates the dynamically feasible and collision-free trajectories of the flying base and the manipulator in Cartesian space, respectively. The resulting trajectories are coordinated for successful execution. By solving the motion planning problem in Cartesian space, the high DoF of the aerial manipulator can be handled more efficiently than planning in the configuration space with a much lower computational load. Compared with the existing methods that plan trajectories in configuration space (e.g., [30]), this method does not suffer from the problem of increased computation in high-dimensional configuration spaces. Compared with the existing method that

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 2. Coordinates, revised workspace, and shape polyhedron of the aerial manipulator.

plans trajectories in Cartesian space [34], the proposed method ensures that the trajectories are collision-free.

2) We propose novel geometric feasibility constraints to ensure the trajectories of the quadcopter and the end-effector can be successfully executed. Our proposed constraints are derived from the intersection of boundaries of the nonlinear geometric relationship. They are linearly represented by the positions of the quadcopter and the end-effector, whereas the original geometry constraints are nonlinearly represented by the configuration of the aerial manipulator. By using the constraints, our method ensures that the resulting trajectories satisfy the geometry of the aerial manipulator. The proposed constraints can also be used in developing other efficient motion planning methods in Cartesian space. The efficiency can be significantly enhanced by converting nonlinear constraints into linear ones.

3) Collision avoidance for the Delta arm is achieved through an efficient iterative approach based on a proposed pinhole mapping method. At each iteration, a quadratic programming (QP) problem is solved to determine the collision-free trajectory for the end-effector. A collision avoidance term, designed based on the pinhole mapping method and collision check results, is formulated into the QP problem, so that the aerial manipulator is driven away from the obstacles in the local environment. Compared to collision avoidance in configuration space [27], [30], the proposed iterative approach is faster as it does not require intensive sampling in configuration space. Compared to the traditional artificial potential method [37], the proposed method does not rely on a predefined artificial potential field. Moreover, compared to the control-barrierfunction method [38], the proposed method can directly handle changes in the shape of the aerial manipulator during motion in Cartesian space.

The proposed algorithms are verified by five experiments on an aerial manipulator platform in the real aerial pick-and-place task. Unlike the traditional Delta arm [6], the Delta arm used in this paper drives the joint angles by three four-bar linkages to magnify the control forces [39]. Experiments including collision avoidance, aerial retrieval, and aerial transport are conducted to validate the novelties. The rest of this paper is structured as follows. The problem statement and preliminaries are given in Section II. Kinematics and geometric feasibility constraints of the aerial manipulator are presented in Section III. The motion planning of the quadcopter base is proposed in Section IV. Section V gives the motion planning of the Delta arm. Then, the experimental verification is given in Section VI. Conclusions are drawn in Section VII.

# **II. PROBLEM STATEMENT AND PRELIMINARIES**

# A. Problem Statement

The platform is an aerial manipulator that consists of a quadcopter and a Delta arm (see Fig. 2(a)). The base of the Delta arm is attached underneath the quadcopter. The end-effector used in this paper is a gripper and it is mounted on the end of the Delta arm, whose position can be controlled by the three actuators attached to the base of the Delta arm. The aerial manipulator has nine DoFs since the quadcopter has six DoFs and the Delta arm has three translational DoFs. Because the quadcopter only has four control inputs (i.e., the velocities of the four blades), the attitude and position of the quadcopter are coupled. In addition, each kinematic chain in the Delta arm consists of a series of connected links and joints, forming a closed loop from the base to the end. It means the three kinematic chains of the Delta arm are not independent.

The aerial manipulator has three reference frames: the inertial frame  $\Sigma_I$ , the quadcopter body-fixed frame  $\Sigma_B$ , and the Delta arm frame  $\Sigma_D$  (see Fig. 2(a)).  $\Sigma_I$  is an inertial frame where the *z*-axis is in the direction of the gravity vector.  $\Sigma_B$  is rigidly attached to the quadcopter base. Its origin coincides with the center of gravity of the quadcopter.  $\Sigma_D$  is rigidly attached to the Delta arm base at its geometric center  $p_C$ .

Let  $p_B \in \mathbb{R}^3$  and  $R_B \in SO(3)$  denote the position of the quadcopter in  $\Sigma_I$  and the rotation matrix from  $\Sigma_B$  to  $\Sigma_I$ , respectively. Let  $p_E \in \mathbb{R}^3$  denote the position of the end-effector in  $\Sigma_I$ . Then, the geometric relationship between  $p_E$  and  $p_B$  can be represented as

$$\boldsymbol{p}_E - \boldsymbol{p}_B = \boldsymbol{R}_B \boldsymbol{p}_E^B, \qquad (1)$$

4

where  $p_E^B \in \mathbb{R}^3$  is a function of the Delta arm's actuated joint angles  $q_1, q_2, q_3$ .

For a pick-and-place task, denote  $p_O \in \mathbb{R}^3$  and  $\psi_O \in \mathbb{R}$  as the position and the orientation of the target object in  $\Sigma_I$ , respectively, whereas  $\psi_E \in \mathbb{R}$  denotes the orientation angle of the end-effector. Let  $t_G$  denote the time that  $p_E$  arrives at  $p_O$  and  $t_{grip}$  the closing time of the gripper.

The goal of the motion planning for the aerial pick-andplace is to calculate the collision-free trajectories for the quadcopter and the Delta arm to move from a starting position to a feasible grasping configuration and from that grasping configuration to the end position. Given the geometric relationship, dynamical feasibility constraints, obstacles in the environment, the start position  $p_{B,\text{start}}$ , and the end position  $p_{B,\text{end}}$ , the resulting trajectories must be collision-free and satisfy that  $p_E(t) = p_O$  and  $\psi_E(t) = \psi_O$  when  $t \in [t_G, t_G+t_{\text{grip}}]$ .

# B. Overview of the Proposed Motion Planning Method

The proposed motion planning method is partially decoupled, which calculates the trajectories of the quadcopter base  $p_B(t)$  and the end-effector  $p_E(t)$  in Cartesian space, respectively. The geometric feasibility constraints are proposed to coordinate the trajectories to ensure successful execution (see Section III-B for details). The overall architecture of the motion planning and control system is shown in Fig. 3. The system is decomposed into three components.

1) The first component is the motion planning of the quadcopter base. Its inputs are the positions of the object and the obstacles. Its output is the trajectory of the quadcopter base  $p_{B,ref}(t)$ . The motion planning of the quadcopter base can be further decomposed into four steps. The first step is feasible grasping position calculation. Its role is to find a suitable position for the quadcopter base to allow the aerial manipulator to grasp the object. The details of this step can be seen in Section IV-A. The second step is path planning. Its role is to find a path for the quadcopter base to move from a given starting position to a feasible grasping position and from that grasping position to a given end position. In this paper, we use the A\* method to calculate the path [41, Section 12.1.1]. The third step is flight corridor generation. Its role is to generate a safe flight corridor for the quadcopter base which constrains the motion of the quadcopter base to avoid collisions. The details of this step can be seen in Section IV-B. The fourth step is trajectory generation. Its role is to calculate the trajectory of the quadcopter base based on the piecewise Bézier curve. We use the method proposed in [42] to ensure the resulting trajectory satisfies the safety, the dynamical feasibility, and the waypoints constraints. Compared with the existing methods of aerial manipulators (e.g., [30]), the proposed method calculates the trajectory of the quadcopter base in Cartesian space. Compared with the existing methods of the standard quadcopter (e.g., [42]), the proposed method guarantees the aerial manipulator arrives at the feasible grasping configuration without collisions.

2) The second component is the motion planning of the Delta arm. Its inputs are the position of the object  $p_O$  and  $p_{B,ref}(t)$ . Its output is the trajectory of the end-effector  $p_{E,ref}(t)$ . This motion planning method can be further

decomposed into three steps. The first step is the initial condition calculation. Its role is to calculate the position, velocity, and acceleration of the end-effector at the beginning of the manipulation stage. The second step is the optimal trajectory planning of the end-effector based on the Bézier curve. Its role is to calculate the trajectory of the end-effector from the initial position to the object with several constraints. The trajectory planning of the end-effector is represented as a QP problem form. In particular, we propose geometric feasibility constraints of the aerial manipulator and encode this constraint into the QP problem to ensure the trajectories satisfy the geometry of the aerial manipulator. The third step is collision avoidance. It is important and its role is to ensure the trajectory of the end-effector is collision-free. In this step, the collisions between the aerial manipulator and the obstacles in a local map are detected based on the GJK method. The second and third steps are run iteratively. If there is a collision, then the objective function of the QP problem in the second step is updated by a pinhole mapping method. Repeat the second and third steps until no collision occurs. All the corresponding sections introducing these steps are listed in Fig. 3. Compared with the existing methods [30], [43], the proposed method requests lower computational power since the collision avoidance of the Delta arm is achieved by an iterative approach in Cartesian space.

3) The third component is the controller of the aerial manipulator. Its inputs are the trajectories of the quadcopter base and the end-effector. Its outputs are total force  $f \in \mathbb{R}$ of the rotors, torque vector  $\boldsymbol{\tau} \in \mathbb{R}^3$  of the rotors, torque  $\tau_G \in \mathbb{R}$  of the gripper, and the torque vector that each actuator should generate  $\tau_M \in \mathbb{R}^3$ . The controller consists of three subcomponents. The first step is an extended state observer (ESO) -based flight controller. It was proposed in our previous work [44] and uses ESOs to estimate dynamic coupling between the aerial manipulator and the Delta arm. Its role is to generate the force f and torque vector  $\boldsymbol{\tau}$  for the quadcopter base so that the trajectory of the quadcopter can be tracked. The second step is the end-effector controller. Its role is to control the gripper to grasp or release objects. The third step is the Delta arm controller. Its role is to generate the torque vector  $\boldsymbol{\tau}_M$  for the Delta arm so that the trajectory of the end-effector can be tracked. The details of the Delta arm controller can be seen in our previous work [44].

The steps can be classified into real-time processes and nonreal-time processes. In Fig. 3, all steps are executed on board. Steps in the small grey rectangle are calculated before the task begins. Due to the efficiency of the proposed motion planning method, it also has the potential to generate new trajectories for the aerial manipulator in response to unexpected situations. The steps in the white rectangle run onboard the aerial manipulator during flights. Considering that motion planning doesn't require real-time calculations, it doesn't impose an excessive burden on the computing resource.

## C. preliminaries to Bézier Curves

A *n*-th degree Bézier curve is defined by a set of control points and Bernstein polynomial bases. Let  $c_i \in \mathbb{R}^3$ ,

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 3. Structure of the proposed motion planning method for aerial pick-and-place.

 $b_{i,n}(\tau) \in \mathbb{R}$  denote the *i*-th control point and Bernstein polynomial basis, respectively. Then, the *n*-th degree 3D Bézier curve is written as  $B(\tau) = \sum_{i=0}^{i=n} c_i^T b_{i,n}(\tau)$ , where

$$b_{i,n}(\tau) = \binom{n}{i} \tau^i (1-\tau)^{n-i}, \qquad (2)$$

where  $\tau \in [0, 1]$ ,  $\binom{n}{i}$  is the binomial coefficient. According to [45, Section 2.4], the derivative of the Bézier curve can be obtained by Lemma 1. In addition, the Bézie curve B(t) is entirely confined within the convex hull defined by all these control points, which is referred to as the convex hull property (see Lemma 2).

Lemma 1 (Derivative [45]): Let  $\mathbf{B}^{(k)}(\tau) = \sum_{i=0}^{n-j} \mathbf{c}_i^{(k)} b_{i,n-j}(\tau)$  denote the k-th derivative of  $\mathbf{B}(\tau)$ , then the control points of  $\mathbf{B}^{(k)}(\tau)$  can be calculated iteratively by  $\mathbf{c}_i^{(k)} = (n-k+1)(\mathbf{c}_{i+1}^{(k-1)} - \mathbf{c}_i^{(k-1)})$ , where  $i = 0, 1, \dots, n-j$ .

Lemma 2 (Convex hull property [45]): Let  $\mathbb{H} = \{a_0 \mathbf{c}_0 + a_1 \mathbf{c}_1 + \cdots + a_n \mathbf{c}_n | a_0 + a_1 + \cdots + a_n = 1, a_i \ge 0\}$  denote the convex hull defined by all the control points, then  $\mathbf{B}(\tau) \in \mathbb{H}$  for all  $\tau \in [0, 1]$ .

# III. KINEMATICS AND GEOMETRIC FEASIBILITY CONSTRAINTS

This section proposes the kinematics and geometric feasibility constraints of the aerial manipulator.

# A. Kinematics of the Aerial Manipulator

According to (1), the time derivative of  $p_E$  is

$$\dot{\boldsymbol{p}}_{E} = \dot{\boldsymbol{p}} + \dot{\boldsymbol{R}}_{B}\boldsymbol{p}_{E}^{B} + \boldsymbol{R}\dot{\boldsymbol{p}}_{E}^{B}$$
$$= \dot{\boldsymbol{p}} + \boldsymbol{R}_{B}\boldsymbol{R}_{D}^{B}\dot{\boldsymbol{p}}_{E}^{D} - [\boldsymbol{R}_{B}\boldsymbol{p}_{E}^{B}]_{\times}\boldsymbol{\omega}, \qquad (3)$$

where  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the angular velocity vector of the quadcopter expressed in  $\Sigma_B$ , and  $[\cdot]_{\times}$  denotes the skew-symmetric matrix.

Let  $p_C^B \in \mathbb{R}^3$  denote the position of the center of the base in  $\Sigma_B$ . Let  $p_E^B \in \mathbb{R}^3$  and  $p_E^D \in \mathbb{R}^3$  denote the positions of the end-effector in  $\Sigma_B$  and  $\Sigma_D$ , respectively. The relationship between  $p_E^B$  and  $p_E^D$  is

$$\boldsymbol{p}_E^B = \boldsymbol{R}_D^B \boldsymbol{p}_E^D + \boldsymbol{p}_C^B, \qquad (4)$$

where  $\mathbf{R}_D^B \in SO(3)$  is the rotation matrix from  $\Sigma_D$  to  $\Sigma_B$ .

The lengths for the upper and lower arms are represented by  $l_U$  and  $l_L$  as illustrated in Fig. 2(a). Circumradius of the top base and the bottom end-effector base are, respectively, defined as  $r_F$  and  $r_M$ . The length of the gripper is denoted as  $l_g$ . The relationship between the end-effector position  $\boldsymbol{p}_E^D$  and the joint vector  $\boldsymbol{q} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$  is

$$\|\boldsymbol{p}_{E}^{D} + \boldsymbol{l}_{G} - \boldsymbol{h}_{i}\|^{2} = l_{L}^{2}, \quad i = 1, 2, 3,$$
 (5)

where  $l_G = [0, 0, l_g]^T$ , and

$$\boldsymbol{h}_{i} = \begin{bmatrix} -(r_{F} - r_{M} + l_{U} \cos q_{i}) \cos[(i - 1)\pi/3] \\ (r_{F} - r_{M} + l_{U} \cos q_{i}) \sin[(i - 1)\pi/3] \\ l_{U} \sin q_{i} \end{bmatrix}.$$
 (6)

On the one hand, given a joint vector  $\boldsymbol{q}$ , the position  $\boldsymbol{p}_E^D$  can be solved from (5) based on the forward kinematics. On the other hand, given a position  $\boldsymbol{p}_E^D$ , the joint vector  $\boldsymbol{q}$  can be solved from (5) by the inverse kinematics. Details can be found in [46] and [47].

As can be seen from Fig. 2(a), the joint angles of the Delta arm are driven by planar four-bar linkages. The relationship between the joint angles and the crank position angles can be calculated by the kinematics of the planar four-bar linkage [48, Section 3.6].

#### B. Geometric Feasibility Constraints

Combining (1) and (4), the geometric relationship between the end-effector and the quadcopter is

$$\boldsymbol{p}_E - \boldsymbol{p}_B = \boldsymbol{R}_B (\boldsymbol{R}_D^B \boldsymbol{p}_E^D + \boldsymbol{p}_C^B), \quad \boldsymbol{p}_E^D \in \mathbb{W}, \tag{7}$$

where  $\mathbb{W}$  is the workspace of the Delta and can be calculated by the forward kinematics of the Delta arm. The workspace is approximated as a convex polyhedron [49]. Therefore, the expression of the workspace is

$$\mathbb{W} = \{ \boldsymbol{p} | \boldsymbol{A}_D \, \boldsymbol{p} \le \boldsymbol{b}_D \}. \tag{8}$$

Authorized licensed use limited to: Westlake University. Downloaded on May 01,2024 at 06:54:17 UTC from IEEE Xplore. Restrictions apply.

6

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

Since  $\mathbf{R}_D^B$  and  $\mathbf{p}_C^B$  are constant matrix and vector, according to (7), the range of  $\mathbf{p}_E - \mathbf{p}_B$  is determined by  $\mathbb{W}$  and  $\mathbf{R}_B$ . We define  $\mathbf{R}_B = \mathbf{R}_{\psi} \mathbf{R}_{\theta,\phi}$ , where  $\mathbf{R}_{\psi}$  is the rotation matrix determined by the yaw angle  $\psi$ ,  $\mathbf{R}_{\theta,\phi}$  are the rotation matrix determined by the pitch angle  $\theta$  and roll angle  $\phi$ . The yaw angle of the quadcopter is constant, i.e.,  $\psi = \psi_O$ , when the aerial manipulator is grasping or placing an object. This is because the quadcopter's yaw angle reaches the desired angle before the manipulation begins. Then, (7) is rewritten as

$$\boldsymbol{R}_{\psi_O}^T(\boldsymbol{p}_E - \boldsymbol{p}_B) = \boldsymbol{R}_{\theta,\phi}(\boldsymbol{R}_D^B \boldsymbol{p}_E^D + \boldsymbol{p}_C^B), \quad \boldsymbol{p}_E^D \in \mathbb{W}.$$
(9)

To make the above equation more concise, we define

$$\mathbb{W}_{\theta,\phi} = \{ \boldsymbol{R}_{\theta,\phi} (\boldsymbol{R}_D^B \boldsymbol{p}_E^D + \boldsymbol{p}_C^B) | \boldsymbol{p}_E^D \in \mathbb{W} \}.$$
(10)

Therefore, (9) is rewritten as  $\boldsymbol{R}_{\psi_0}^T(\boldsymbol{p}_E - \boldsymbol{p}_B) \in \mathbb{W}_{\theta,\phi}$ .

To linearize the geometric relationship (9), we define  $\mathbb{W}_R = \{p | w_{\min} \leq p \leq w_{\max}\}$  as the revised workspace, and it satisfy that  $\mathbb{W}_R \subset \mathbb{W}_{\theta,\phi}$ . According to (7),  $\phi$  and  $\theta$  also are the rotation angles of  $\mathbb{W}_{\theta,\phi}$  in pitch and roll directions. Since the pitch and roll angles of the quadcopter are small when the aerial manipulator is manipulating, the bounds of  $\theta$  and  $\phi$  can be determined with several experiments. Let  $\theta_{\min}, \theta_{\max}$  denote the minimum and maximum of  $\theta$ . We calculate  $\mathbb{W}_R$  by two steps.

The first step is calculating boundaries of  $W_{\theta,\phi}$ . Combining (8) and (10), the expression of  $W_{\theta,\phi}$  can be rewritten as

$$\mathbb{W}_{\theta,\phi} = \{ \boldsymbol{p} | \boldsymbol{A}_D (\boldsymbol{R}_{\theta,\phi} \boldsymbol{R}_D^B)^T \boldsymbol{p} \le \boldsymbol{b}_D + \boldsymbol{A}_D \boldsymbol{R}_D^{BT} \boldsymbol{p}_C^B \}.$$
(11)

According to (11), we obtain the boundaries  $\mathbb{W}_{\theta=\theta_{\min},\phi=0}$ ,  $\mathbb{W}_{\theta=0,\phi=\phi_{\min}}, \mathbb{W}_{\theta=0,\phi=\phi_{\max}}$ .

The second step is calculating the intersection of these sets  $\mathbb{W}_{I}$ . According to the definition of the intersection, we have

$$\mathbb{W}_{I} = \{\boldsymbol{p}|\boldsymbol{A}_{D}(\boldsymbol{R}_{\theta_{\min},0}\boldsymbol{R}_{D}^{B})^{T}\boldsymbol{p} \leq \boldsymbol{b}_{D} + \boldsymbol{A}_{D}\boldsymbol{R}_{D}^{BT}\boldsymbol{p}_{C}^{B}, \\ \boldsymbol{A}_{D}(\boldsymbol{R}_{\theta_{\max},0}\boldsymbol{R}_{D}^{B})^{T}\boldsymbol{p} \leq \boldsymbol{b}_{D} + \boldsymbol{A}_{D}\boldsymbol{R}_{D}^{BT}\boldsymbol{p}_{C}^{B}, \\ \boldsymbol{A}_{D}(\boldsymbol{R}_{0,\phi_{\min}}\boldsymbol{R}_{D}^{B})^{T}\boldsymbol{p} \leq \boldsymbol{b}_{D} + \boldsymbol{A}_{D}\boldsymbol{R}_{D}^{BT}\boldsymbol{p}_{C}^{B}, \\ \boldsymbol{A}_{D}(\boldsymbol{R}_{0,\phi_{\max}}\boldsymbol{R}_{D}^{B})^{T}\boldsymbol{p} \leq \boldsymbol{b}_{D} + \boldsymbol{A}_{D}\boldsymbol{R}_{D}^{BT}\boldsymbol{p}_{C}^{B},$$
(12)

Since the expression of the intersection (12) is complicated, it may be inconvenient when applied to real systems. We let the largest cuboid that can be inscribed within the intersection as  $W_R$ . The cuboid can be calculated by the method proposed in [50]. Then,  $\boldsymbol{w}_{\min} = [w_{x,\min}, w_{y,\min}, w_{z,\min}]^T$  and  $\boldsymbol{w}_{\max} = [w_{x,\max}, w_{y,\max}, w_{z,\max}]^T$  are subsequently determined by the size of the cuboid. Fig. 2(b) gives an illustration for calculating the revised workspace in the pitch direction. Then, the geometric feasibility constraints are

$$\boldsymbol{w}_{\min} \leq \boldsymbol{R}_{\psi_0}^T (\boldsymbol{p}_E - \boldsymbol{p}_B) \leq \boldsymbol{w}_{\max}, \qquad (13)$$

where

$$\boldsymbol{R}_{\psi_{O}} = \begin{bmatrix} \cos\psi_{O} & -\sin\psi_{O} & 0\\ \sin\psi_{O} & \cos\psi_{O} & 0\\ 0 & 0 & 1 \end{bmatrix}.$$
 (14)

| Algorithm 1 Motion Planning for the Quadcopter  |
|---|
| <b>Input:</b> $p_O$ , $p_{B,\text{star}}$ , $p_{B,\text{end}}$ , $\mathbb{M}$         |
| <b>Output:</b> Trajectory $p_{B,ref}(t)$  |
| 1: function PLAN_QUAD( $p_O, p_{B, \text{star}}, p_{B, \text{end}}, \mathbb{M}$ )     |
| 2: $p_{B,f} \leftarrow \text{calculating (16)}$                                       |
| 3: Flight_path $\leftarrow$ A_STAR( $p_{B, star}, p_{B, end}, p_{B, f}, \mathbb{M}$ ) |
| 4: Flight_corridor $\leftarrow$ CONVEX_DECOM(Flight_path)                             |
| 5: Designed_polyh $\leftarrow$ calculating (18)                                       |
| 6: Flight_corridor = Flight_corridor $\bigcup$ Designed_polyh                         |
| 7: $p_{B,ref}(t) \leftarrow TRAJECTORY\_GEN(Flight\_corridor)$                        |
| 8: return $p_{B,ref}(t)$  |
| 9: end function   |

# IV. MOTION PLANNING FOR THE QUADCOPTER BASE

This section presents a method to generate the trajectory of the quadcopter for the aerial pick-and-place task. This method consists of four steps: feasible grasping position calculation, path planning, flight corridor generation, and Bézier curvebased trajectory generation (see Algorithm 1). The purposes and relationships of these steps are given in Section II-B. In the algorithm, the path planning can be achieved by the existing method. In our work, we use the A\* method to obtain the path in the 3D grid map which is used to represent the environment of the task. The Bézier curve-based trajectory generation is achieved by an existing method proposed in [42]. It bounds positions and higher-order dynamics of the trajectory entirely within safe regions by using the Bernstein polynomial basis and formulating the trajectory generation problem as typical convex programs.

Compared to the existing methods for standard quadcopters [42], the proposed method for the quadcopter base of the aerial manipulator has two novelties. First, the feasible grasping position is calculated to ensure the aerial manipulator can manipulate the object. Second, the volume of the aerial manipulator changes with the movement of the Delta arm. To address this issue, the aerial pick-and-place task is divided into two stages: moving and manipulation stages. The flight corridors in the two stages are obtained, respectively. The details are shown as follows.

#### A. Feasible Grasping Position

To grasp the object, the position of the end-effector must arrive at  $p_O$  with an orientation angle of  $\psi_O$ . The feasible grasping position of the quadcopter is constrained by the geometric shape of the aerial manipulator. Let  $p_{B,f} \in \mathbb{R}^3$  denote the feasible grasping position. Let  $R_{B,f} \in SO(3)$  denote the desired rotation matrix of the quadcopter base at the feasible grasping position. According to (1), the feasible grasping position of the quadcopter is

$$\boldsymbol{p}_{B,f} = \boldsymbol{p}_O - \boldsymbol{R}_{B,f} \boldsymbol{p}_E^B. \tag{15}$$

According to (15), one can conclude that  $\mathbf{R}_{B,f}$  and  $\mathbf{p}_{E}^{B}$  need to be determine before calculating  $\mathbf{p}_{B,f}$ .

In the manipulation stage, the yaw angle of the quadcopter is set as  $\psi_0$  to satisfy the grasp angle constraint of the endeffector. We assume that the roll and pitch angles of the

Authorized licensed use limited to: Westlake University. Downloaded on May 01,2024 at 06:54:17 UTC from IEEE Xplore. Restrictions apply

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 4. An illustration for the motion planning of the aerial pick-and-place.

quadcopter are small when the quadcopter base is around  $p_{B,f}$ . This assumption is reasonable since the motion of the quadcopter is conservative. According to the assumption, we have  $R_{B,f} = R_{\psi_0}$ . To ensure the manipulability of the Delta arm, we let the end-effector stay at the center of  $\mathbb{W}_R$  when the aerial manipulator picks up the object. Then, (15) is rewritten as

$$\boldsymbol{p}_{B,f} = \boldsymbol{p}_O - 0.5\boldsymbol{R}_{\psi_O}(\boldsymbol{w}_{\min} + \boldsymbol{w}_{\max}). \tag{16}$$

For an aerial pick-and-place task, we already have the start position  $p_{B,\text{start}}$ , feasible grasping position  $p_{B,f}$ , and the end position  $p_{B,\text{end}}$ . Then, the path of the quadcopter can be obtained by the A\* method.

# B. Flight Corridor Generation

The flight corridor is a collection of convex overlapping polyhedra that models free space and provides a connected corridor containing the resulting path. A convex decomposition method proposed in [51] is adopted to generate the flight corridor by inflating the resulting path. However, this method was originally designed for a traditional quadcopter with a fixed volume, while the volume of the aerial manipulator changes with the movement of the Delta arm. Therefore, the method cannot be directly used for the aerial manipulator. To address this issue, we calculate the flight corridors in the moving and the manipulation stages, respectively. In the moving stage, the position of the end-effector is set to stay at the top point  $p_{top}^B \in \mathbb{R}^3$  of the Delta arm's workspace  $\mathbb{W}_R$ , which folds the Delta arm into its most compact state to reduce the volume of the aerial manipulator. From the definition of  $\mathbb{W}_R$ , we have

$$\boldsymbol{p}_{top}^{B} = \begin{bmatrix} 0.5(w_{x,\min} + w_{x,\max}) \\ 0.5(w_{y,\min} + w_{y,\max}) \\ w_{z,\min} \end{bmatrix}.$$
 (17)

The shape of the aerial manipulator now can be approximated as a sphere with a radius  $r_s$ . The radius  $r_s$  is determined by the wheelbase and the size of the rotors. Since the wheelbase of the quadcopter is relatively large (650 mm in this paper), the Delta arm is completely contained within the sphere in the set state. Then, we can use the convex decomposition method to generate the flight corridor in the moving stage.

In the manipulation stage, we use a designed polyhedron as the flight corridor to ensure the object is reachable for the aerial manipulator (see Fig. 4). It is necessary since the quadcopter and Delta arm can collaborate to enhance efficiency. The designed polyhedron is designed based on the geometric feasibility constraints (13) and it is represented as

$$\boldsymbol{w}_{\min} \leq \boldsymbol{R}_{\psi_O}(\boldsymbol{p}_B - \boldsymbol{p}_{B,f}) \leq \boldsymbol{w}_{\max}.$$
 (18)

According to the geometric relationship (7), the end-effector can touch the object when the position of the quadcopter is in the designed polyhedron. It means that if the quadcopter is inside the designed polyhedron, the aerial manipulator can perform the manipulation. The duration time in this polyhedron is determined by the mechanical behavior of the gripper. We set the duration time as the closing time of the gripper  $t_{grip}$ .

#### V. MOTION PLANNING FOR THE DELTA ARM

In this section, we calculate the collision-free trajectory of the Delta arm in the Cartesian coordinate. The proposed method for the Delta arm utilizes the resulting trajectory of the quadcopter (see Algorithm 2). In the moving stage, the Delta arm stays at an initial state and its end-effector stays at a fixed position  $p_{top}^{B}$  relative to the quadcopter base. The position  $p_{top}^{B}$  can be calculated by (17). Therefore, the Delta arm does not require additional motion planning calculations in the moving stage.

Let  $t_B \in \mathbb{R}$  denote the time at which the quadcopter base enters the designed polyhedron and it is the beginning time of the manipulation stage. The time  $t_B$  can be determined by  $t_B = t_G - k_B t_A$ , where  $t_A \in \mathbb{R}$  is the approach time of the end-effector. The time  $t_A$  can be calculated by  $t_A =$  $\max(v_{E,\max}/a_{E,\max}, l_w/v_{E,\max})$ , where  $l_w = ||\boldsymbol{w}_{\max} - \boldsymbol{w}_{\min}|| \in$  $\mathbb{R}$  is the size of the revised workspace,  $v_{E,\max} \in \mathbb{R}$  and  $a_{E,\max} \in \mathbb{R}$  are maximum velocity and acceleration of the end-effector, respectively. The reason why we set  $t_B$  as this is Algorithm 2 Motion Planning for the Delta Arm

**Input:**  $p_O$ ,  $\mathbb{M}$ ,  $p_{B,ref}(t)$ **Output:** Trajectory  $p_{E,ref}(t)$ 1: **function** PLAN\_DEALTA\_ARM( $\boldsymbol{p}_{O}$ ,  $\mathbb{M}$ ,  $\boldsymbol{p}_{B,\text{ref}}(t)$ ) 2:  $\boldsymbol{p}_{E,t_B}, \, \boldsymbol{\dot{p}}_{E,t_B}, \, \boldsymbol{\ddot{p}}_{E,t_B} \leftarrow \text{INI\_CONDITION}(\boldsymbol{p}_{B,\text{ref}}(t))$ Set  $\mathbb{O}_M = \emptyset$  and  $\Delta \lambda_i = 0$ 3: while is\_collision do 4:  $c_{E,i} \leftarrow \text{calculating} (24)$ 5:  $\boldsymbol{p}_{E,\mathrm{ref}}(\tau_M) \leftarrow \sum_{i=0}^{n_E} \boldsymbol{c}_{E,i} b_{i,n_E}(\tau_M)$ 6:  $\mathbb{O}_M, \lambda_i \leftarrow \text{COLLISION}_\text{AVOID}(\boldsymbol{p}_{E, \text{ref}}(t), \mathbb{M})$ 7: end while 8: return  $p_{e,ref}(t)$ 9: 10: end function

to enable the end-effector to achieve the desired manipulation speed and position within the specified time. The switching from the moving stage to the manipulation stage is achieved by the manipulation beginning time  $t_B$ . The switching from the manipulation stage to the moving stage is achieved by the manipulation ending time  $t_G + t_{grip}$ , where  $t_{grip} \in \mathbb{R}$  is the closing time of the gripper.

The procedure of the manipulation stage is given here. From  $t_B$  to  $t_G$ , the end-effector moves to the object. Then, the aerial manipulator keeps the position of the end-effector for the duration time  $t_{grip}$  to pick up or place the object. After picking up or placing the object, the Delta arm returns to its initial state. According to the procedure, one can find that the trajectory of the end-effector from  $t_B$  to  $t_G$  needs to be calculated. The details to calculate the trajectory from  $t_B$  to  $t_G$  are shown as follows.

#### A. Initial Condition

The initial condition for the end-effector consists of the initial position  $p_{E,t_B}$ , the initial velocity  $\dot{p}_{E,t_B}$ , and the initial acceleration  $\ddot{p}_{E,t_B}$ . They are calculated as follows.

1) Initial Position: According to (1), the initial position  $p_{E,t_R}$  is calculated by

$$\boldsymbol{p}_{E,t_B} = \boldsymbol{p}_{B,t_B} + \boldsymbol{R}_{B,t_B} \boldsymbol{p}_{\text{top}}^B, \qquad (19)$$

where  $\boldsymbol{p}_{B,t_B}$ ,  $\boldsymbol{R}_{B,t_B} = [\boldsymbol{r}_{1,t_B}, \boldsymbol{r}_{2,t_B}, \boldsymbol{r}_{3,t_B}]$  denote  $\boldsymbol{p}_B$ ,  $\boldsymbol{R}_B$  at the time  $t_B$ , respectively. According to (19), we calculate  $\boldsymbol{p}_{B,t_B}$  and  $\boldsymbol{R}_{B,t_B}$  to obtain  $\boldsymbol{p}_{E,t_B}$ .  $\boldsymbol{p}_{B,t_B}$  can be directly obtained by the trajectory of the quadcopter.

The matrix  $\mathbf{R}_{B,t_B}$  is calculated based on the differential flatness of the quadcopter. At the time  $t_B$ , the yaw angle of the quadcopter base is  $\psi_O$  to ensure the orientation angle of the end-effector equals  $\psi_O$ . The unit orientation vector in the ground plane is  $\mathbf{r}_g = [\cos \psi_O, \sin \psi_O, 0]^T$ . According to [52], we have

$$\boldsymbol{r}_{3,t_B} = \frac{\boldsymbol{\ddot{p}}_{t_k} + g\boldsymbol{e}_3}{\|\boldsymbol{\ddot{p}}_{t_B} + g\boldsymbol{e}_3\|},\tag{20}$$

and vectors  $\mathbf{r}_{1,t_{B}}$  and  $\mathbf{r}_{2,t_{B}}$  can be determined by

$$\mathbf{r}_{2,t_B} = \frac{\mathbf{r}_{3,t_B} \times \mathbf{r}_g}{\|\mathbf{r}_{3,t_B} \times \mathbf{r}_g\|}, \mathbf{r}_{1,t_B} = \mathbf{r}_{2,t_B} \times \mathbf{r}_{3,t_B}.$$
 (21)

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING

2) Initial Velocity and Acceleration: Let  $\delta_I$  denote a small time step. We can calculate  $p_{E,t_B-\delta_I}$  and  $p_{E,t_B+\delta_I}$  according to the above method. Then, the derivatives are approximated by the differences. The initial velocity and acceleration are

$$\dot{\boldsymbol{p}}_{E,t_B} = (\boldsymbol{p}_{E,t_B} - \boldsymbol{p}_{E,t_B - \delta_I})/\delta_I, \ddot{\boldsymbol{p}}_{E,t_B} = (\boldsymbol{p}_{E,t_B + \delta_I} - 2\boldsymbol{p}_{E,t_B} + \boldsymbol{p}_{E,t_B - \delta_I})/\delta_I^2.$$
(22)

#### B. Optimal Trajectory Planning

The trajectory of the end-effector is calculated by an iterative approach. The trajectory planning of the end-effector is formulated as a QP problem. At each iteration, the objective function of the QP problem is updated and the QP problem is solved to calculate the collision-free trajectory of the endeffector. Let  $\mathbf{p}_{E,\text{ref}}(t), t \in [t_B, t_G]$  denote the trajectory. A  $n_E$ -th order Bézier curve is adopted to represent the trajectory and it is

$$\boldsymbol{p}_{E,\text{ref}}(\tau_M) = \sum_{i=0}^{n_E} \boldsymbol{c}_{E,i} b_{i,n_E}(\tau_M), \qquad (23)$$

where  $\mathbf{c}_{E,i} = [c_{E,x,i}, c_{E,y,i}, c_{E,z,i}]^T \in \mathbb{R}^3$  and  $b_{i,n_E}(\tau_M)$  are the *i*-th control point and Bernstein polynomial basis of the Bézier curve, respectively, and  $\tau_M = (t - t_B)/(t_G - t_B)$ . We denote the parameter vector of the trajectory as  $\underline{c}_E = [c_{E,x,0}, \ldots, c_{E,x,n_E}, c_{E,y,0}, \ldots, c_{E,y,n_E}, c_{E,z,0}, \ldots, c_{E,z,n_E}]^T$ . The QP problem is then formulated as

min 
$$J = \underline{c}_{E}^{T} Q_{O,E} \underline{c}_{E} + q_{O,E}^{T} \underline{c}_{E}$$
  
s.t.  $A_{E,eq} \underline{c}_{E} = b_{E,eq},$   
 $A_{E,ie} \underline{c}_{E} \leq b_{E,ie},$  (24)

where  $Q_{O,E} \in \mathbb{R}^{3(n_E+1)\times 3(n_E+1)}$  is the Hessian matrix of the objective function and it is semidefinite,  $q_{O,E} \in \mathbb{R}^{3(n_E+1)}$  is a vector,  $A_{E,eq} \in \mathbb{R}^{18\times 3(n_E+1)}$  and  $A_{E,ie} \in \mathbb{R}^{(9n_E+4)\times 3(n_E+1)}$  are constraint matrices,  $b_{eq} \in \mathbb{R}^{18}$  and  $b_{ie} \in \mathbb{R}^{9n_E+4}$  are constraint vectors. The linear equality constraint ( $A_{E,eq} = b_{E,eq}$ ) is endpoint constraints. The linear inequality constraint ( $A_{ie} = b_{ie}$ ) consists of dynamical feasibility, geometric feasibility, and grasp constraints. These constraints are adopted to ensure the solution of the problem (24) is collision-free and can be executed successfully. Definitions and roles of the objective and the constraints are as follows.

1) Objective Function: The objective function is denoted as  $J = J_J + J_O$ , where  $J_J$  is the cost function to minimize the jerk along the trajectory, and  $J_O$  is a penalty function for the collision. The details of the two terms are

$$J_{J} = \sum_{i=1}^{m} \int_{T_{i-1}}^{T_{i-1}} (j_{x}^{2}(t) + j_{y}^{2}(t) + j_{z}^{2}(t))dt, \qquad (25)$$
$$J_{O} = \sum_{k=1}^{n_{O}} \lambda_{k} \sum_{i=0}^{n_{E}} (c_{i,x} - x_{M,k})^{2} + (c_{i,y} - y_{M,k})^{2} + (c_{i,z} - z_{M,k})^{2}, \qquad (26)$$

where  $j_x$ ,  $j_y$ ,  $j_z$  denote the jerks of the trajectory in the corresponding three dimensions, respectively,  $\lambda_k$  is a changing weighting factor,  $x_{M,k}$ ,  $y_{M,k}$ ,  $z_{M,k}$  are corresponding elements of the obstacle mirror position  $p_{M,k}$ . The obstacle mirror position  $p_{M,k}$  is introduced to guide the trajectory of

Authorized licensed use limited to: Westlake University. Downloaded on May 01,2024 at 06:54:17 UTC from IEEE Xplore. Restrictions apply.

the end-effector away from obstacles. Due to the nature of the objective function, which encourages the trajectory to approach the obstacle mirror position, the iterative process enables the trajectory to move away from the obstacle. We define the obstacle mirror set as  $\mathbb{O}_M$  =  $\{\boldsymbol{p}_{M,1}, \boldsymbol{p}_{M,2}, \dots, \boldsymbol{p}_{M,n_O}\},\$  where  $n_O$  is the number of the obstacles that collide with the aerial manipulator during the whole iteration process. The obstacle mirror position  $p_{Mk}$ can be obtained through pinhole mapping of the corresponding obstacle position. As the iterations progress, the algorithm can guide the trajectory of the end-effector towards the obstacle mirror positions while ensuring that it is collision-free for the obstacles. See Section V-C for details of calculating the changing weighting factors  $\lambda_1, \lambda_2, \ldots, \lambda_{n_0}$  and the obstacle mirror set  $\mathbb{O}_M$ . By using the Lemma 1 into (25), we can obtain  $J = \underline{c}_E^T Q_{o,E} \underline{c}_E + q_{o,E}^T \underline{c}_E$ . We leave the details of the  $Q_{o,E}$ and  $\boldsymbol{q}_{o,E}$  for brevity.

2) Constraints: The constraints for the trajectory planning problem of the end-effector consist of endpoint, dynamical feasibility, geometric feasibility, and grasp constraints. The details of the constraints are given as follows:

The endpoint constraints are introduced to ensure the trajectory of the end-effector starts at  $p_{E,t_B}$  and ends at  $p_O$  with desired velocities and accelerations. The endpoint constraints are given as

$$c_{0,\mu,E} = \mu_{E,t_B}, s_E^{-1} c_{0,\mu,E}^{(1)} = \dot{\mu}_{E,t_B}, s_E^{-2} c_{0,\mu,E}^{(2)} = \ddot{\mu}_{E,t_B},$$
  

$$c_{n_E,\mu,E} = \mu_O, s_E^{-1} c_{n_E,\mu,E}^{(1)} = \dot{\mu}_{E,t_G}, s_E^{-2} c_{n_E,\mu,E}^{(2)} = \ddot{\mu}_{E,t_G}, \quad (27)$$

where  $c_{i,\mu,E}^{(k)}$  denotes the *i*-th control point of  $d^k f_{\mu}(\tau)/d\tau^k$  and can be calculated by Lemma 1,  $\mu \in [x, y, z]$ ,  $s_E = t_G - t_B$ ,  $p_{E,t_B}$ ,  $\dot{p}_{E,t_B}$ ,  $\ddot{p}_{E,t_B}$  can be obtained from the Section V-A and  $\dot{p}_{E,t_B} = 0$ ,  $\ddot{p}_{E,t_B} = 0$ 

The dynamical feasibility constraints consist of velocity and acceleration constraints to ensure the generated trajectory is dynamically feasible. The dynamical feasibility constraints are

$$\dot{\mu}_{\min} \leq s_E^{-1} c_{i,\mu,E}^{(1)} \leq \dot{\mu}_{\max}, i = 0, 1, \dots, n_E - 1, \ddot{\mu}_{\min} \leq s_E^{-2} c_{i,\mu,E}^{(2)} \leq \ddot{\mu}_{\max}, i = 0, 1, \dots, n_E - 2,$$
(28)

where  $\mu \in [x, y, z]$ , the subscript min denotes the lower bound of the corresponding variable, and the subscript max denotes the upper bound of the corresponding variable.

The geometric feasibility constraints are introduced to ensure the trajectories of the quadcopter and the end-effector are geometrically feasible for the Delta arm. According to (13), it can be describe as

$$\boldsymbol{R}_{\psi_O} \boldsymbol{w}_{\min} \le \boldsymbol{p}_{E, \text{ref}}(t) - \boldsymbol{p}_{B, \text{ref}}(t) = \boldsymbol{R}_{\psi_O} \boldsymbol{w}_{\max}.$$
 (29)

As stated above, the trajectory of the end-effector is represented by a  $n_E$ -th order Bézier curve. The trajectory of the quadcopter is part of a  $n_B$ -th order Bézier curve.

To reveal the geometric feasibility constraints on the parameters, we use a  $n_E$ -th order Bézier curve to fit the trajectory of the quadcopter from  $t_B$  to  $t_G$ . Then, the geometric feasibility constraints on the parameters can be formulated as linear algebraic equations. Let  $p_{B,0}, p_{B,1}, \ldots, p_{B,n_E}$  denote  $n_E + 1$  points of the trajectory  $p_{B,ref}(t), t \in [t_B, t_G]$ . These points divide the trajectory into  $n_E$  segments. The time interval between each two adjacent points is the same. The  $n_E$ -th order Bézier curve is denoted as  $\mathbf{h}(t) = \sum_{i=0}^{n_E} \mathbf{c}_{B,i} b_{i,n_E}(\tau_M)$ , where  $\mathbf{c}_{B,i} = [c_{B,x,i}, c_{B,y,i}, c_{B,z,i}]^T$  is the *i*-th control point of  $\mathbf{h}(t)$ . The control points can be obtained by fitting  $\mathbf{h}(t)$  to the points  $\mathbf{p}_{B,0}, \mathbf{p}_{B,1}, \ldots, \mathbf{p}_{B,n_E}$ . Then, the geometric feasibility constraints can be rewritten as

$$\boldsymbol{R}_{\psi_{O}}\boldsymbol{w}_{\min} \leq \sum_{i=0}^{n_{E}} (\boldsymbol{c}_{E,i} - \boldsymbol{c}_{B,i}) b_{i,n_{E}}(\tau_{M}) \leq \boldsymbol{R}_{\psi_{O}} \boldsymbol{w}_{\max}, \quad (30)$$

According to the convex hull property (see Lemma 2), the geometric feasibility constraints on the parameters is

$$w_{r,\mu,\min} + c_{B,\mu,i} \le c_{E,\mu,i} \le w_{r,\mu,\max} + c_{B,\mu,i},$$
  
 $i = 0, 1, \dots, n_E,$  (31)

where  $w_{r,\mu,\min}$  is the element of  $\mathbf{R}_{\psi_0} \mathbf{w}_{\min}$ ,  $w_{r,\mu,\max}$  is the element of  $\mathbf{R}_{\psi_0} \mathbf{w}_{\max}$ , and  $\mu \in \{x, y, z\}$ .

The grasp constraints are introduced to ensure the gripper does not collide with the object. To avoid such collision, we let the end of the trajectory in a cone. Let  $t_C$  denote the time to enter the cone. Let  $p_{E,ref}(t_C) = [x_{E,t_C}, y_{E,t_C}, z_{E,t_C}]^T$  denote the position of the end-effector at the time  $t_c$ . Then, we have

$$-\tan \gamma \leq \frac{x_{E,t_C} - x_O}{z_{E,t_C} - z_O} \leq \tan \gamma,$$
  
$$-\tan \gamma \leq \frac{y_{E,t_C} - y_O}{z_{E,t_C} - z_O} \leq \tan \gamma,$$
 (32)

where  $\gamma$  is the angle of the cone. By substituting the defined (23) into (32), the grasp constraints (32) can be rewritten as a linear form

$$\sum_{i=0}^{n_{E}} (c_{E,x,i} + c_{E,z,i} \tan \gamma) b_{i,n_{E}}(\tau_{C}) \leq x_{O} + z_{O} \tan \gamma,$$

$$\sum_{i=0}^{n_{E}} (c_{E,x,i} - c_{E,z,i} \tan \gamma) b_{i,n_{E}}(\tau_{C}) \geq x_{O} - z_{O} \tan \gamma,$$

$$\sum_{i=0}^{n_{E}} (c_{E,y,i} + c_{E,z,i} \tan \gamma) b_{i,n_{E}}(\tau_{C}) \leq y_{O} + z_{O} \tan \gamma,$$

$$\sum_{i=0}^{n_{E}} (c_{E,y,i} - c_{E,z,i} \tan \gamma) b_{i,n_{E}}(\tau_{C}) \geq y_{O} - z_{O} \tan \gamma,$$
(33)

where  $\tau_C = (t_C - t_B)/(t_G - t_B)$ .

# C. Collision Avoidance

The subsection proposes a method to detect collisions and calculate the changing weighting factors  $\lambda_1, \lambda_2, \ldots, \lambda_{n_0}$  and the obstacle mirror set  $\mathbb{O}_M$  in (26). Before the iteration process,  $\mathbb{O}_M$  is set as an empty set, i.e.,  $\mathbb{O}_M = \emptyset$ , and  $n_0$  is set as zero. At each iteration, the collision is detected using the solution of the QP problem (24). If the solution is collision-free, the iteration process is terminated and the collision-free solution is outputted as the trajectory of the end-effector. If there are collisions between the aerial manipulator and the obstacles in the environment with the solution, we calculate the changing weighting factors and  $\mathbb{O}_M$  for the next iteration calculation.

The collision detection method is proposed to detect if the solution of the QP problem (24) is collision-free. The 10

Algorithm 3 Collision Avoidance Method

Input:  $p_{E,ref}$ , M **Output:**  $\mathbb{O}_M$ ,  $\lambda_i$ 1: **function** COLLISION\_AVOID( $p_{E,ref}(t)$ ,  $\mathbb{M}$ ) Shape\_polyhedron  $\leftarrow$  calculating (34) and (36) 2: 3:  $\mathbb{M}_{\text{local}} \leftarrow \text{calculating (38)}$ 4:  $T_{i,L}, T_{i,R} \leftarrow \text{GJK}(\text{Shape_polyhedron}, \mathbb{M}_{\text{local}})$  $\mathbb{O}_M \leftarrow \text{calculating (40)}$ 5:  $\Delta \lambda_i \leftarrow \text{calculating (41)}$ 6:  $\lambda_i = \lambda_i + k_\alpha \Delta \lambda_i$ 7: return  $\mathbb{O}_M, \lambda_i$ 8: 9: end function

method considers the collision of the Delta arm and the endeffector. The trajectory of the quadcopter is collision-free, which is ensured by the flight corridor. Therefore, we do not consider the collision of the quadcopter. The proposed collision detection method consists of three steps.

We first use a shape polyhedron to represent the Delta arm and the end-effector in collision detection. The vertices of the shape polyhedron are  $p_{U,i}$ ,  $p_{L,i}$ , i = 1, 2, 3 (see the blue points in Fig.2(c)). Then, we have

$$\boldsymbol{p}_{U,i} = \boldsymbol{p}_B + \boldsymbol{R}_{\psi} \boldsymbol{R}_D^B \tilde{\boldsymbol{p}}_{U,i}, i = 1, 2, 3, \tag{34}$$

where

$$\tilde{\boldsymbol{p}}_{U,i} = \begin{bmatrix} r_S \cos(\frac{1+2i}{3}\pi) \\ r_S \sin(\frac{1+2i}{3}\pi) \\ 0 \end{bmatrix}$$
(35)

In addition, the lower vertices can be calculated by

$$\boldsymbol{p}_{L,i} = \boldsymbol{P}_E + \boldsymbol{R}_{\psi} \, \boldsymbol{R}_D^B \, \tilde{\boldsymbol{p}}_{L,i}, \, i = 1, 2, 3, \tag{36}$$

where

$$\tilde{\boldsymbol{p}}_{L,i} = \begin{bmatrix} l_C \cos(\frac{1+2i}{6}\pi) \\ l_C \sin(\frac{1+2i}{6}\pi) \\ l_C \end{bmatrix}$$
(37)

where  $l_C$  is a constant parameter for safety and is determined by the size of the gripper when the gripper is open.

Second, we introduce a local map to reduce the computational cost of collision detection. This is because detecting collisions in the entire environment can be computationally expensive, especially if the environment is large or if there are many obstacles. We decrease the number of obstacles to be calculated by adding a box around the end-effector and the object and thus only detecting collisions inside it. The size of the box is determined by  $p_{B,t_B}$  and  $p_Q$ . We let

$$\mathbb{M}_{\text{local}} = \{ \boldsymbol{p} \in \mathbb{R}^3 | \boldsymbol{l}_{\min} \le \boldsymbol{p} \le \boldsymbol{l}_{\max} \}$$
(38)

denote the box. In addition,  $l_{\min}$  and  $l_{\max}$  can be calculated by

$$l_{\mu,\min} = \min\{\mu_{B,t_B}, \mu_O\} - l_s, l_{\mu,\max} = \max\{\mu_{B,t_B}, \mu_O\} + l_s,$$
(39)

where  $\mu \in \{x, y, z\}$ ,  $l_{\mu,\min}$ ,  $l_{\mu,\max}$ ,  $\mu_{B,t_B}$ ,  $\mu_O$  are corresponding element of  $l_{\min}$ ,  $l_{\min}$ ,  $p_{B,t_B}$ ,  $p_O$ , respectively,  $l_s$  is a constant parameter for safety.

Third, we use the GJK method proposed in [53] to detect collisions. If the QP problem solution reveals a collision between the aerial manipulator and obstacle *i*, then the two endpoints  $T_{i,L}$  and  $T_{i,R}$  of the solution within the collision area with respect to the obstacle can be determined (see Fig. 4). Let  $O_i$  denote the center of the obstacle *i*. Then, we calculate the obstacle mirror position  $p_{M,i}$  by the following pinhole mapping method. Let  $p_{P,i} = 0.5(T_{i,L} + T_{i,R}) \in \mathbb{R}^3$  denote the position of the pinhole. Then, we let  $\alpha_i = p_{P,i} - O_i$  represent the distance between the obstacle and the pinhole. According to the pinhole mapping, we have

$$\boldsymbol{p}_{M,i} = \begin{cases} 2\boldsymbol{p}_{P,i} - \boldsymbol{O}_i, & \text{if } z_{\alpha_i} \ge 0 \text{ and } \|\boldsymbol{\alpha}_i\| \ge \delta_{\alpha} \\ \boldsymbol{O}_i + 2\delta_{\alpha}\boldsymbol{\alpha}_i / \|\boldsymbol{\alpha}_i\|, & \text{if } z_{\alpha_i} \ge 0 \text{ and } 0 < \|\boldsymbol{\alpha}_i\| < \delta_{\alpha} \\ \boldsymbol{O}_i + 2\delta_{\alpha}\boldsymbol{e}_3, & \text{if } \|\boldsymbol{\alpha}_i\| = 0 \\ \boldsymbol{O}_i - 2\delta_{\alpha}\boldsymbol{\alpha}_i / \|\boldsymbol{\alpha}_i\|, & \text{if } z_{\alpha_i} < 0, \end{cases}$$

$$(40)$$

where  $z_{\alpha_i} = e_3^T \alpha_i$  is the 3-th element of  $\alpha_i$ ,  $\delta_{\alpha} > 0$  is constant,  $e_3 = [0, 0, 1]^T$ . The values of the weighting factors are updated by  $\lambda_i = \lambda_i + k_{\alpha} \Delta \lambda_i$ , where  $i = 1, 2, ..., n_O$ , and  $k_{\alpha} > 0$  is a constant gain. The parameter  $\Delta \lambda_i$  is the step size used for updating the *i*-th weighting factor and is a critical factor that affects the computation time of the method. The expression for  $\Delta \lambda_i$  is given as

$$\Delta \lambda_i = \| \boldsymbol{T}_{i,L} - \boldsymbol{T}_{i,R} \|.$$
(41)

The proposed collision avoidance method can ensure the resulting trajectory is collision-free. It is shown as follows. According to (40), there are four possible cases. When  $z_{\alpha_i} \geq$ 0 and  $\|\boldsymbol{\alpha}_i\| \geq \delta_{\alpha}$ , the iteration process is shown in Fig. 5(a). As the iteration proceeds, the calculated mirror position  $p_{M_i}$ gradually moves away from the obstacle i. As a result, the candidate trajectory also moves away from the obstacle *i* with the influence of the cost function that contains the distance to  $p_{M,i}$ . When  $z_{\alpha_i} \ge 0$  and  $0 < \|\boldsymbol{\alpha}_i\| < \delta_{\alpha}$ , the iteration process is shown in Fig. 5(b). After several iterate steps, the candidate trajectory is moved upward and  $\|\boldsymbol{\alpha}_i\| \geq \delta_{\alpha}$ . Now, it becomes the first case. Based on the above analysis, the collision-free trajectory can be obtained after the iteration process. When  $\|\boldsymbol{\alpha}_i\| = 0$ , the iteration process is shown in Fig. 5(c). Since  $\boldsymbol{\alpha}_i = 0$ , the direction vector  $\boldsymbol{\alpha}_i / \|\boldsymbol{\alpha}_i\|$  can not be used to calculate  $p_{M,i}$ . We let  $p_{M,i} = O_i + 2\delta_{\alpha}e_3$  to move the candidate trajectory upward. After several iterate steps, we have  $z_{\alpha_i} \geq 0$  and  $0 \leq \|\boldsymbol{\alpha}_i\| < \delta_{\alpha}$ . Now, it becomes the second case. Based on the above analysis, the collision-free trajectory can be obtained after the iteration process. When  $z_{\alpha_i} < 0$ , the iteration process is shown in Fig. 5(d). In this case, the vector  $\boldsymbol{\alpha}_i$  points downward. However, the aerial manipulator can not approach obstacles from below. To move the candidate trajectory upward, we let  $p_{M,i} = O_i - 2\delta_\alpha \alpha_i / \|\alpha_i\|$  when  $z_{\alpha_i} < 0$ . With the given  $p_{M,i}$ , the candidate trajectory can move upward. After several iterate steps, we have  $z_{\alpha_i} \geq 0$ . Now, it becomes the first three cases. Based on the above analysis, the collision-free trajectory can be obtained after the iteration process.

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 5. The role of the obstacle mirror position in the iteration process.

#### D. Local Optimums Analysis

The proposed method may get stuck in local optimums if the obstacles are extremely complex. This is because it is a hybrid method that combines soft and hard-constrained trajectory optimization methods. Although the hard constraints are convex, the soft constraints related to obstacle avoidance can become non-convex in cases involving complex obstacles [54]. However, in such cases, other existing methods (e.g., the sampling-based method [30], the artificial potential method [37], the control-barrier-function method [38]) also encounter challenges when applied to solve the motion planning problem. The sampling-based method suffers from the problem of increased computation in high-dimensional configuration spaces and highdensity sampling. The artificial potential method and the control-barrier-function method also may be stuck in local optimums.

There are two techniques that can be used to avoid getting stuck in local optimums. First, we can merge obstacles that are close together into one obstacle. The quickhull algorithm [55] is used to merge the obstacles. It can quickly find the convex hull containing the obstacles. Second, we simplify obstacles with complex shapes into simple convex polyhedrons. This technique can avoid getting stuck in local optimums due to non-convex obstacles.

## VI. EXPERIMENTAL VERIFICATION

This section presents experimental results to verify the effectiveness of the proposed motion planning algorithms. The experimental video is available at https://youtu.be/q709v7l2Oho.

First of all, we describe the experimental setup. The aerial manipulator platform used in the experiments consists of a quadcopter and a Delta arm. The wheelbase of the quadcopter is 0.65 m. The mass of the quadcopter (including a battery) is 3.60 kg. The Delta arm consists of a mounting base (0.56 kg), a movable robotic arm (0.44 kg), and a gripper (0.32 kg). An ESO-based flight controller proposed in our previous work [44] runs on a Pixhawk 4 autopilot. This controller uses extended state observers (ESOs) to estimate dynamic coupling between the aerial manipulator and the Delta arm. The proposed motion planning method runs on an onboard Intel NUC i7-1165G7 computer with ROS (an open-source robotics middleware suite). The Delta arm and the end-effector are controlled by four actuators. We use Dynamixel AX28 servomotors as the actuators of the Delta arm and the end-effector. The controllers of the servomotors are designed by the traditional PID method [56]. The experiments are conducted in a Vicon system, which provides accurate position measurements of the quadcopter base and the end-effector. The measurement data of the Vicon system



Fig. 6. Results of the collision avoidance experiment.

is sent to a ground control station through an ethernet switch. Then, the ground control station sends the measurement data to the aerial manipulator with a frequency of 100 Hz through a 5 GHz wireless router.

The perception of the aerial manipulator is not surveyed in this paper. We assume that the obstacles in the environment are already known. In particular, the locations of the obstacles and the object can be obtained by the Vicon system. Then, the environment can be previously built as a grid map which consists of a set of cubes. The size of each cube is set as 0.1 m. This map is used for the path planning of the quadcopter base. The description of the controllers is provided in Section II-B.

In all the examples, we use the same set of parameters of the motion planner:  $\alpha = 3.0$ ,  $r_S = 0.50$  m,  $l_C = 0.06$  m,  $l_S = 0.20$  m. The velocity and the acceleration constraints for the quadcopter base are set as 1.0 m/s and 1.0 m/s<sup>2</sup>, respectively. The velocity and the acceleration constraints for the end-effector are set as 1.0 m/s and 2.0 m/s<sup>2</sup>, respectively. The bounds of the geometric feasibility constraints are set as  $\boldsymbol{w}_{\min} = [-0.06, -0.06, -0.60]^T$  and  $\boldsymbol{w}_{\max} = [0.06, 0.06, -0.40]^T$ .

#### A. Example 1: Collision Avoidance

We validate the effectiveness of the proposed method in the collision avoidance task. The environment of this example is illustrated in Fig. 6. There are two types of obstacles in the environment. The first type of obstacles restrict the motion of the quadcopter base and the size of the flight corridor. The collision avoidance for this type of obstacles is achieved by the flight corridor. The second type of obstacles restrict the motion planning of the Delta arm. In Section V-C, we propose an iterative collision avoidance method to avoid the second type of obstacles. In order to show its effectiveness, the result of the motion planning with the collision avoidance method is calculated.

Fig. 6 shows the results of the motion planning with the collision avoidance method and without the collision avoidance method. The generated flight corridor is shown in Fig. 6(a). As shown in Fig. 6(b), there are four obstacles near the object. In particular, the aerial manipulator collides with one of these obstacles in the resulting trajectory without the collision avoidance method. The collision area is shown as a red dot line in Fig. 6 and its length is 0.26 m. The resulting trajectory with the collision avoidance method is shown in Fig. 6 (see the blue line). As can be seen, the result of the proposed method is collision-free. The computational time for calculating the path, flight corridor, and trajectory of the quadcopter in the collision avoidance task is 11.8 ms, 20.7 ms, and 13.8 ms, respectively. The computational time for calculating the trajectory of the end-effector is 36.4 ms.

# B. Example 2: Aerial Retrieval

The goal of this experiment is to retrieve an object by the aerial manipulator. In the task, the aerial manipulator moves to and picks up the object. Then, the aerial manipulator returns to the start position. The start position is set as [0, 0, -2.00]. The position and the orientation angle of the object are set as [0, -2.00, -1.24] and  $0^{\circ}$  (see Fig. 7(a)), respectively. As shown in Fig. 7(b), screens are in the flight environment, which needs to be avoided by the aerial manipulator in the moving stage. The aerial manipulator is set to fly around the screens. As shown in Fig. 7(a), there are three obstacles near the object. The aerial manipulator has to avoid colliding with these obstacles.

Fig. 7(b)-(d) shows the result of the aerial retrieval experiment. The generated flight corridor is shown in Fig. 7(c). The task duration of the experiment is 58 s. The mean tracking error of the quadcopter base in the moving stage is 0.05 m, while that in the manipulation stage is 0.01 m. The quadcopter flies faster in the moving stage than in the manipulation stage. However, the higher velocity also causes a larger tracking error. The computational time for calculating the path, flight corridor, and trajectory of the quadcopter in the aerial retrieval task is 11.9 ms, 23.6 ms, and 11.9 ms, respectively. The computational time for calculating the trajectory of the end-effector is 25.7 ms. The attitude of the quadcopter in the manipulation stage is shown in Fig. 7(e). The roll angle ranges from

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 7. Results of the aerial retrieval experiment.

 $0^{\circ}$  to  $2^{\circ}$ , and the pitch angle spans from  $-4^{\circ}$  to  $0^{\circ}$ . This confirms that the assumption that the pitch and roll angles are small in the manipulation stage is reasonable.

# C. Example 3: Aerial Retrieval in the Complex Environment

To demonstrate the performance of our proposed method, we conducted simulations in Gazebo and compared its results with those of two baseline methods. The first baseline method is a sampling-based method and proposed in [30]. This method plans the path in the configuration space of the aerial manipulator by the RRT\* method and calculates the trajectories based on the Bézier curve. The second baseline method is a naive approach that decouples the planning problem for the aerial platform and the Delta arm [33]. In this method, the quadcopter flies over the object, descends in altitude, and then stays hovering to manipulate the object. The trajectory of the quadcopter can be calculated by the method proposed in Section IV. The obstacles near the target can influence the calculations of the motion planning (see Fig. 8(a)). The flight environment of this experiment is intentionally designed to be complex, aiming to simulate real-world scenarios with a higher level of complexity (see Fig. 8(b)).

Fig. 8(c) shows the results of the three methods. The proposed method outputs the trajectories of the quadcopter and the end-effector in  $\Sigma_I$ . The sampling-based method provides trajectories in the configuration space, resulting in the trajectory of the quadcopter in  $\Sigma_I$  and the trajectory of the end-effector in  $\Sigma_D$ . For display purposes, the result of the end-effector is converted into the trajectory in  $\Sigma_I$ . The naive method outputs the trajectory of the quadcopter in  $\Sigma_I$ .

To further verify the performance of the proposed method, a quantitative comparison between the results of the proposed

Qι

IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING



Fig. 8. Results of the aerial retrieval experiment in the complex environment.

|                        |                            | TABLE I   |  |     |
|------------------------|----------------------------|---|--|-----|
| JANTITATIVE COMPARISON | BETWEEN THE RESULTS OF THE | PROPOSED METHOD, THE SAMP   | LING-BASED METHOD, AND THE NAIVE METH                  | HOD |
| N ( 1                  | $\mathbf{C}$               | $\mathbf{D}_{1} = (1^{2} + 1^{2}$ | $\mathbf{D}$ and $\mathbf{C}$ (1) and $\mathbf{C}$ (1) |     |

| Method                    | Computational time (s) | Duration of the whole task (s) | Duration of the manipulation stage (s) |
|---------------------------|------------------------|--------------------------------|--|
| The proposed method       | 0.19                   | 72.0                           | 8.0                                    |
| The sampling-based method | 89.41                  | 96.0                           | 16.0                                   |
| The naive method          | 0.17                   | 85.0                           | 25.0                                   |

method, the sampling-based method, and the naive method is given in Table I. Compared with the sampling-based method, the proposed method performs better in terms of computational time and task duration. The computational time of the proposed method is 0.19 s, while the computational time of the sampling-based method is 89.41 s. Compared to the sampling-based method, our proposed method can reduce the computational time by 99.7%. This is because sampling-based methods necessitate dense sampling in a 6-DoF configuration space to avoid obstacles, which significantly increases computational time. The duration of the whole task using the proposed method and the sampling-based method is 72.0 s and 96.0 s, respectively. Compared to the sampling-based method, our proposed method can reduce the task duration by 25.0%. This reduction is particularly significant during the manipulation stage, where it is reduced by 50%. This is because the proposed method allows real-time closed-loop tracking of the end effector in Cartesian space, enabling fast completion of the task. Real-time closed-loop tracking of the end-effector can reduce the duration time by allowing higher velocities. The error of the end-effector increases with higher

velocities of the aerial manipulator. Real-time closed-loop tracking helps mitigate this error. In contrast, without real-time closed-loop tracking, the end-effector's error may become too large, resulting in task failure. Therefore, without real-time closed-loop tracking, the aerial manipulator's velocity must be kept relatively low. This leads to a longer duration time for the baseline method compared to the proposed method. Compared with the naive method, the proposed method can reduce the duration of the manipulation stage by 68.0%. This is because, in the naive method, the quadcopter needs to remain stationary for a certain period to ensure the accuracy of the manipulation.

*Remark 1:* The comparison between the proposed method and the first baseline method is mainly to highlight the advantages of the proposed decoupled motion planning framework over the coupled algorithms that plan trajectories in the configuration space with a sampling-based method. The resulting computational time of this method is 89.41 s while our's is significantly reduced. The resulting computational time is reasonable since a similar result can also be found in Table I of [30].

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS



Fig. 9. Results of the aerial transport experiment.

# D. Example 4: Aerial Transport

The goal of this experiment is to grasp an object and put it on the target location. In the task, the aerial manipulator first flies to and picks up the object. Then, the aerial manipulator flies to the target location and puts the object on the target location. Finally, the aerial manipulator returns to the start position. The start position is set as [0, 0, -2.00]. The position and the orientation angle of the object are set as [0, 2.00, -1.22] and  $0^\circ$ , respectively. The position of the target location is set as [0, -2.00, -1.24]. The whole process of the experiment is shown in Fig. 9(c). In the experiment, the aerial manipulator is also set to fly around the screens to utilize the experiment field. As shown in Fig. 9(a) and (b), there are two obstacles near the object and three obstacles near the target location. The aerial manipulator has to avoid colliding with these obstacles.

Fig. 9(c)-(e) show the result of the aerial transport experiment. The generated flight corridor is shown in Fig. 9(d). The task duration of the experiment is 88 s. The mean tracking error of the quadcopter base in the moving stage is 0.06 m, while that in the manipulation stage is 0.01 m. The computational time for calculating the path, flight corridor,

and trajectory of the quadcopter in the aerial transport task is 12.3 ms, 19.8 ms, and 13.5 ms, respectively. The computational time for calculating the trajectory of the end-effector is 48.6 ms. The experiment result validates the effectiveness of the proposed motion planning method in the aerial transport task.

# E. Example 5: Replanning

Since the proposed method is computationally efficient, it inherently enables the capability for replanning. The goal of this experiment is to demonstrate the proposed method's capability to replan trajectories in response to changes in the object's position. Initially, the proposed planner of the aerial manipulator generates trajectories for retrieving the object from its original location at [0, -1.8, -1.24]. Subsequently, the object is relocated to a new position at [0, -2.2, -1.24], prompting the planner to generate new trajectories for retrieving the object from its updated location. As shown in Fig. 10(a), numerous obstacles are present near the object, necessitating the aerial manipulator to navigate carefully to avoid collisions with them. In the experiment, the aerial



Fig. 10. Results of the replanning experiment.

manipulator is also set to fly around the screens to utilize the experiment field (see Fig. 10(b)).

Fig. 10(c) illustrates the results of the replanning experiment. The blue line represents the original trajectory generated by the proposed planner, while the green dotted line depicts the replanned trajectory in response to the change in the object's position. The computational time for the original planning is 82.3 ms, whereas the replanning process takes 36.2 ms. The reduced computational time for replanning is attributed to the absence of the need to recalculate the path and flight corridor for the quadcopter. This experiment demonstrates that the proposed method is capable of replanning.

# VII. CONCLUSION

This paper proposed a novel partially decoupled motion planning scheme of the aerial manipulator for the aerial pick-and-place task. This method calculates the dynamically feasible and collision-free trajectories of the flying base and the manipulator respectively in Cartesian space. The proposed geometric feasibility constraints can ensure the resulting trajectories are coordinated to complete tasks. The proposed method is verified by five experimental results. It is verified by these experiments that the proposed geometric feasibility constraints can ensure the trajectories of the quadcopter base and the end-effector satisfy the geometry of the aerial manipulator. The results also illustrate the ability of the proposed method to avoid obstacles. This ability is limited by the partially decoupled structure since the obstacles near the object are avoided by the Delta arm rather than the whole aerial manipulator. However, in order to avoid large obstacles near the object, both the quadcopter base and the Delta arm must be used. Algorithms such as the bidirectional RRT and optimization-based methods could be used to solve this challenge. This will be one important research direction for future research. Although the proposed method is computationally efficient and has the potential to replan new trajectories for the aerial manipulator in response to unexpected situations, the development of an efficient replanning framework is also necessary. This constitutes another important research direction for the future.

#### REFERENCES

- A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 626–645, Feb. 2022.
- [2] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chin. J. Aeronaut.*, vol. 32, no. 1, pp. 200–214, Jan. 2019.
- [3] A. Mohiuddin, T. Tarek, Y. Zweiri, and D. Gan, "A survey of single and multi-UAV aerial manipulation," *Unmanned Syst.*, vol. 8, no. 2, pp. 119–147, Apr. 2020.
- [4] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1957–1964, Jul. 2018.
- [5] H. Nguyen, S. Park, J. Park, and D. Lee, "A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 353–369, Apr. 2018.
- [6] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac, "An integrated delta manipulator for aerial repair: A new aerial robotic system," *IEEE Robot. Autom. Mag.*, vol. 26, no. 1, pp. 54–66, Mar. 2019.

CAO et al.: MOTION PLANNING FOR AERIAL PICK-AND-PLACE WITH GEOMETRIC FEASIBILITY CONSTRAINTS

- [7] K. Bodie et al., "Active interaction force control for contact-based inspection with a fully actuated aerial vehicle," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 709–722, Jun. 2021.
- [8] K. Zhang et al., "Aerial additive manufacturing with multiple autonomous robots," *Nature*, vol. 609, no. 7928, pp. 709–717, 2022.
- [9] R. Naldi, A. Macchelli, N. Mimmo, and L. Marconi, "Robust control of an aerial manipulator interacting with the environment," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 537–542, 2018.
- [10] K. Bodie, M. Tognon, and R. Siegwart, "Dynamic end effector tracking with an omnidirectional parallel aerial manipulator," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8165–8172, Oct. 2021.
- [11] Y. Chen, J. Liang, Y. Wu, Z. Miao, H. Zhang, and Y. Wang, "Adaptive sliding-mode disturbance observer-based finite-time control for unmanned aerial manipulator with prescribed performance," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3263–3276, May 2023.
- [12] T. G. Chen, K. A. W. Hoffmann, J. E. Low, K. Nagami, D. Lentink, and M. R. Cutkosky, "Aerial grasping and the velocity sufficiency region," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10009–10016, Oct. 2022.
- [13] H. Lee, H. Kim, and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 189–201, Jan. 2018.
- [14] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: Planning and control," *Auton. Robots*, vol. 40, no. 4, pp. 631–655, Apr. 2016.
- [15] M. Tognon, E. Cataldi, H. A. T. Chavez, G. Antonelli, J. Cortés, and A. Franchi, "Control-aware motion planning for task-constrained aerial manipulation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2478–2484, Jul. 2018.
- [16] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2014, pp. 2113–2118.
- [17] H. Seo, S. Kim, and H. J. Kim, "Aerial grasping of cylindrical object using visual servoing based on stochastic model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6362–6368.
- [18] Y. Chen et al., "Image-based visual servoing of unmanned aerial manipulators for tracking and grasping a moving target," *IEEE Trans. Ind. Informat.*, vol. 19, no. 8, pp. 8889–8899, Aug. 2023.
- [19] P. Ramon-Soria, B. C. Arrue, and A. Ollero, "Grasp planning and visual servoing for an outdoors aerial dual manipulator," *Engineering*, vol. 6, no. 1, pp. 77–88, Feb. 2020.
- [20] D. K. D. Villa, A. S. Brand ao, and M. Sarcinelli-Filho, "A survey on load transportation using multirotor UAVs," *J. Intell. Robot. Syst.*, vol. 98, no. 2, pp. 267–296, May 2020.
- [21] J. Zha and M. W. Mueller, "Exploiting collisions for sampling-based multicopter motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.* (ICRA), May 2021, pp. 7943–7949.
- [22] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-repeatreplan: A complete and robust system for aggressive flight in complex environments," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1526–1545, Oct. 2020.
- [23] T. Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators," *IEEE J. Robot. Autom.*, vol. RA-3, no. 3, pp. 224–238, Jun. 1987.
- [24] C. Yuan, W. Zhang, G. Liu, X. Pan, and X. Liu, "A heuristic rapidlyexploring random trees method for manipulator motion planning," *IEEE Access*, vol. 8, pp. 900–910, 2020.
- [25] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 844–851, Apr. 2020.
- [26] L. Jiang, S. Liu, Y. Cui, and H. Jiang, "Path planning for robotic manipulator in complex multi-obstacle environment based on improved\_RRT," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 4774–4785, Dec. 2022.
- [27] A. Ivanovic, M. Car, M. Orsag, and S. Bogdan, "Exploiting null space in aerial manipulation through model-in-the-loop motion planning," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 686–693.
- [28] A. Caballero et al., "First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 8471–8477.
- [29] A. Caballero, P. J. Sanchez-Cuevas, M. Bejar, G. Heredia, M. A. Trujillo, and A. Ollero, "An aerodynamic extension for motion planning with dynamics awareness in aerial long-reach manipulators," *Int. J. Aerosp. Eng.*, vol. 2020, pp. 1–17, Sep. 2020.

- [30] H. Kim, H. Seo, J. Kim, and H. J. Kim, "Sampling-based motion planning for aerial pick-and-place," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7402–7408.
- [31] W. Luo, J. Chen, H. Ebel, and P. Eberhard, "Time-optimal handover trajectory planning for aerial manipulators based on discrete mechanics and complementarity constraints," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4332–4349, Dec. 2023.
- [32] S. R. Lindemann and S. M. LaValle, "Current issues in sampling-based motion planning," in Proc. 11th Int. Symp. Robot. Res., 2005, pp. 36–54.
- [33] G. Garimella, M. Sheckells, S. Kim, G. Baraban, and M. Kobilarov, "Improving the reliability of pick-and-place with aerial vehicles through fault-tolerant software and a custom magnetic end-effector," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7501–7508, Oct. 2021.
- [34] J. Welde, J. Paulos, and V. Kumar, "Dynamically feasible task space planning for underactuated aerial manipulators," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3232–3239, Apr. 2021.
- [35] M. Zucker et al., "CHOMP: Covariant Hamiltonian optimization for motion planning," *Int. J. Robot. Res.*, vol. 32, no. 9, pp. 1164–1193, Aug. 2013.
- [36] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, Aug. 2014.
- [37] E. Rimon, Exact Robot Navigation Using Artificial Potential Functions. New Haven, CT, USA: Yale Univ. Press, 1990.
- [38] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2021, pp. 8129–8136.
- [39] M. Müller, "A novel classification of planar four-bar linkages and its application to the mechanical analysis of animal systems," *Philos. Trans. Roy. Soc. London, B, Biol. Sci.*, vol. 351, no. 1340, pp. 689–720, 1996.
- [40] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control.* London, U.K.: Springer, 2010.
- [41] N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [42] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 344–351.
- [43] A. Caballero, M. Bejar, A. Rodriguez-Castaño, and A. Ollero, "Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 3, May 2018, Art. no. 172988141877052.
- [44] H. Cao, Y. Li, C. Liu, and S. Zhao, "ESO-based robust and high precision tracking control for aerial manipulation," *IEEE Trans. Autom. Sci. Eng.*, early access, pp. 1–17, 2023.
- [45] H. Prautzsch, W. Boehm, and M. Paluszny, Bézier and B-Spline Techniques, vol. 6. London, U.K.: Springer, 2002.
- [46] M. López, E. Castillo, G. García, and A. Bashir, "Delta robot: Inverse, direct, and intermediate Jacobians," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 220, no. 1, pp. 103–109, Jan. 2006.
- [47] X. Yang, Z. Feng, C. Liu, and X. Ren, "A geometric method for kinematics of delta robot and its path tracking control," in *Proc. 14th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2014, pp. 509–514.
- [48] J. E. Shigley, C. R. Mischke, and T. H. Brown, Standard Handbook of Machine Design. New York, NY, USA: McGraw-Hill, 2004.
- [49] M. A. Laribi, L. Romdhane, and S. Zeghloul, "Analysis and dimensional synthesis of the delta robot for a prescribed workspace," *Mechanism Mach. Theory*, vol. 42, no. 7, pp. 859–870, Jul. 2007.
- [50] S. Mondal, A. Biswas, and A. Sarkar, "Finding the largest empty cuboid inside a 3D digital object," *Multimedia Tools Appl.*, vol. 80, nos. 28–29, pp. 35521–35541, Nov. 2021.
- [51] S. Liu et al., "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017.
- [52] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2520–2525.
- [53] C. Jin Ong and E. G. Gilbert, "Fast versions of the Gilbert–Johnson–Keerthi distance algorithm: Additional results and comparisons," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 531–539, Aug. 2001.
- [54] M. Guo and M. M. Zavlanos, "Probabilistic motion planning under temporal tasks and soft constraints," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4051–4066, Dec. 2018.

- IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING
- [55] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996.
- [56] R. Fabre, Q. Rouxel, G. Passault, S. N. Guyen, and O. Ly, "Dynaban, an open-source alternative firmware for dynamixel servo-motors," in *Robot World Cup.* Cham, Switzerland: Springer, 2016, pp. 169–177.



Huazi Cao received the B.Eng. degree in aerospace engineering from Harbin Engineering University, Harbin, China, in 2013, and the Ph.D. degree in aerospace engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2018.

From 2018 to 2019, he was a Research Scientist with Southwest China Research Institute of Electronic Equipment, Chengdu, China. He is currently a Research Fellow with the School of Engineering, Westlake University, Hangzhou, China. His research

interests include the control and planning of aerial robotic systems.



Jiahao Shen received the B.Eng. degree in electrical engineering from China Jiliang University, Hangzhou, China, in 2019, and the M.Eng. degree in electrical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2022.

He is currently a Flight Control Engineer with the School of Engineering, Westlake University, Hangzhou. His research interests include hardware, software, and controller design of unmanned aerial vehicle systems. He was a co-recipient of the First

Prize of China Aeromodelling Design Challenge, Zhenjiang, China, in 2017.



**Cunjia Liu** (Member, IEEE) received the Ph.D. degree in autonomous vehicle control from Loughborough University in 2011.

In 2013, he became an Academic at Loughborough University and was promoted to a Reader in unmanned vehicles in 2021. He has a strong background in flight control, autonomous vehicles, and Bayesian estimation. His recent work focuses on novel information-gathering and planning algorithms for autonomous robotics and their applications in defense, security, environment monitoring, and pre-

cision agriculture. He currently serves as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS and *IEEE Robotics & Automation Magazine.* 



**Bo Zhu** (Member, IEEE) received the B.E. and Ph.D. degrees from Beihang University, Beijing, China, in 2004 and 2010, respectively.

He was a Lecturer and then promoted to an Associate Professor with the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in July 2013. He visited the FSC Laboratory, University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, ON, Canada, from 2013 to 2014. He is currently a tenured Associate Professor with Nanjing University (Suzhou

Campus). His research interests include disturbance rejection techniques, autonomous, dependable and affordable (ADA) control of swarm systems, and related applications.



Shiyu Zhao (Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from the National University of Singapore, Singapore, in 2014.

From 2014 to 2016, he was a Post-Doctoral Researcher with the Technion—Israel Institute of Technology, Haifa, Israel, and the University of California at Riverside, Riverside, CA, USA.

From 2016 to 2018, he was a Lecturer with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K. He is currently an Associate Professor with the School of Engineering, Westlake University, Hangzhou, China. His research interests include theories and applications of robotic systems.