An End-Effector-Oriented Coupled Motion Planning Method for Aerial Manipulators in Constrained Environments

Zhaopeng Zhang, Hai Yu, *Graduate Student Member, IEEE*, Yi Chai, Zhichao Yang, Xiao Liang, *Senior Member, IEEE*, Yongchun Fang, *Senior Member, IEEE*, and Jianda Han, *Member, IEEE*

***Abstract*—The aerial manipulator, composed of a multirotor and a robotic arm, could provide extra active operation capability and has the potential for applications in both military and civilian fields. Due to the complex working scenarios, planning a safe and reliable motion trajectory for aerial manipulators is of great importance. To achieve target grasping with aerial manipulators in complex environments, a coupled motion planning method for the aerial manipulator is proposed in this article, which conducts the motion planning for the multirotor and the robotic arm simultaneously. Specifically, the aerial manipulator is considered as enclosed by a convex polyhedron other than a big sphere so as to reduce the conservatism of the trajectory. Furthermore, the trajectory of the aerial manipulator could be represented by a set of polynomial functions of the flat outputs, including the position and the yaw angle of the multirotor and the joint angles of the robotic arm. Utilizing polynomial-based trajectory representation for both the multirotor's and the robotic arm's states simplifies the optimization problem and ensures dynamic feasibility. The simulation and experimental results show the effectiveness of the proposed method. Unlike the other works that only consider the position of the end-effector, the proposed method also takes into account the orientation of the end-effector as well, resulting in improved performance in grasping tasks.**

***Index Terms*—Aerial manipulator, motion planning, trajectory optimization.**

Nomenclature

Symbols	*Definitions*
$m \in \mathbb{R}$	Mass of the aerial manipulator.
$g \in \mathbb{R}$	Gravitational acceleration.
$J \in \mathbb{R}^{3\times3}$	Multirotor's moment of inertia.
$\boldsymbol{x} \in \mathbb{R}^3$	Multirotor's position vector with respect to $\{I\}$.
$R \in SO(3)$	Rotation matrix from $\{B\}$ to $\{I\}$.
$\boldsymbol{\theta} \in \mathbb{R}^2$	Joint angles of the robotic arm.
$\dot{\boldsymbol{\theta}} \in \mathbb{R}^2$	Joint angle rates of the robotic arm.
$\boldsymbol{\nu} \in \mathbb{R}^3$	Multirotor's linear velocity with respect to $\{I\}$.
$\boldsymbol{\omega} \in \mathbb{R}^3$	Multirotor's angular velocity with respect to $\{B\}$.
$f \in \mathbb{R}$	Thrust generated by the multirotor.
$\boldsymbol{\tau} \in \mathbb{R}^3$	Torque generated by the multirotor.
$\boldsymbol{\tau}_\theta \in \mathbb{R}^2$	Control input of the robotic arm.

I. Introduction

With the advantages of low cost, strong maneuverability, and vertical takeoff and landing capability, the unmanned aerial vehicle has been widely applied in military and civilian fields [1], [2], [3]. However, a single multirotor can only be used to perform noncontact tasks, such as aerial photography and environmental monitoring. To enhance the operational capabilities of the multirotor, researchers have equipped it with a multilink robotic arm, creating a new type of system known as the aerial manipulator.

Almost at the same time, Jimenez-Cano et al. [4] and Kim et al. [5] proposed different control frameworks for the aerial manipulator, which is referred to as decoupled control and coupled control, respectively [6]. The decoupled control strategy is to model the multirotor and the robotic arm separately and design two individual controllers for each part [4], while the coupled control strategy is to model the aerial manipulator as a whole system and design a single controller for it [5].

The aerial manipulator has been widely used in various fields with the help of the aforementioned coupled or decoupled control strategy. To solve the problem of opening a hinged door, Lee et al. [7] proposed the model predictive control (MPC) method with the coupled system dynamics between the aerial

Received 6 January 2025; accepted 4 March 2025. Recommended by Technical Editor S. Roy and Senior Editor M. Indri. This work was supported in part by the National Natural Science Foundation of China under Grant 62273187, Grant 62233011, and Grant 623B2054, in part by the Natural Science Foundation of Tianjin under Grant 23JCQNJC01930, and in part by the Key Technologies R & D Program of Tianjin under Grant 23YFZCSN00060. *(Corresponding author: Xiao Liang.)*

The authors are with the Institute of Robotics and Automatic Information System, College of Artificial Intelligence, and Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China, also with the the Engineering Research Center of Trusted Behavior Intelligence, Ministry of Education, Nankai University, Tianjin 300350, China, and also with the Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen 518083, China (e-mail: zhangzp@mail.nankai.edu.cn; yuhai@mail.nankai.edu.cn; chaiyi@nankai.edu.cn; yangzc@mail.nankai.edu.cn; liangx@nankai.edu.cn; fangyc@nankai.edu.cn; hanjianda@nankai.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2025.3550562.

Digital Object Identifier 10.1109/TMECH.2025.3550562

manipulator and the hinged door. Tognon et al. [8] used rapidly-exploring random tree (RRT) algorithm as the global planner and proposed the control aware planning method to maintain contact with the surface. Bodie et al. [9] presented an active interaction force control and planning method for omnidirectional aerial manipulation platforms, with the goal of aerial contact inspection in unstructured environments. With the purpose of environmental health monitoring or the collection of fruits by vibration, González-Morgado et al. [10] realized the controlled shaking of flexible systems with the aerial manipulator.

In recent years, research on aerial manipulators has primarily focused on the motion control issue. However, in order to enhance the capability of performing tasks for the aerial manipulator in a complex environment, motion planning is of vital necessity. The aim of motion planning is to generate a collision-free and dynamically feasible trajectory, which is the core of this article.

Recently, several methods have been developed to address the motion planning problem for various types of robotic systems, including multirotors and mobile manipulators. The motion planning problem of multirotors is usually established by planning a mass point [11], [12], [13] enclosed within a sphere in the space with the differential flatness property [14]. Besides, in some other works, the multirotor can be enclosed within an ellipsoid [15] or a convex polyhedron [16] to reduce the conservatism of the trajectories. The authors in [17] and [18] utilized the MPC-based methods to achieve the coupled, dynamically feasible and safe motion planning for ground mobile manipulators. Wu et al. [19] proposed a decoupled path searching method and realize trajectory optimization for the entire mobile manipulator system. However, the aforementioned methods are difficult to be applied to aerial manipulators. Unlike multirotors, the motion planning problem of aerial manipulators aims to perform tasks with the end-effector, involving more degrees of freedom. In contrast to aerial manipulators, ground mobile manipulators do not face the challenge of platform tilting, and their system inputs are typically simpler, such as wheel and joint velocities. Therefore, the proposed method for multirotors or mobile manipulators cannot be directly applied to the aerial manipulator.

As shown in Fig. 1, the motion planning of the aerial manipulator could also be divided into two categories: Fig. 1(a) Decoupled motion planning designs the trajectory of the multirotor and the robotic arm in separate stages [6]. First, the multirotor's trajectory is planned to ensure it ends in a suitable position for manipulation, followed by planning the motion of the robotic arm. Fig. 1(b) Coupled motion planning designs the trajectory of both the multirotor and the robotic arm simultaneously.

Zhang et al. [20] drived the multirotor to the target area and then determines the trajectory of the robotic arm to grasp a target from a moving platform. In [21], the aerial manipulator approaches the target and subsequently executes the peg-in-hole task using the robotic arm. Cao et al. [22] proposed the decoupled motion planning framework, which could be divided into strategies for the approaching stage and the pick-and-place stage. However, the decoupling motion between the aerial platform and the robotic arm may not be optimal temporally or

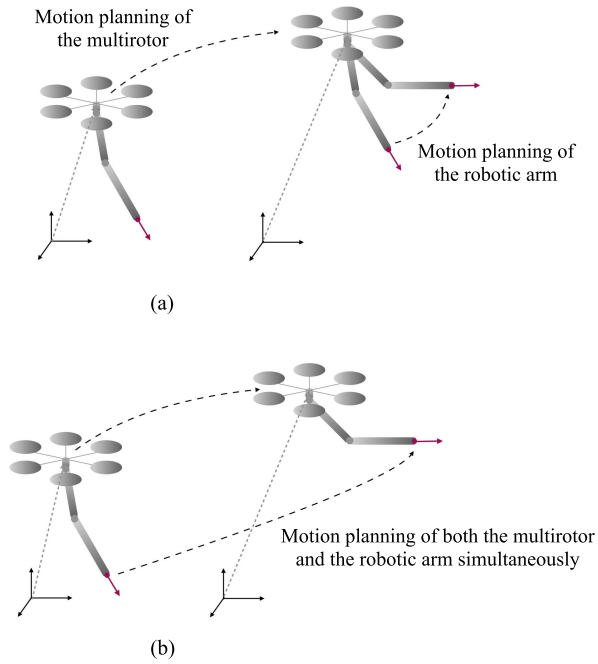

Fig. 1. Two categories of the motion planning of the aerial manipulation. (a) Decoupled motion planning. (b) Coupled motion planning.

spatially [20], [21], [22]. Besides, due to the decoupled motion planning framework, the aerial manipulator is considered as a big sphere in the safety constraint [22], which brings a more conservative trajectory.

In contrast to the decoupled method, the motion planning for the multirotor and the robotic arm is conducted simultaneously in the coupled method. Thomas et al. [23] directly developed a dynamic model of the aerial manipulator in the image plane and propose the trajectory optimization method. With the establishment of the aerial manipulator's discretized Euler–Lagrange model, Luo et al. [24] proposed a time-optimal trajectory planning method to pick up a parcel from a moving mobile robot. Different from the discretization of the dynamic model, Welde and Kumar [25] proposed the coordinate-free dynamic modeling method for a special class of 6-degrees of freedom (DoF) aerial manipulator and analyze its differential flatness. Furthermore, they extend the analysis of differential flatness to any configuration of the aerial manipulator systems and propose the task planning method in [26]. The collision-free problem is not the focus of the aforementioned works [23], [24], [25], [26], which mainly discuss the planning problem in open-space environments. However, it plays a critical role in the real-world applications of aerial manipulators. When aerial manipulators move to the operating location and perform the tasks, it is necessary to consider the potential collisions with the surrounding environment. The locally optimal motion planning method is proposed for aerial manipulators in [27]. Then, Kim et al. [28] utilized informed-RRT$*$ as a global planner, and combine it with the method in [27] as the local planner to generate the collision-free trajectory for the aerial manipulator in complex surrounding environment. However, the method proposed in

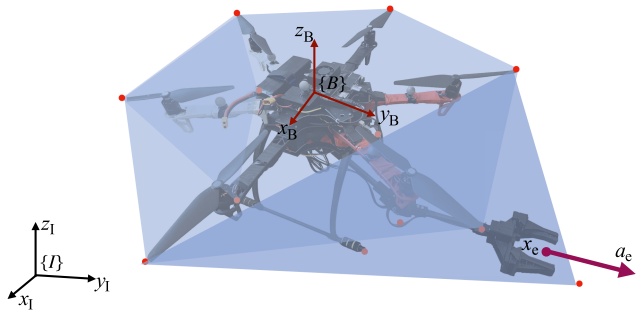

Fig. 2. Aerial manipulator in dynamic convex polyhedron.

[27] and [28] only considers the position of the end-effector while neglecting the orientation of the end-effector, which is crucial for practical tasks.

Taking all these factors into account, this article proposes a coupled trajectory planning method for the aerial manipulator in complex environments, resulting in an efficient, dynamically feasible, and collision-free trajectory for the aerial manipulator. The trajectory planning problem is formulated as a nonlinear optimization problem while considering the system dynamics and the safety constraints, in which the safety constraint is formulated by a dynamic convex polyhedron enclosing the aerial manipulator within the safe flight corridor. By setting constraints appropriately and solving the optimization problem, the desired trajectory can be obtained to ensure the success of the grasping task. The main contributions of this article are listed as follows.

1) The aerial manipulator is enclosed by a dynamic convex polyhedron instead of a large sphere, resulting in reduced conservatism. In some cases, the inherent conservatism of the large sphere can cause motion planning to fail.
2) In contrast to most of the current research, not only the desired position of the end-effector but also the desired orientation of the end-effector is taken into account in the proposed method. Actually, the consideration of the end-effector orientation in practical tasks is crucial in helping avoid collisions with the target object.

The rest of this article is organized as follows. In Section II, some preliminaries including the system dynamics, the differential flatness, and the problem statement are introduced. Then, Section III presents the motion planning method for the aerial manipulator. In Section IV, the algorithm implementation and the experimental results are provided. Finally, Section V concludes this article and presents the future work.

II. Preliminary and Problem Statement

A. Modeling of the Aerial Manipulator

The schematic configuration of the aerial manipulator system is shown in Fig. 2. The aerial manipulator is composed of a multirotor and a 2-DoF pitch–pitch robotic arm, with a total of 8 degrees of freedom. $\{I\} = \{\boldsymbol{x}_I, \boldsymbol{y}_I, \boldsymbol{z}_I\}$ represents the fixed inertia frame, and $\{B\} = \{\boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$ is the body-fixed frame. The symbols and the definitions are listed in the Nomenclature.

The state variable of the aerial manipulator is defined as follows:

$$\boldsymbol{q} = \begin{Bmatrix} \boldsymbol{x} \\ R \\ \boldsymbol{\theta} \end{Bmatrix} \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^2. \tag{1}$$

The corresponding generalized velocity is defined as

$$\boldsymbol{\varphi} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{\omega} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \in \mathbb{R}^8. \tag{2}$$

The generalized control input is defined as

$$\boldsymbol{u} = \begin{bmatrix} f \\ \boldsymbol{\tau} \\ \boldsymbol{\tau}_\theta \end{bmatrix} \in \mathbb{R}^6. \tag{3}$$

The dynamic model of the multirotor with the effects generated by the robotic arm is described as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{\nu} \tag{4a}$$

$$m\dot{\boldsymbol{\nu}} = fR\boldsymbol{e}_3 - mg\boldsymbol{e}_3 + \boldsymbol{f}_d \tag{4b}$$

$$\dot{R} = R\mathrm{sk}(\boldsymbol{\omega}) \tag{4c}$$

$$J\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{\tau}_d \tag{4d}$$

where $\boldsymbol{e}_3 = [0, 0, 1]^\top$ is the unit vector, $\mathrm{sk}(\cdot)$ maps $\mathbb{R}^3 \to so(3)$, and $\boldsymbol{f}_d$ and $\boldsymbol{\tau}_d$ are the force and torque effects generated by the robotic arm, which can be expressed as follows:

$$\boldsymbol{f}_d = -mR\left[\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{bc}) + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_{bc} + 2\boldsymbol{\omega} \times \dot{\boldsymbol{r}}_{bc} + \ddot{\boldsymbol{r}}_{bc}\right] \tag{5}$$

$$\boldsymbol{\tau}_d = -m\left[\boldsymbol{r}_{bc} \times R^\top(g\boldsymbol{e}_3 + \dot{\boldsymbol{\nu}})\right] - \boldsymbol{\omega} \times (J_r\boldsymbol{\omega}) - \dot{J}_r\boldsymbol{\omega} - J_r\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{L}_r - \dot{\boldsymbol{L}}_r \tag{6}$$

where $\boldsymbol{r}_{bc} \in \mathbb{R}^3$ represents the position vector of the mass center of the aerial manipulator with respect to $\{B\}$. $\boldsymbol{L}_r \in \mathbb{R}^3$ is the angular momentum of the robotic arm with respect to $\{B\}$, $J_r \in \mathbb{R}^{3\times3}$ is the inertia matrix of the robotic arm referenced to the mass center of the multirotor $\boldsymbol{x}$ with respect to $\{B\}$. $\dot{\boldsymbol{L}}_r$ and $\dot{J}_r$ are their time derivatives, respectively.

As shown in Fig. 2, the position and the direction vector of the end-effector can be expressed as follows:

$$\boldsymbol{t}_e = \left\{(\boldsymbol{x}_e, \boldsymbol{a}_e) | \boldsymbol{x}_e \in \mathbb{R}^3, \boldsymbol{a}_e \in \mathbb{S}^2\right\} \tag{7}$$

where $\boldsymbol{x}_e$ is the position vector of the end-effector, $\boldsymbol{a}_e$ is the direction vector of the end-effector with respect to the inertia frame $\{I\}$. Both of them are functions of the state variable $\boldsymbol{q}$ of the aerial manipulator, which can be expressed as

$$\boldsymbol{x}_e = \boldsymbol{x}_e(\boldsymbol{x}, R, \boldsymbol{\theta}) \tag{8}$$

$$\boldsymbol{a}_e = \boldsymbol{a}_e(R, \boldsymbol{\theta}). \tag{9}$$

Considering that the 2-DoF pitch-pitch robotic arm is mounted on the multirotor, the pose of the end-effector belongs to the $\mathbb{R}^3 \times \mathbb{S}^2$ space.

B. Differential Flatness of the Aerial Manipulator

According to the dynamic model of the multirotor with the generalized force effects generated by the robotic arm (4), the force and torque acting on the multirotor due to the motion of the robotic arm are $\boldsymbol{f}_d$ and $\boldsymbol{\tau}_d$, whose explicit form are expressed as (5) and (6). The generalized forces are related to the mass, the joint velocity and acceleration of the robotic arm. Since the mass of the robotic arm is much smaller than that of the multirotor, and the motion of the robotic arm is prevented from being excessive by limiting its maximum velocity and acceleration. Thus, $\boldsymbol{f}_d$ and $\boldsymbol{\tau}_d$ become sufficiently small to be negligible for the multirotor. Moreover, the robust controller is utilized for the multirotor, which ensures that the effects exerted by the robotic arm are effectively rejected. The generalized forces acting on the robotic arm are equal in magnitude and opposite in direction to the generalized forces acting on the base due to the arm. Due to the high gear ratio servos typically used in robotic arms of the aerial manipulators, the actuators can provide sufficient torque to reduce the dynamic coupling from the multirotor.

Based on the above analysis, the dynamic coupling effects between the multirotor and the robotic arm can be neglected, allowing their states and control inputs to be independently expressed. Specifically, the states and control inputs of the multirotor can be expressed by the functions of $\boldsymbol{x}$, ψ and their derivatives [14], while the states and the control inputs of the robotic arm can be expressed by $\boldsymbol{\theta}$ and its derivatives [29]. Thus, the flat outputs of the aerial manipulator are chosen as $\boldsymbol{y} = [\boldsymbol{x}^\top, \psi, \boldsymbol{\theta}^\top]^\top$. The position and the orientation of the end-effector $\boldsymbol{x}_e$ and $\boldsymbol{a}_e$ can also be expressed as the functions of the flat outputs

$$\boldsymbol{x}_e = \boldsymbol{x}_e(\boldsymbol{x}, \ddot{\boldsymbol{x}}, \psi, \boldsymbol{\theta}) \tag{10}$$

$$\boldsymbol{a}_e = \boldsymbol{a}_e(\ddot{\boldsymbol{x}}, \psi, \boldsymbol{\theta}). \tag{11}$$

Subsequently, all variables in the following trajectory optimization problem can be expressed in terms of the derived flat outputs $\boldsymbol{y}$.

C. Problem Statement

This article aims to find a safe trajectory for the aerial manipulator. The problem could be formulated as follows.

Give the initial state $\boldsymbol{q}_0 = \{\boldsymbol{x}_0, R_0, \boldsymbol{\theta}_0\}$ of the aerial manipulator and the goal pose of the end-effector $\boldsymbol{t}_{ed} = \{\boldsymbol{x}_{ed}, \boldsymbol{a}_{ed}\}$, plan a dynamically feasible and safe trajectory $\boldsymbol{q}(t), 0 \le t \le T$ for the aerial manipulator to reach the goal position $\boldsymbol{t}_e(T) = \boldsymbol{t}_{ed}$.

III. Main Method

In this section, the end-effector-oriented coupled planning method for the aerial manipulator is introduced. The framework is shown in Fig. 3.

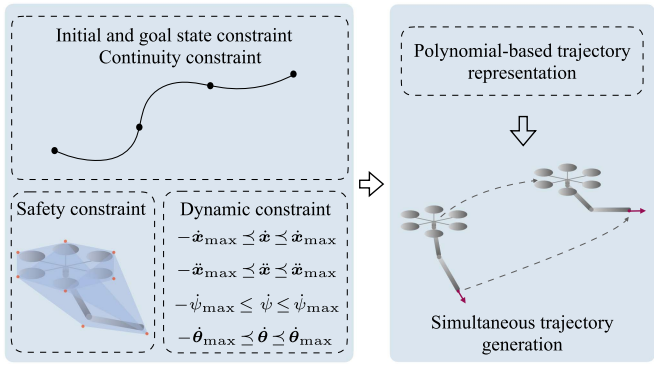

Fig. 3. Trajectory optimization framework.

A. Dynamic Convex Polyhedron Setting

It is assumed that the aerial manipulator is enclosed in a convex polyhedron as shown in Fig. 2. Unlike the polyhedron that encloses a multirotor [16], where all vertices remain fixed with respect to the coordinate frame of the multirotor $\{B\}$, the convex polyhedron enclosing the aerial manipulator dynamically changes with the motion of the manipulator. All vertices of the polyhedron are denoted as $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_l\}$, where $\boldsymbol{v}_k (k = 1, 2, \ldots, l) \in \mathbb{R}^3$ and l is the total number of vertices of the polyhedron. All vertices of the dynamic polyhedron are indicated by the red dots in Fig. 2. The polyhedron enclosing the aerial manipulator is denoted as $P(V)$.

The passable area of the environment is described by the safe flight corridor, a collection of convex overlapping polyhedra. The ith convex polyhedron is defined as follows:

$$\mathbb{C}_i = \{\boldsymbol{v} \in \mathbb{R}^3 | \boldsymbol{A}_i \boldsymbol{v} \preceq \boldsymbol{b}_i\} \, (i = 1, 2, \ldots, n) \tag{12}$$

where n denotes the number of the convex polyhedra representing the safe flight corridor. The symbol $\preceq$ denotes the element-wise inequality. For two vectors $\boldsymbol{a} = [a_1, a_2, a_3]^\top$ and $\boldsymbol{b} = [b_1, b_2, b_3]^\top$, $\boldsymbol{a} \preceq \boldsymbol{b}$ is equivalent to $a_1 \le b_1$, $a_2 \le b_2$ and $a_3 \le b_3$. The entire safe flight corridor is defined as

$$\mathbb{C} = \bigcup_{i=1}^{n} \mathbb{C}_i. \tag{13}$$

The safety constraint indicates that the polyhedron enclosing the aerial manipulator must be within the safe flight corridor, which is defined as

$$P(V) \subseteq \mathbb{C}. \tag{14}$$

B. Dynamic Constraint of Aerial Manipulator

Because of the actuator limitation, the dynamic constraint of the aerial manipulator should be considered.

1) Dynamic Constraint of Multirotor: The maximum values of the velocity $\dot{\boldsymbol{x}}_{\max}$ and the acceleration $\ddot{\boldsymbol{x}}_{\max}$ of the multirotor should be set. The dynamic constraint of the multirotor is defined as

$$-\dot{\boldsymbol{x}}_{\max} \preceq \dot{\boldsymbol{x}} \preceq \dot{\boldsymbol{x}}_{\max} \tag{15a}$$

$$-\ddot{\boldsymbol{x}}_{\max} \preceq \ddot{\boldsymbol{x}} \preceq \ddot{\boldsymbol{x}}_{\max}. \tag{15b}$$

The yaw rate of the multirotor should also be limited as

$$-\dot{\psi}_{\max} \leq \dot{\psi} \leq \dot{\psi}_{\max}. \tag{16}$$

2) Dynamic Constraint of the Robotic Arm: The maximum values of the joint angle rates $\dot{\boldsymbol{\theta}}_{\max}$ is set as follows:

$$-\dot{\boldsymbol{\theta}}_{\max} \preceq \dot{\boldsymbol{\theta}} \preceq \dot{\boldsymbol{\theta}}_{\max}. \tag{17}$$

C. Piece-Wise Polynomial-Based Trajectory Optimization

The trajectory of the aerial manipulator is divided into n segments by $n-1 (n \geq 1)$ waypoints, which correspond to n convex polyhedra of the safe flight corridor. The ith segment trajectory with the time duration T_i is defined as follows:

$$\boldsymbol{x}_i = \sum_{j=0}^{s_x} \boldsymbol{c}_{j,i} t^j \in \mathbb{R}^3, t \in [0, T_i] \tag{18a}$$

$$\psi_i = \sum_{j=0}^{s_\psi} a_{j,i} t^j \in \mathbb{R}, t \in [0, T_i] \tag{18b}$$

$$\boldsymbol{\theta}_i = \sum_{j=0}^{s_\theta} \boldsymbol{b}_{j,i} t^j \in \mathbb{R}^2, t \in [0, T_i] \tag{18c}$$

where s_x, s_ψ, and s_θ denote the orders of the trajectories. $\boldsymbol{c}_{j,i}$, $\boldsymbol{a}_{j,i}$, and $\boldsymbol{b}_{j,i}$ are the coefficients of the corresponding polynomial trajectories given as follows:

$$\boldsymbol{c}_i = \begin{bmatrix} \boldsymbol{c}_{0,i} & \boldsymbol{c}_{1,i} & \cdots & \boldsymbol{c}_{s_x,i} \end{bmatrix} \in \mathbb{R}^{3\times(s_x+1)}$$

$$\boldsymbol{a}_i = \begin{bmatrix} a_{0,i} & a_{1,i} & \cdots & a_{s_\psi,i} \end{bmatrix} \in \mathbb{R}^{1\times(s_\psi+1)}$$

$$\boldsymbol{b}_i = \begin{bmatrix} \boldsymbol{b}_{0,i} & \boldsymbol{b}_{1,i} & \cdots & \boldsymbol{b}_{s_\theta,i} \end{bmatrix} \in \mathbb{R}^{2\times(s_\theta+1)}.$$

When n polynomials are considered, the trajectory of the aerial manipulator is a concatenation of the n segments. The coefficients of the trajectory are denoted as $\boldsymbol{C}$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{T}$, which are defined as follows:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n \end{bmatrix} \in \mathbb{R}^{3\times n(s_x+1)}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \end{bmatrix} \in \mathbb{R}^{1\times n(s_\psi+1)}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_n \end{bmatrix} \in \mathbb{R}^{2\times n(s_\theta+1)}$$

$$\boldsymbol{T} = \begin{bmatrix} T_1 & T_2 & \cdots & T_n \end{bmatrix} \in \mathbb{R}^{n}.$$

For instance, the trajectory of the position vector $\boldsymbol{x}(t)$ is a concatenation of (18a), which is defined as follows:

$$\boldsymbol{x}(t) = \begin{cases} \boldsymbol{x}_1(t),\ T_0 \leq t < T_1 \\ \boldsymbol{x}_2(t-T_1),\ T_1 \leq t < T_1 + T_2 \\ \vdots \\ \boldsymbol{x}_n(t-(T_1+\cdots+T_{n-1})) \\ \quad T_1+\cdots+T_{n-1} \leq t < T_1+\cdots+T_n \end{cases} \tag{19}$$

where $T_0 = 0, T_n = T$. With the description of the polynomial-based trajectory in (18), the trajectory optimization problem could be formulated as follows:

$$\min_{\boldsymbol{C},\boldsymbol{A},\boldsymbol{B},\boldsymbol{T}} \int_0^T \left(\lambda_1 \|\boldsymbol{x}^{(4)}\|^2 + \lambda_2 \ddot{\psi}^2 + \lambda_3 \|\ddot{\boldsymbol{\theta}}\|^2\right) d\mathrm{t} + \lambda_4 T \tag{20a}$$

$$\text{s.t.} \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,\ \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0,\ \psi(0) = \psi_0 \tag{20b}$$

$$\dot{\boldsymbol{x}}(0) = \ddot{\boldsymbol{x}}(0) = \boldsymbol{0},\ \dot{\boldsymbol{\theta}}(0) = \boldsymbol{0}, \dot{\psi}(0) = 0 \tag{20c}$$

$$\boldsymbol{x}_e(T) = \boldsymbol{x}_{ed},\ \boldsymbol{a}_e(T) = \boldsymbol{a}_{ed} \tag{20d}$$

$$\dot{\boldsymbol{x}}(T) = \boldsymbol{0},\ \dot{\boldsymbol{\psi}}(T) = 0,\ \dot{\boldsymbol{\theta}}(T) = \boldsymbol{0} \tag{20e}$$

$$\boldsymbol{A}_i \boldsymbol{v} \preceq \boldsymbol{b}_i \quad \forall \boldsymbol{v} \in V \tag{20f}$$

$$-\dot{\psi}_{\max} \leq \dot{\psi} \leq \dot{\psi}_{\max},\ -\dot{\boldsymbol{\theta}}_{\max} \preceq \dot{\boldsymbol{\theta}} \preceq \dot{\boldsymbol{\theta}}_{\max} \tag{20g}$$

$$-\dot{\boldsymbol{x}}_{\max} \preceq \dot{\boldsymbol{x}} \preceq \dot{\boldsymbol{x}}_{\max},\ -\ddot{\boldsymbol{x}}_{\max} \preceq \ddot{\boldsymbol{x}} \preceq \ddot{\boldsymbol{x}}_{\max} \tag{20h}$$

$$\boldsymbol{x}_i(T_i) = \boldsymbol{x}_{i+1}(0) = \overline{\boldsymbol{x}}_i,\ \boldsymbol{x}_i^{(d)}(T_i) = \boldsymbol{x}_{i+1}^{(d)}(0) \tag{20i}$$

$$\psi_i(T_i) = \psi_{i+1}(0),\ \dot{\psi}_i(T_i) = \dot{\psi}_{i+1}(0) \tag{20j}$$

$$\boldsymbol{\theta}_i(T_i) = \boldsymbol{\theta}_{i+1}(0),\ \dot{\boldsymbol{\theta}}_i(T_i) = \dot{\boldsymbol{\theta}}_{i+1}(0) \tag{20k}$$

where the constraints in (20d) can be replaced as follows:

$$\boldsymbol{a}_{ed1} \preceq \boldsymbol{a}_e(T) \preceq \boldsymbol{a}_{ed2}. \tag{21}$$

Here, λ_1, λ_2, λ_3, and λ_4 are the weighting factors of the optimization problem. $(\cdot)^{(d)} (d = 1, 2, 3)$ denotes the dth order derivative of the variable. $\overline{\boldsymbol{x}}_i$ is the ith waypoint of the aerial manipulator, which is generated by the 3-D jump point search algorithm [13], where $i = 1, 2, \ldots, n-1$. In (20a), the optimization problem aims to minimize both the actuation cost and the time cost. According to the differential flatness property, the terms $\|\boldsymbol{x}^{(4)}\|^2$ and $\ddot{\psi}^2$ represent the actuation cost of the multirotor, while $\|\ddot{\boldsymbol{\theta}}\|^2$ reflects the actuation cost of the robotic arm, and T represents the time cost. The constraints in (20b)–(20e) are the initial and the desired states of the aerial manipulator. The desired orientation of the end-effector can be configured as either a fixed value or within a specified range, i.e., one can also replace the constraint as (21). The safety constraint in (20f) ensures that the polyhedron enclosing the aerial manipulator are within the safe flight corridor. The constraints in (20g) and (20h) are the dynamic limitations of the multirotor and the robotic arm, because of the parameterization of the trajectory, the velocity and acceleration of the aerial manipulator can be obtained by differentiating the trajectory. The dynamic constraints in (20g) and (20h) can be applied by sampling method. Due the trajectory's continuity, the dynamic constraints are satisfied in the whole trajectory. The continuity constraints in (20i)–(20k) ensure the continuity of the trajectory at the waypoints. By solving the aforementioned optimization problem (20), the collision-free and efficient trajectory of the aerial manipulator can be obtained.

The constraints (20b)–(20e) and (20i)–(20k) are the equality constraints. As long as the orders of the polynomials are sufficient, the continuity constraints can be satisfied. The safety constraint (20f) and the dynamic constraints (20h) and (20g) are the inequality constraints. If the distribution of obstacles is not overly dense, the generated safe flight corridor is large enough to contain the dynamic convex polyhedron, as shown in Fig. 2, the

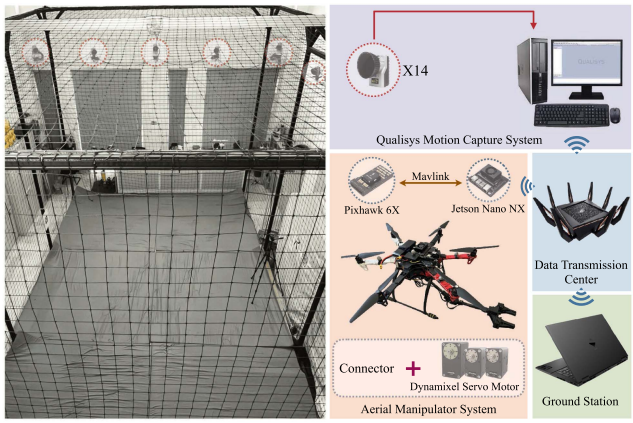

Fig. 4. Experimental platform.

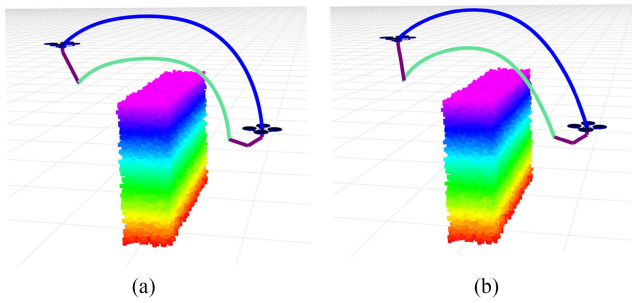

Fig. 5. Trajectories generated in *Scenario 1*. The blue curves represent the trajectory of the multirotor, and the green curves represent the trajectory of the end-effector. Two purple cylinders below the multirotor represent the 2-DoF robotic arm. The colored cube in the figure represents the obstacle in the environment. (a) Enclosed by the sphere. (b) Enclosed by the convex polyhedron.

safety constraint can be satisfied. In addition, when the trajectory of the aerial manipulators is smooth, with low velocities and accelerations, the dynamic constraints (20h) and (20g) can also be met.

In some cases that the constraints in the optimization problem result in infeasibility, the number of waypoints on the generated path can be added, which will increase the degrees of freedom for the optimization problem (20a). In addition, the constraint on $\boldsymbol{a}_e(T)$ in (20d) can be substituted with the one in (21), which are relaxed to enhance the feasibility of the optimization problem.

IV. Algorithm Implementation and Experimental Verification

In this section, the algorithm implementation and experimental results will be introduced. The algorithm implementation platform and the experiment setup will be introduced in Section IV-A. The experimental results are presented in Section IV-B.

A. Algorithm Implementation and Experiment Setup

The main algorithm is implemented with Python3. The optimization problem (20) in Section III-C is solved by CasADi [30]. The safe flight corridor is generated by the algorithm proposed in [13]. The whole algorithm is implemented in the Robot Operating System (ROS) environment. The code is released online.[1] As shown in Fig. 4, the experiment platform contains the aerial manipulator system, the Qualisys motion capture system, the ground station, and the data transmission center. The aerial manipulator system is composed of an F550 hexrotor and a self-built robotic arm. The mass of the multirotor is 2.58 kg, and the mass of the robotic arm is 0.35 kg, respectively. Specifically, the F550 hexrotor is equipped with a Pixhawk 6X flight controller connecting to the onboard computer through Mavlink-based communication protocol. The flight controller uses the cascaded proportional-proportional integral derivative (P-PID) controller with the feedforward term to compensate for the effect brought by the robotic arm. The robotic arm consists of one Dynamixel XM540 servo motor (gear ratio 272.5:1), two Dynamixel XM430 servo motors (gear ratio 353.5:1), and several 3D-printed connectors. The Qualisys motion capture system is built with 14 cameras and a ground station, which is utilized to measure the hexrotor's motion. The ground station runs the 64-bit Ubuntu 20.04 operating system, which is employed to send the desired trajectory of the aerial manipulator with the support of ROS. The data transmission center is supported by a router, which is responsible for exchanging all data in the experiment process.

[1][Online]. Available: https://github.com/cheungsiupaang/Safe-Planner

B. Experimental Results

The orders of the polynomial trajectories s_x, s_ψ and s_θ in (18) are chosen as 5, 3, and 3, respectively. The velocity and acceleration constraints of the multirotor are set as $\dot{\boldsymbol{x}}_{\max} = [1.5,\ 1.5,\ 1.5]^\top \mathrm{m/s}$, $\ddot{\boldsymbol{x}}_{\max} = [1.0,\ 1.0,\ 1.0]^\top \mathrm{m/s^2}$, and $\dot{\psi}_{\max} = 0.1 \mathrm{rad/s}$. The joint angle rates constraints of the robotic arm are set as $\dot{\boldsymbol{\theta}}_{\max} = [0.2,\ 0.2]^\top \mathrm{rad/s}$, in which the limitations on the joint angle rates are stringent, preventing the robotic arm from reaching excessively high speed.

To verify the adaptability of the proposed method and the necessity of constraining the end-effector's desired orientation, two sets of scenarios are conducted. Both *Scenario 1* and *Scenario 2* share the same dynamic constraint parameters.

1) Scenario 1: In this scenario, the initial states of the aerial manipulator are set as $\boldsymbol{x}_0 = [-1.0,\ 0.0,\ 0.6]^\top \mathrm{m}$, $\boldsymbol{\theta}_0 = [0.6,\ -0.9]^\top \mathrm{rad}$, and $\psi_0 = 0.0\,\mathrm{rad}$. The desired position of the end-effector is set as $\boldsymbol{x}_{ed} = [-1.6,\ -2.0,\ 1.1]^\top \mathrm{m}$. This scenario is to test the performance of the proposed method in the rest-to-rest task and the desired orientation of the end-effector $\boldsymbol{a}_e$ is not set.

The aerial manipulator enclosed by a big sphere for the safety constraint is chosen as the baseline method for comparison. Fig. 5 shows the trajectories generated by the sphere and the convex polyhedron envelopes, respectively. The quantitative results are shown in Table I, demonstrating that the proposed method has a shorter arrival time and a reduced multirotor path length, without significantly increasing the computational resource. This is because the proposed method plans a more aggressive trajectory based on the dynamic convex polyhedron

TABLE I
QUANTITATIVE RESULTS IN *SCENARIO 1*

Method	Arrival Time (s)	Multirotor Path Length (m)	Compute Time (s)
proposed	**14.67**	**3.60**	1.16
baseline	17.38	3.83	**0.86**

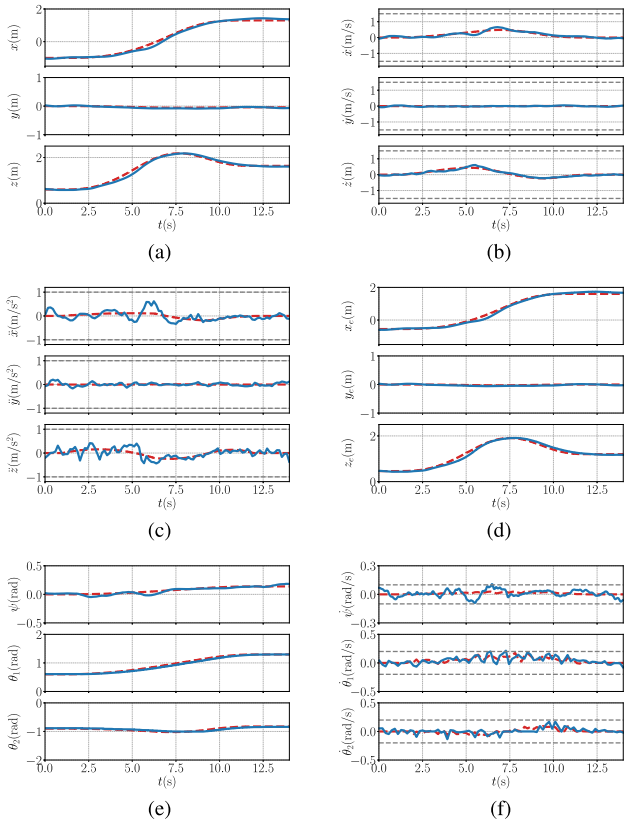

Fig. 6. History of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$, $\boldsymbol{x}_e$, ψ, $\boldsymbol{\theta}$, $\dot{\psi}$, and $\dot{\boldsymbol{\theta}}$ in the *Scenario 1*. The blue curves represent the actual trajectories, and the dashed red curves represent the planned trajectories. The dashed gray curves represent the respective limitations of the optimization problem. (a) Position of the multirotor. (b) Velocity of the multirotor. (c) Acceleration of the multirotor. (d) Position of the end-effector. (e) Yaw angle and the joint angles. (f) Rates of yaw and joint angles.

setting. The experimental results with the aerial manipulator enclosed by the convex polyhedron are presented in Fig. 6, including the curves of the multirotor position in Fig. 6(a), velocity in Fig. 6(b) and acceleration in Fig. 6(c), respectively, as well as the curves of the end-effector position in Fig. 6(d). The curves of the yaw angle and the joint angles, and the rates of the yaw angle and the joint angles are shown in Fig. 6(e) and (f), respectively.

2) Scenario 2-A: In this scenario, the initial states of the aerial manipulator are set as $\boldsymbol{x}_0 = [1.2,\ 2.4,\ 0.6]^\top \mathrm{m}$, $\boldsymbol{\theta}_0 = [0.6,\ -0.9]^\top \mathrm{rad}$, and $\psi_0 = -1.57\,\mathrm{rad}$. The desired states of the end-effector are set as $\boldsymbol{x}_{ed} = [-1.6,\ -2.0,\ 1.1]^\top \mathrm{m}$, $\boldsymbol{a}_{ed} = [0.0,\ -1.0,\ 0.0]^\top$. Compared to the previous scenario, this scenario is to validate the performance of the proposed method in the grasp task, which is dependent on both the position and the orientation of the end-effector.

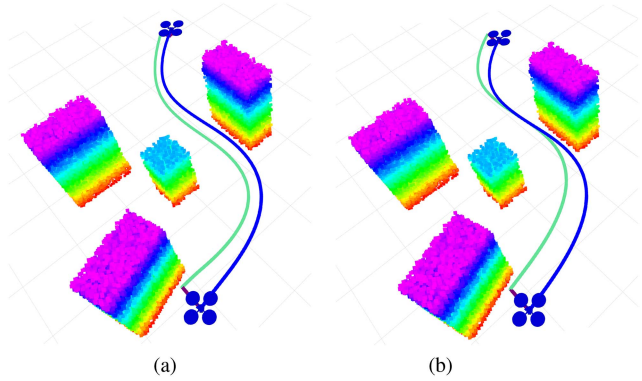

Fig. 7. Trajectories generated in *Scenario 2-A*. The blue curves represent the trajectory of the multirotor, and the green curves represent the trajectory of the end-effector. Two purple cylinders below the multirotor represent the 2-DoF robotic arm. The colored cube in the figure represents the obstacle in the environment. (a) Enclosed by the sphere. (b) Enclosed by the convex polyhedron.

TABLE II
QUANTITATIVE RESULTS IN *SCENARIO 2-A*

Method	Arrival Time (s)	Multirotor Path Length (m)	Compute Time (s)
proposed	**18.55**	**5.65**	2.99
baseline	19.15	5.78	**2.62**

Fig. 7 shows the trajectories generated by the sphere and the convex polyhedron envelopes, respectively. The quantitative results are shown in Table II. Similar to *Scenario 1*, one can also conclude that the proposed method has a reduced arrival time and a shorter multirotor path length, without significant cost in computational resource usage.

The experimental results with the aerial manipulator enclosed by the convex polyhedron are presented in Fig. 8.

In this scenario, because the desired orientation of the end-effector is specified as a naive direction, i.e., $\boldsymbol{a}_{ed} = [0.0,\ -1.0,\ 0.0]^\top$, the end-effector collides with the target object, which is illustrated in the red dashed circle in Fig. 10(a).

3) Scenario 2-B: In order to avoid the collision, another well-considered selection of the desired orientation of the end-effector is $\boldsymbol{a}_{ed} = [a_{edx},\ a_{edy},\ a_{edz}]^\top$, where $-0.6 \leq a_{edx} \leq -0.4$, $-0.9 \leq a_{edy} \leq -0.8$, $a_{edz} = 0.0$. The experimental results are presented in Fig. 9. All the planned trajectories represented by the dashed red curves in Figs. 6, 8, and 9 are within the limitations of the optimization problem (20). It should be noted that due to the insufficient control accuracy of the practical aerial manipulator system, the actual yaw rate curve exceeded the set limits, as shown in Figs. 8(f) and 9(f). Fortunately, according to the final experimental results, the slight deviation in yaw rate tracking has a negligible impact on the system's task execution.

When the desired orientation of the end-effector is set as the well-considered situation, the aerial manipulator can smoothly grasp the target object without any collision as shown in Fig.

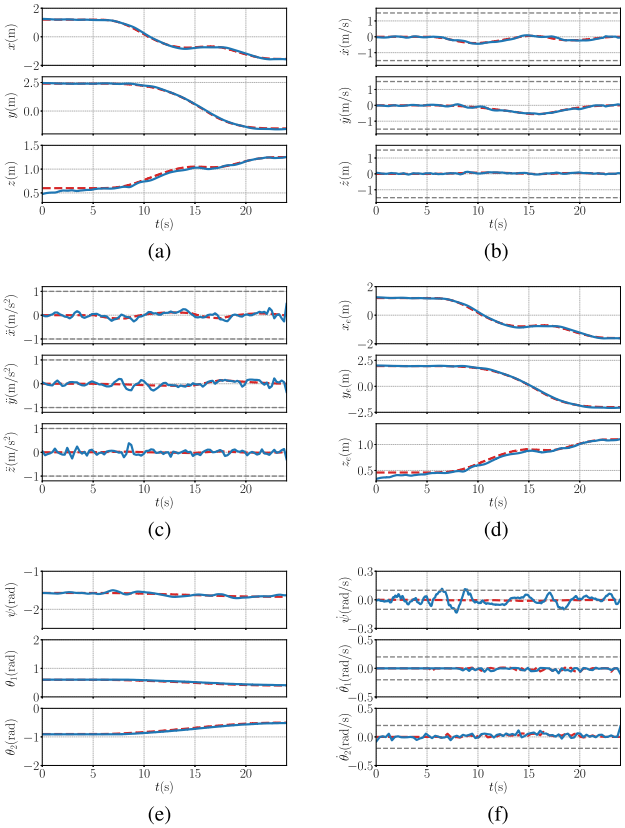

Fig. 8. History of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$, $\boldsymbol{x}_e$, ψ, $\boldsymbol{\theta}$, $\dot{\psi}$, and $\dot{\boldsymbol{\theta}}$ with the naive selection of the end-effector orientation in *Scenario 2-A*. The blue curves represent the actual trajectories, and the dashed red curves represent the planned trajectories. The dashed gray curves represent the respective limitations of the optimization problem. (a) Position of the multirotor. (b) Velocity of the multirotor. (c) Acceleration of the multirotor. (d) The Position of the end-effector. (e) Yaw angle and the joint angles. (f) Rates of yaw and joint angles.

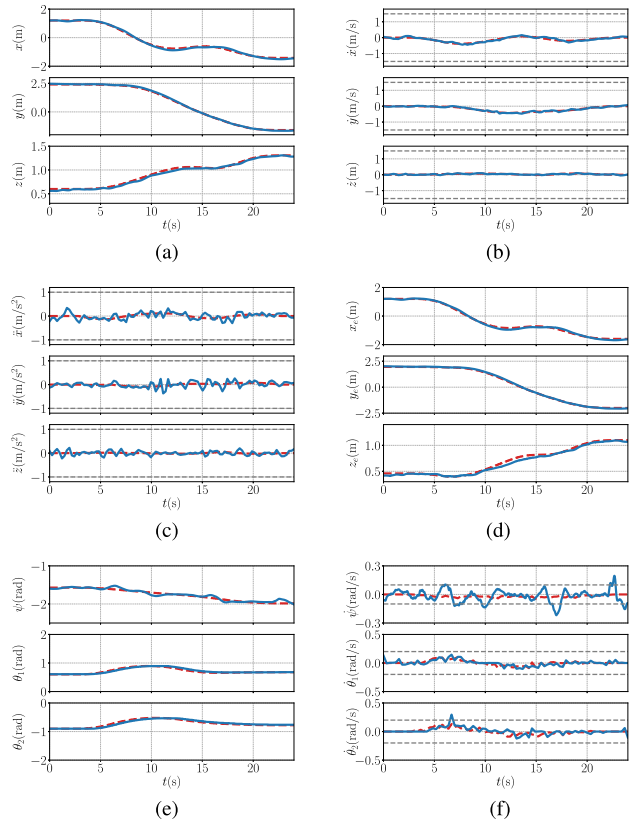

Fig. 9. History of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$, $\boldsymbol{x}_e$, ψ, $\boldsymbol{\theta}$, $\dot{\psi}$, and $\dot{\boldsymbol{\theta}}$ with the well-considered selection of the end-effector orientation in *Scenario 2-B*. The blue curves represent the actual trajectories, and the dashed red curves represent the planned trajectories. The dashed gray curves represent the respective limitations of the optimization problem. (a) Position of the multirotor. (b) Velocity of the multirotor. (c) Acceleration of the multirotor. (d) Position of the end-effector. (e) Yaw angle and the joint angles. (f) Rates of yaw and joint angles.

10(b). The top views of the trajectories generated in Fig. 10 are shown in Fig. 11. The blue curves represent the trajectories of the multirotor, and the green curves represent the trajectories of the end-effector. The green arrows represent the desired position and orientation of the end-effector, and the red dashed ellipse highlights the area where the collision between the end-effector (the gripper) and the target object occurred under the naive situation. The collision was caused by the misalignment between the end-effector's velocity direction $\dot{\boldsymbol{x}}_e$ and its desired orientation $\boldsymbol{a}_e$ in the red dashed area, which resulted in the end-effector moving along an unsuitable path.

4) Scenario 2-C: In this part, we conduct the ablation study on the weights of the optimization problem. The scenario is configured the same as *Scenario 2-A*, with initial weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ set as $(10, 10, 10, 1)$. In Table III, C_x denotes the integral of $\|\boldsymbol{x}^{(4)}\|^2$ over time, i.e., $C_x = \int_0^T \|\boldsymbol{x}^{(4)}\|^2 \mathrm{d}t$. Similarly, C_ψ and C_θ are defined as the time integrals of $\ddot{\psi}^2$ and $\|\ddot{\boldsymbol{\theta}}\|^2$, respectively. The ablation study results are summarized in Table III. For the ablation of λ_1, the weights are adjusted to $(0, 10, 10, 1)$, resulting in an increase in the cost C_x from 2.05×10^{-2} to 2.64×10^{-2}. Similarly, the ablation of λ_2 and

TABLE III
ABLATION STUDY RESULTS IN *SCENARIO 2-C*

$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	Cost Metrics			
	C_x $(\times 10^{-2})$	C_ψ $(\times 10^{-5})$	C_θ $(\times 10^{-5})$	T
$(10, 10, 10, 1)$	2.05	1.06	20.58	18.55
$(\ 0, 10, 10, 1)$	2.64	1.06	6.07	18.54
$(10,\ 0, 10, 1)$	1.42	1.92	9.96	18.77
$(10, 10,\ 0, 1)$	1.42	1.02	1377.84	18.76
$(10, 10, 10, 0)$	optimization failure			

λ_3 also leads to significant changes in the corresponding costs, as shown in Table III. It can be observed that changes in individual weights $(\lambda_1, \lambda_2, \lambda_3)$ primarily impact their corresponding costs (C_x, C_ψ, C_θ). Nevertheless, the coupling between different objectives in the optimization problem leads that adjusting one weight can also induce secondary effects on other cost terms indirectly.

For the ablation of λ_4, removing it causes optimization failure as the time cost is no longer penalized. Without this term, the solver focuses solely on minimizing actuation costs, ignoring time efficiency, which leads to solutions with increasingly

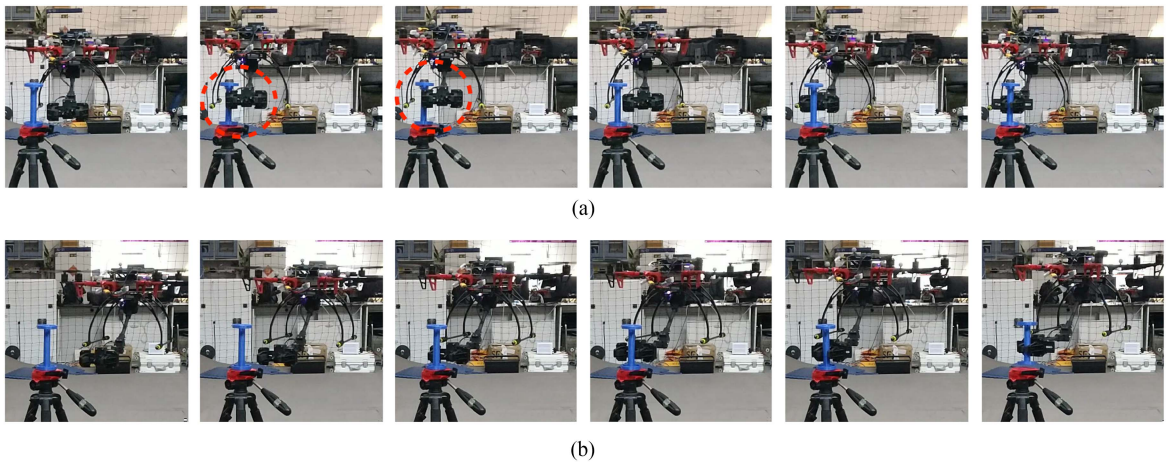

Fig. 10. Snapshots in *Scenario 2-A* and *Scenario 2-B*. Due to the naive selection of $\boldsymbol{a}_e$, the end-effector collides with the target object, which is shown in the two dashed red circles. With $\boldsymbol{a}_e$ reset in a well-considered range, the end-effector grasps the target object smoothly. (a) Naive desired orientation of the end-effector. (b) Well-considered desired orientation of the end-effector.

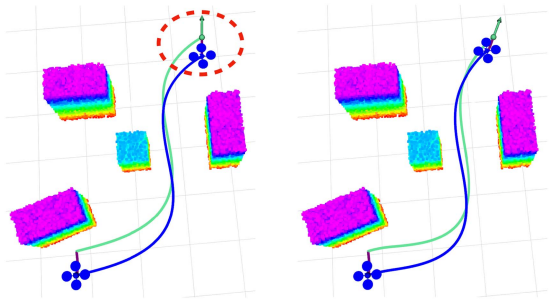

Fig. 11. Top views of generated trajectories. (a) Naive situation. (b) Well-considered situation.

TABLE IV
IMPACT OF λ_4 IN *SCENARIO 2-C*

λ_4	Cost Metrics			
	C_x	C_ψ	C_θ	T
10^{-1}	2.09×10^{-3}	6.60×10^{-6}	1.44×10^{-6}	21.77
10^{-2}	4.12×10^{-4}	4.06×10^{-6}	8.16×10^{-6}	25.55
10^{-3}	5.79×10^{-5}	1.46×10^{-6}	3.78×10^{-6}	33.60
10^{-4}	7.17×10^{-6}	7.37×10^{-8}	1.82×10^{-6}	45.58

larger trajectory times. This demonstrates the necessity of λ_4 in balancing performance and time efficiency. Table IV presents the impact of λ_4 on the cost metrics in *Scenario 2-C*, the other weights are set as $\lambda_1 = \lambda_2 = \lambda_3 = 10$. As λ_4 decreases, there is a significant trade-off between the time cost T and the actuation costs C_x, C_ψ, and C_θ. When $\lambda_4 = 10^{-1}$, the time cost T is the lowest at 21.77, but the actuation cost are relatively high. Reducing λ_4 to 10^{-4} minimizes the actuation costs, but it drastically increases the trajectory time increases to 45.58.

V. CONCLUSION

This work proposes an end-effector-oriented coupled motion planning technique for the aerial manipulator. The dynamic convex polyhedron is used to envelop the aerial manipulator to guarantee the safety constraint. The polynomial functions of the flat outputs could be used to depict the aerial manipulator's trajectory. The proposed method's efficiency is demonstrated by the outcomes of the simulation and experiment. Both the end-effector's desired position and the desired orientation are taken into account, which is ignored in most existing works. With consideration of the end-effector's orientation, the aerial manipulator could avoid the collision between the end-effector and the target object in the grasp task. The dynamic convex polyhedron approach can be generalized to other multirigid body robotic systems, such as mobile manipulators. By incorporating the kinematics and the geometric properties of multirigid body robotic systems, the dynamic convex polyhedron can describe the safety constraints associated with them.

While this article primarily focuses on a specific configuration of aerial manipulators, the proposed method is also applicable to other configurations of robotic arms, including redundant manipulators. By taking into account the kinematics and geometric features of the manipulators, the dynamic convex polyhedron and the end-effector's pose can be derived. Subsequently, the proposed method solves the optimization problem by integrating the derived dynamic convex polyhedron and the constraints on the end-effector's pose. The motion planning problem is then tackled within the same optimization framework. In the future, the real-time performance of the algorithm will be improved.

REFERENCES

[1] M. Chen, S. Xiong, and Q. Wu, "Tracking flight control of quadrotor based on disturbance observer," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 3, pp. 1414–1423, Mar. 2021.
[2] H. Yu, X. Liang, J. Han, and Y. Fang, "Adaptive trajectory tracking control for the quadrotor aerial transportation system landing a payload onto the

mobile platform," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 23–37, Jan. 2024.
[3] B. Xian and W. Hao, "Nonlinear robust fault-tolerant control of the tilt trirotor UAV under rear servo's stuck fault: Theory and experiments," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2158–2166, Apr. 2019.
[4] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 4916–4921.
[5] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4990–4995.
[6] 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.
[7] D. Lee, H. Seo, D. Kim, and H. J. Kim, "Aerial manipulation using model predictive control for opening a Hinged door," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1237–1242.
[8] 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. Automat. Lett.*, vol. 3, no. 3, pp. 2478–2484, Jul. 2018.
[9] 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.
[10] A. González-Morgado, E. Cuniato, M. Tognon, G. Heredia, R. Siegwart, and A. Ollero, "Controlled shaking of trees with an aerial manipulator," *IEEE/ASME Trans. Mechatron.*, early access, Jun. 14, 2024, doi: 10.1109/TMECH.2024.3410167.
[11] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015.
[12] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.
[13] S. Liu et al., "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017.
[14] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2520–2525.
[15] S. Liu, K. Mohta, N. Atanasov, and V. Kumar, "Search-based motion planning for aggressive flight in SE(3)," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2439–2446, Jul. 2018.
[16] Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu, and F. Gao, "Fast-racing: An open-source strong baseline for SE(3) planning in autonomous drone racing," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8631–8638, Oct. 2021.
[17] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6177–6184, Oct. 2020.
[18] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12759–12765.
[19] C. Wu, R. Wang, M. Song, F. Gao, J. Mei, and B. Zhou, "Real-time whole-body motion planning for mobile manipulators using environment-adaptive search and spatial-temporal optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 1369–1375.
[20] G. Zhang et al., "Grasp a moving target from the air: System & control of an aerial manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1681–1687.
[21] M. Wang et al., "Millimeter-level pick and peg-in-hole task achieved by aerial manipulator," *IEEE Trans. Robot.*, vol. 40, pp. 1242–1260, 2024.
[22] H. Cao, J. Shen, C. Liu, B. Zhu, and S. Zhao, "Motion planning for aerial pick-and-place with geometric feasibility constraints," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 2577–2594, 2024, doi: 10.1109/TASE.2024.3382296.
[23] 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. Automat.*, 2014, pp. 2113–2118.
[24] 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.
[25] J. Welde and V. Kumar, "Coordinate-free dynamics and differential flatness of a class of 6DOF aerial manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4307–4313.
[26] J. Welde, J. Paulos, and V. Kumar, "Dynamically feasible task space planning for underactuated aerial manipulators," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3232–3239, Apr. 2021.
[27] H. Seo, S. Kim, and H. J. Kim, "Locally optimal trajectory planning for aerial manipulation in constrained environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1719–1724.
[28] 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.*, 2019, pp. 7402–7408.
[29] R. Featherstone and D. E. Orin, *Dynamics*. Cham, Switzerland: Springer, 2016.
[30] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADI: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, pp. 1–36, 2019.

Zhaopeng Zhang received the B.S. degree in automation in 2021 from Nankai University, Tianjin, China, where he is currently working toward the Ph.D. degree in control science and engineering with the Institute of Robotics and Automation Information System.

His research interests include visual servo control and motion planning of aerial manipulator systems.

Hai Yu (Graduate Student Member, IEEE) received the B.S. degree in automation from Jilin University, Changchun, China, in 2020. He is currently working toward the Ph.D. degree in control science and engineering with the Institute of Robotics and Automatic Information System, Nankai University, Tianjin, China.

His research focuses on nonlinear control of unmanned aerial vehicles.

Yi Chai received the B.S. degree in automation and the M.S. degree in control science and engineering from Harbin Engineering University, Harbin, China, in 2015 and 2018, respectively, and the Ph.D. degree in control science and engineering from Nankai University, Tianjin, China, in 2022.

He is currently a Postdoctoral Fellow with the Institute of Robotics and Automatic Information System, Nankai University. His research interests include motion planning and nonlinear control for collaborative aerial transportation systems.

Zhichao Yang received the B.S. degree in automation in 2019 from Nankai University, Tianjin, China, where he is currently working toward the Ph.D. degree in control science and engineering with the Institute of Robotics and Automatic Information System.

His research interests include motion planning and nonlinear control of unmanned aerial vehicles.

Xiao Liang (Senior Member, IEEE) received the B.S. degree in intelligence science and technology from the Hebei University of Technology, Tianjin, China, in 2013, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, in 2018.

He is currently an Associate Professor with the Institute of Robotics and Automatic Information System, Nankai University. His research interests include motion planning and nonlinear control of unmanned aerial vehicle systems.

Yongchun Fang (Senior Member, IEEE) received the B.S. and M.S. degrees in control theory and applications from Zhejiang University, Hangzhou, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2002.

From 2002 to 2003, he was a Postdoctoral Fellow with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA. He is currently a Professor with the Institute of Robotics and Automatic Information System, Nankai University, Tianjin, China. His research interests include nonlinear control, visual servoing, control of underactuated systems, and atomic force microscopy-based nanosystems.

Dr. Fang was the recipient of the National Science Fund for Distinguished Young Scholars of China. He was an Associate Editor for the *ASME Journal of Dynamic Systems, Measurement, and Control.*

Jianda Han (Member, IEEE) received the Ph.D. degree in mechatronic engineering from the Harbin Institute of Technology, Harbin, China, in 1998.

From 1998 to 2003, he was a Visiting Scientist with the City University of Hong Kong, Hong Kong, China, Michigan State University, East Lansing, MI, USA, and Cornell University, Ithaca, NY, USA. He is currently a Professor with the Institute of Robotics and Automatic Information System, Nankai University, Tianjin, China. His research interests include nonlinear estimation and control, robotics, and mechatronics systems.