

Autonomous Hook-Based Grasping and Transportation With Quadcopters

Péter Antal^{ID}, Tamás Péni^{ID}, *Member, IEEE*, and Roland Tóth^{ID}, *Senior Member, IEEE*

Abstract—Payload grasping and transportation with quadcopters is an active research area that has rapidly developed over the last decade. To grasp a payload without human interaction, most state-of-the-art approaches apply robotic arms that are attached to the quadcopter body. However, due to the large weight and power consumption of these aerial manipulators, their agility and flight time are limited. This article proposes a motion control and planning method for transportation with a lightweight, passive manipulator structure that consists of a hook attached to a quadrotor using a 1-DoF revolute joint. To perform payload grasping, transportation, and release, first, time-optimal reference trajectories are designed through specific waypoints to ensure the fast and reliable execution of the tasks. Then, a two-stage motion control approach is developed based on a robust geometric controller for precise and reliable reference tracking and a linear-quadratic payload regulator for rapid setpoint stabilization of the payload swing. Furthermore, mathematical guarantees are provided for closed-loop stability to ensure safe operation. The proposed control architecture and design are evaluated in a high-fidelity physical simulator and also in real flight experiments, using a custom-made quadrotor-hook manipulator platform.

Index Terms—Stability of nonlinear systems, UAV planning and control.

I. INTRODUCTION

DUE to their advanced actuators, sensors, and autonomous navigation capability, interest in quadcopters is continuously growing both in industry and research. Nowadays, most industrial applications of quadrotors are limited to passive tasks, such as agricultural monitoring or recording camera footage. However, as drone technology matures, more and more challenges arise that require direct interaction and active manipulation of the environment. Our work is motivated by current industrial developments, where drones are used to solve autonomous transportation tasks, for example, transporting parts between production cells [1]. In these applications,

Received 14 October 2024; accepted 22 December 2024. Date of publication 23 January 2025; date of current version 25 April 2025. This work was supported in part by the European Union within the Framework of the National Laboratory for Autonomous Systems under Grant RRF-2.3.1-21-2022-00002 and in part by the Hungarian Research Network under Grant SA-77/2021. Recommended by Associate Editor L. Imsland. (*Corresponding author: Péter Antal.*)

Péter Antal and Tamás Péni are with the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control, 1111 Budapest, Hungary (e-mail: antalpetar@szaki.hun-ren.hu; peni@szaki.hun-ren.hu).

Roland Tóth is with the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control, 1111 Budapest, Hungary, and also with the Control Systems Group, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: tothroland@szaki.hun-ren.hu).

Digital Object Identifier 10.1109/TCST.2025.3529356

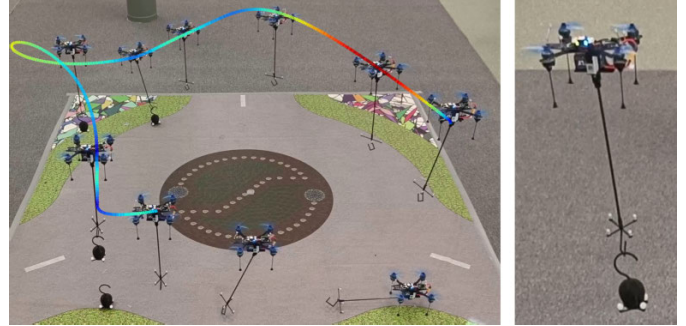


Fig. 1. Payload transportation by a hook-based manipulator.

the grasping and release require a gripping mechanism that can connect the payload to the drone, preferably without human interaction. Moreover, an advanced control system has to be developed to safely and reliably execute the task.

In the literature of transportation with quadcopters, the payload is usually connected to the vehicle via flexible cables that are attached manually before the drone takes off [2], [3], [4], [5]. The desired payload motion is then either achieved by swing-free trajectory planning and control [6], [7] or exploiting differential flatness of the coupled system to design dynamically feasible, smooth trajectories [8], [9]. In [10], a meta-reinforcement learning approach is proposed for payload grasping and transportation with a magnetic gripper. Although the grasping is solved automatically, putting down the payload requires an external detaching mechanism, limiting fully autonomous operation. The advantages of cable suspension include good maneuverability, simple mechanical structure, and the ability to apply multiple vehicles to collaboratively transport a heavy payload. However, due to the large number of degrees of freedom, object grasping and accurate placement without human interaction is highly challenging.

For automatic object grasping and transportation, quadcopters can be equipped with robotic extensions for aerial manipulations [11], [12], [13]. In [14], autonomous grasping is proposed with robotic claws, inspired by how predatory birds grasp their prey. The modeling and control design of the aerial manipulator are simplified by considering only the dynamics in the vertical plane. In [15], a custom-designed gripper is proposed specifically for grasping and carrying foam elements. In [16], a hex-rotor equipped with a 7-DoF manipulator is applied to grasp a moving payload. To control the motion of the multirotor, a PID controller has been developed. In [17], a 1-DoF magnetic gripper is attached to the quadcopter, and

minimum-snap trajectories are generated between predefined waypoints to complete the payload grasping and transportation task. Using the 1-DoF manipulator, the authors proposed a trajectory planning method in [18] to grasp a moving object. The proposed algorithm can design directly time-optimal trajectories (without the inherent suboptimality of predefined parametrizations, such as polynomials) and automatically find the optimal grasping time by employing complementarity constraints, making it a powerful approach to solve this challenging task. However, the corresponding optimization problem is complex and highly nonlinear, resulting in a computation time of more than 10 s, which makes it difficult to use when real-time replanning is needed.

Compared to a cable suspension, the main advantage of robotic arms and grippers is that they are actuated, which makes gripping possible even in the presence of small positioning errors. However, most of them are either designed for a specific task, material, and payload or require large and less-agile quadcopter platforms, which, due to their heavy weight and power consumption, limit general applicability.

In this work, we combine the agility and maneuverability of cable suspension with the automatic grasping and transportation ability of aerial manipulators. We propose a grasping mechanism that consists of a hook fixed to a rigid rod, which is connected to a quadrotor by a 1-DoF passive revolute joint. As it is illustrated in Fig. 1, we intend to move objects (with a hook attached on top) by picking them up and putting them down precisely at prespecified locations in minimal time. The advantages of our setup compared to the previously introduced configurations are as follows. First, the applied pole is more rigid than flexible cables and only has one degree of freedom, which decreases the complexity of precise positioning and makes modeling and tracking of the cable tension unnecessary. Second, the gripping mechanism is not actuated, resulting in a simple, lightweight, and cheap mechanical design. Moreover, the solution is also scalable: by equipping a heavy payload with hooks, it can be transported using multiple vehicles collaboratively.

The operational advantages brought by the proposed gripper mechanism come with the price of major unresolved challenges in control system design. To attach and detach the hooks, advanced planning and precise positioning are required because large tracking errors can lead to unsuccessful grasping and release of the payload. Moreover, the control system of the quadcopter needs to handle the force and torque disturbances introduced by the hook mechanism with the attached load. Finally, many existing methods, such as differential flatness-based algorithms, cannot be applied to 1-DoF hook-based manipulators, and few works address the complete task of payload grasping, transportation, and release.

To address the introduced challenges, the contributions of our work are as follows.

- 1) We propose a path planning solution adapted to hook-based transportation: the reference trajectory is constructed based on five segments, each designed using a computationally efficient, minimal time planning method inspired by [19].

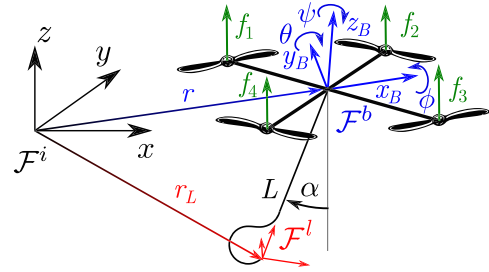


Fig. 2. Coordinate frames describing the geometric relations of the quadcopter and the environment.

- 2) We derive an extension of the robust geometric controller originally suggested in [20] to achieve precise and reliable tracking of agile trajectories despite all the disturbances caused by the hook and the payload.
- 3) For rapidly damping the swing of the payload and ensuring precise release at the target location, we switch to a linear-quadratic payload regulator (LQR). We provide mathematical guarantees for closed-loop stability despite the switching between the geometric controller and the LQR.
- 4) We test and analyze the performance of the proposed method in a high-fidelity simulation environment and also in real-world flight experiments using a custom-made aerial manipulator platform.

To the best of our knowledge, we provide the first solution for fully automated payload grasping, transportation, and placement using a passive manipulator attached to a quadcopter.

This work is organized as follows. First, we introduce the dynamic model of the quadrotor and the hook-based manipulator in Section II. In Sections III and IV, the proposed trajectory planning and control algorithms are presented. In Sections V and VI, we evaluate the performance of the proposed methods in high-fidelity physical simulations and also in real-world flight experiments. Finally, the conclusions are presented in Section VII.

II. MODELING

A. Quadrotor Dynamics

In this section, the dynamic model of the quadrotor without the hook is derived. The vehicle is modeled as a rigid body in 3-D space affected by the gravitational force and the rotation of the propellers. Using the inertial frame ($\mathcal{F}^i$) and the body frame ($\mathcal{F}^b$) illustrated in Fig. 2, the motion of the quadcopter is governed by the following equations:

$$m\ddot{r} = \underbrace{-mg\mathbf{e}_3 + FRe_3}_{\mathbf{f}_{\text{quad}}} \quad (1a)$$

$$J\dot{\omega} = \underbrace{\tau - \omega \times J\omega}_{\mathbf{f}_{\text{quad}}^{\omega}}, \quad \dot{R} = R\hat{\omega} \quad (1b)$$

where $r(t) \in \mathbb{R}^3$ is the position of the drone, $R(t) \in \text{SO}(3)$ is the rotation matrix from $\mathcal{F}^b$ to $\mathcal{F}^i$, $\text{SO}(3)$ denotes the special orthogonal group, $\omega(t) \in \mathbb{R}^3$ is the angular velocity in $\mathcal{F}^b$, m is the mass of the drone, J is the inertia matrix in $\mathcal{F}^b$, g is

the gravitational acceleration, and e_3 is the unit vector of the z -axis in $\mathcal{F}^i$. The notation $\hat{\cdot}$ stands for the projection: $\mathbb{R}^3 \rightarrow \text{SO}(3)$ ensuring that $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$, where $\times$ corresponds to the vector product. Besides R , we also use the Euler angles (roll: ϕ , pitch: θ , and yaw: ψ) to characterize the rotation from $\mathcal{F}^b$ to $\mathcal{F}^i$, as it is illustrated in Fig. 2. The inputs of the system are the collective thrust F and the torque vector $\tau = [\tau_x \ \tau_y \ \tau_z]^\top$ [21].

The dynamic model of the quadcopter is underactuated as it has 6 DoFs, but only four control inputs. However, in trajectory design, we exploit that the model is *differentially flat* [21], [22], that is, all state variables and control inputs can be expressed from a finite number of time derivatives of the *flat outputs*, namely the position x, y, z and the yaw angle ψ . This simplifies the trajectory planning as all other states depend on the desired $x_d(t), y_d(t), z_d(t), \psi_d(t)$ trajectories, therefore only these have to be constructed instead of considering the full state. Moreover, utilizing differential flatness resolves underactuation, because the reference trajectories designed for the flat outputs inherently satisfy the dynamics given by (1).¹

B. Manipulator Dynamics

Next, we complete the quadcopter model with the manipulator dynamics based on the modeling concept in [23]. We choose the vector of generalized coordinates as $\xi = [r^\top \ \lambda^\top \ \alpha]^\top$, $\xi(t) \in \mathbb{R}^7$, where $\lambda = [\phi \ \theta \ \psi]^\top$ and $\alpha(t) \in \mathbb{R}$ is the swing angle, shown in Fig. 2. Then, we use the Euler–Lagrange formulation to construct the equations of motion. For this, we introduce the Lagrange function $\mathcal{L} = \mathcal{K} - \mathcal{U}$, where $\mathcal{K}$ and $\mathcal{U}$ are the kinetic and potential energy of the system, respectively, computed as follows:

$$\mathcal{K} = \frac{1}{2} m \dot{r}^\top \dot{r} + \frac{1}{2} \dot{\lambda}^\top Q^\top J Q \dot{\lambda} + \frac{1}{2} m_L \dot{r}_L^\top \dot{r}_L \quad (2a)$$

$$\mathcal{U} = m g r^\top e_3 + m_L g (r + R r_b^\top)^\top e_3 \quad (2b)$$

where $\dot{r}_L = \dot{r} - \widehat{R r_b^\top} R Q \dot{\lambda} + R \dot{r}_b^\top$, $Q(t) \in \mathbb{R}^{3 \times 3}$ is the standard transformation matrix from the derivatives of the Euler angles to the angular velocity [23], L is the pole length, $r_b^\top = [-L \sin(\alpha) \ 0 \ -L \cos(\alpha)]^\top$, and m_L is the payload mass. Here, we assume that the revolute joint is located at the center of mass of the quadcopter. Although these are not perfectly aligned on the experimental platform, the gap between them is relatively small compared to the pole length (around 10%), therefore it can be neglected. This simplification leads to a reasonable tradeoff between model complexity and control performance.

We use the Euler–Lagrange equations $(d/dt)(\partial \mathcal{L} / \partial \dot{\xi}_i) - (\partial \mathcal{L} / \partial \xi_i) = u_{g,i}$ for $i = 1, \dots, 7$ to describe the dynamics of the manipulator, as follows:

$$H(\xi) \ddot{\xi} + C(\xi, \dot{\xi}) \dot{\xi} + G(\xi) = \Xi(\xi) u \quad (3)$$

¹ Although the underactuation is resolved, input constraints are challenging to handle when the trajectory is designed for the flat outputs. Our solution for this problem is discussed in Section III-C.

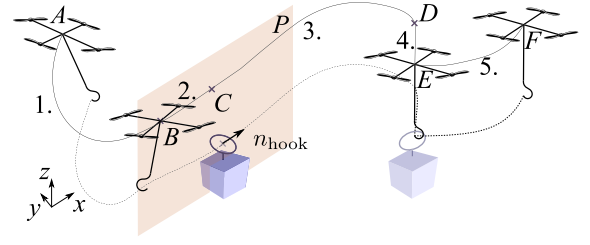


Fig. 3. Motion trajectory segments and waypoints.

with coefficient matrices

$$H_{i,j} = \frac{\partial^2 \mathcal{K}}{\partial \dot{\xi}_i \partial \dot{\xi}_j}, \quad G_k = \frac{\partial \mathcal{U}}{\partial \xi_k}, \quad \Xi = \begin{bmatrix} R e_3 & 0_{3 \times 3} \\ 0_{3 \times 1} & Q^\top \\ 0_{3 \times 1} & 0_{3 \times 3} \end{bmatrix}$$

$$C_{i,j} = \sum_{k=1}^7 \frac{1}{2} \left(\frac{\partial H_{i,j}}{\partial \xi_k} + \frac{\partial H_{i,k}}{\partial \xi_j} + \frac{\partial H_{j,k}}{\partial \xi_i} \right) \dot{\xi}_k \quad (4)$$

for $i, j, k \in \{1, \dots, 7\}$. The control input vector is $u = [F \ \tau^\top]^\top$, and $u_g = \Xi(\xi)u$ is the generalized force vector. From (3), $\ddot{\xi} = f_{\text{full}}(\xi, \dot{\xi}, u)$ can be expressed, which we will use later in controller design.

Despite the singularity of (3) due to the use of Euler angles, this model is adequate for the control problem at hand where we will only use it to describe motions of the payload around the stable equilibrium point. Compared to the quadrotor dynamics, the hook mechanism with attached load introduces zero moment in pitch, a nonzero moment in roll and yaw, and a force in the xz plane.

III. MOTION TRAJECTORY PLANNING

A. Overall Concept

To perform successful payload grasping and transportation, we propose a motion sequence for the aerial manipulator consisting of five segments: 1) reach the plane of the payload with the quadrotor; 2) approach the payload and attach the hook; 3) take the payload to the target position; 4) release the payload; and 5) detach the hook. The trajectory segments are depicted in Fig. 3 with waypoints A, B, C, D, E, and F describing the connection points between the segments and the plane P , which is spanned by the normal vector of the hook on the payload (n_{hook}), and the z -axis of the world frame. The initial and target configurations of the quadrotor and the payload are assumed to be specified by the user in each individual scenario. Based on these inputs, the position of the waypoints A, C, D, E, and F are fixed, while the position of waypoint B is determined by the solution of the trajectory optimization.

In trajectory planning, we neglect the dynamics introduced by the manipulator and the payload as their effects will be compensated by the trajectory tracking controller (see Section IV). Consequently, we design reference trajectories for the flat outputs of the quadrotor system derived in Section II-A, namely the position and the yaw angle, while the rest of the states in terms of roll and pitch and their velocities will be completely determined through the kinematic relations.

For the position, each segment is designed in two steps, based on an indirect time optimal trajectory planning method inspired by [19]. First, we design a *spatial trajectory* $r_d(s)$

as a function of the path variable s and then we construct a *temporal trajectory* by determining the time dependence of s . Both the spatial and temporal trajectories of a segment are designed before moving on to the next one.

B. Spatial (x, y, z) Trajectory

We represent the spatial trajectory by B-spline curves defined via their basis functions and coefficients as follows:

$$r_d(s) = [x_d(s) \quad y_d(s) \quad z_d(s)]^\top = c^\top \vartheta(s) \quad (5)$$

where $\vartheta : \mathbb{R} \rightarrow \mathbb{R}^{n_c}$ is the B-spline basis of degree n_r , $c = [c_x \ c_y \ c_z] \in \mathbb{R}^{n_c \times 3}$ is the coefficient matrix, and $s(t) \in [0, 1]$ is the progress variable along the B-spline curve. This formulation ensures that the designed path is $n_r - 2$ times continuously differentiable given the knots of the B-splines are distinct. Furthermore, the B-spline and its derivatives are linear in the coefficients allowing us to formulate trajectory optimization as a quadratic problem with linear constraints.

For the spatial trajectory, we minimize the jerk and the arc length of the curves with the following objective function [24]:

$$J_{\text{jerk}} = \sum_{f \in \{x_d, y_d, z_d\}} \int_0^1 \left(\frac{d^3 f(s)}{ds^3} \right)^2 ds \quad (6a)$$

$$J_{\text{length}} = \int_0^1 \sqrt{(x'_d(s))^2 + (y'_d(s))^2 + (z'_d(s))^2} ds \quad (6b)$$

$$J = w J_{\text{jerk}} + (1 - w) J_{\text{length}}, \quad w \in [0, 1] \quad (6c)$$

where w is a tradeoff parameter chosen by the user. Throughout this work, $g'(s)$ denotes the derivative of function g with respect to (w.r.t.) s , and $\dot{g}(s(t))$ denotes the derivative of g w.r.t. time: $\dot{g}(s(t)) = g'(s(t))\dot{s}(t)$.

To incorporate the waypoints illustrated in Fig. 3, we prescribe boundary conditions on the spatial trajectory and also on its derivatives. This leads to a quadratic optimization problem of the coefficients of the spatial trajectory

$$\tilde{c}^* = \arg \min J(\tilde{c}) \quad (7a)$$

$$\text{s.t. } A\tilde{c} = b \quad (7b)$$

$$G\tilde{c} \leq h \quad (7c)$$

where $\tilde{c} \in \mathbb{R}^{3n_c}$ corresponds to the column-wise vectorization of c and the constraints determined by A, b, G , and h are discussed later in Section III-D.

C. Temporal (x, y, z) Trajectory

The optimal time allocation method we employ has been proposed in [19] and applied to quadcopter trajectory planning in [3] and [24]. The objective of the planning is to design the velocity profile of $s(t) : \mathbb{R} \rightarrow [0, 1]$ such that it minimizes the execution time $T = \int_0^1 (1/\dot{s}) ds$ of the trajectory while satisfying motion constraints for the quadrotor. Similar to [24], we penalize the acceleration of the drone along the path to reduce the control effort, as the inputs F, τ appear at the acceleration level in the dynamic model (1). Given that the spatial path $r_d(s)$ is already designed based on (7), the

constrained optimal time allocation problem is formulated as follows:

$$\min_{T \in \mathbb{R}_0^+, s \in [0, 1]^{[0, T]}} J_T = \int_0^1 \left(\rho \dot{s}^2 + (1 - \rho) \frac{1}{\dot{s}} \right) ds \quad (8a)$$

$$\text{s.t. } \dot{r}_d(0) = v_0 \quad (8b)$$

$$\dot{r}_d(1) = v_f \quad (8c)$$

$$-v_{\max} \leq \dot{r}_d(s(t)) \leq v_{\max} \quad (8d)$$

$$-a_{\max} \leq \ddot{r}_d(s(t)) \leq a_{\max} \quad (8e)$$

$$-\lambda_{\max} \leq \lambda(s(t)) \leq \lambda_{\max} \quad (8f)$$

$$\dot{s}(t) \geq 0, \quad t \in [0, T] \quad (8g)$$

where $\rho \in [0, 1]$ is a tradeoff parameter, and $\lambda(s(t))$ is the rate of the acceleration (similar to jerk, but differentiated w.r.t. the path variable s [24]). The initial and final velocity $v_0, v_f \in \mathbb{R}^3$ ensure the continuity of the trajectory segments, and the inequality constraints guarantee that the velocity, acceleration, and acceleration rate remain in their allowed range. The bounds $v_{\max}, a_{\max}, \lambda_{\max} \in \mathbb{R}^3$ depend on the specific quadrotor configuration and we treat them as hyperparameters. Later, we give an optimization-based approach to choose their values in Section III-F.

Optimization (8) is nonlinear and complicated to solve in this form. However, by the direct transcription method introduced in [19], it can be reformulated as a second-order cone optimization problem (SOCP) by discretizing the temporal trajectory over a grid of s , and assuming that $\ddot{s}$ is piecewise constant between adjacent grid points. The numerical solution of the resulting SOCP can be obtained in polynomial time by using an off-the-shelf convex solver.

D. Boundary Conditions of the Trajectory Segments

In this section, we formulate the corresponding boundary conditions for both the spatial and temporal trajectories of each segment, depicted in Fig. 3. The constraints on position and spatial derivatives are then incorporated into (7b) and (7c), while the constraints on time derivatives are included in (8b) and (8c). The segments are distinct in the sense that we construct both the spatial and temporal trajectories of a segment before moving on to the next one. From here on, we use the notation $r_{d,i}(s_i(t_i))$, $i \in \{1, \dots, 5\}$ for each trajectory segment, where $s_i(0) = 0$, $s_i(T_i) = 1$, and $t_i \in [0, T_i]$. To connect two successive trajectory segments, the continuity of the position and velocity are ensured by the following constraints:

$$r_{d,i+1}(0) = r_{d,i}(1) \quad (9a)$$

$$\dot{r}_{d,i+1}(0) = \dot{r}_{d,i}(1) \quad (9b)$$

where (9a) is added to (7b), and (9b) is included in (8b) and (8c).

Segment 1: In the first segment, the aim is to reach plane P from any given initial position r_0 and velocity v_0 , therefore at point A , the following constraints are set:

$$r_{d,1}(0) = r_0 \quad (10a)$$

$$\dot{r}_{d,1}(0) = v_0, \quad (10b)$$

where (10a) and (10b) are incorporated into (7b) and (8b), respectively.

At the end of the first segment (point B), we wish to achieve a smooth transition to plane P , therefore the position is required to be in the plane with zero velocity and acceleration perpendicular to the plane. To simplify the mathematical formulation of the constraints, without loss of generality, we fix the object plane P to the global xz plane. In the x -direction, we set the position to a safety distance from the object to ensure that the controller drives the vehicle to the plane even in the presence of small tracking errors. We also constrain the direction of the spatial trajectory to ensure that the projection of the velocity vector to the x -axis at point B points toward the payload. In the z -direction, we prescribe inequality constraints for both the position and the direction of the spatial trajectory to make the hook attach easier in the second segment. These constraints are formulated as follows:

$$\begin{aligned} y_{d,1}(1) &= 0, & y'_{d,1}(1) &= 0, & y''_{d,1}(1) &= 0 \\ x_{d,1}(1) &= x_B, & x'_{d,1}(1) &\geq \partial_s x_B \\ z_{B,\min} &\leq z_{d,1}(1) \leq z_{B,\max} \\ \partial_s z_{B,\min} &\leq z'_{d,1}(1) \leq \partial_s z_{B,\max} \end{aligned} \quad (11)$$

where x_B , $\partial_s x_B$, $z_{B,\min}$, $z_{B,\max}$, $\partial_s z_{B,\min}$, and $\partial_s z_{B,\max}$ are hyperparameters. Each constraint of (11) correspond to either position variables or spatial derivatives, hence these are included in (7b) and (7c).

Segment 2: At point C ,² we prescribe constraints for the position and the direction of the spatial path to ensure that it lies in plane P above the initial position of the payload, guaranteeing safe attachment of the hook

$$\begin{aligned} r_{d,2}(1) &= r_{L,\text{init}} + Le_3 \\ x'_{d,2}(1) &\geq 0, & y'_{d,2}(1) &= 0, & z'_{d,2}(1) &= 0 \end{aligned} \quad (12)$$

where $r_{L,\text{init}}$ is the initial position of the payload, and e_3 is the unit vector of the z -axis in the world frame. These constraints are added to (7b) and (7c).

Segment 3: Between points C and D , we transport the payload to be above the target position and stop there. We intend to achieve precise positioning at point D by fixing the first and second derivatives of the spatial path to zero in x - and y -directions

$$\begin{aligned} r_{d,3}(1) &= r_{L,\text{target}} + z_D \\ x'_{d,3}(1) &= y'_{d,3}(1) = x''_{d,3}(1) = y''_{d,3}(1) = 0 \\ \dot{r}_{d,3}(1) &= 0 \end{aligned} \quad \begin{aligned} (13a) \\ (13b) \end{aligned}$$

where $r_{L,\text{target}}$ is the target position of the payload and z_D is a safety height, treated as a hyperparameter, the value of which is determined based on preliminary simulation experiments. To construct the spatial and temporal trajectories, (13a) is added to (7b) and (13b) is included in (8c).

Segment 4: From point D to E , the quadcopter descends to release the payload. As an end condition, we set the z position

²When designing the trajectory of Segments 2–5, the initial conditions are set by the continuity constraints (9) based on the previous segment, therefore we only discuss the boundary conditions corresponding to the endpoint.

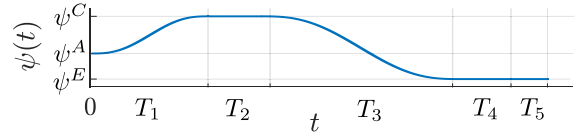


Fig. 4. Schematic trajectory of the yaw angle reference.

of the drone such that the hook is detached after the end of this segment. These constraints are written as follows:

$$r_{d,4}(1) = r_{L,\text{target}} + Le_3 - 0.5 d_h \quad (14a)$$

$$\dot{r}_{d,4}(1) = 0 \quad (14b)$$

where d_h is the diameter of the hook. Constraints (14a) and (14b) are incorporated into (7b) and (8c), respectively.

Segment 5: Unlike all other waypoints, the spatial trajectory at point E is not continuous as Segment 5 starts with a horizontal direction to ensure that the hook is detached successfully and the payload stays on the ground. The position r_F at point F can be chosen arbitrarily by the user, the only goal of this trajectory segment is to make sure that the payload is released. The constraints of this segment are formulated as

$$r_{d,5}(1) = r_F, \quad z'_{d,5}(0) = z''_{d,5}(0) = 0 \quad (15)$$

all of which are incorporated into (7b) in trajectory design.

Using the introduced boundary conditions, the spatial and temporal trajectories of each segment can be designed sequentially to obtain the reference trajectory of the quadcopter. All boundary conditions are linear in the decision variables, so they do not increase the complexity of the optimizations.

E. Yaw Trajectory

The yaw dynamics of a quadrotor are independent of the (x, y, z) path, therefore the yaw reference ψ_d can be freely chosen. The orientation of the hook is only restricted at points A , C , and E in terms of the initial quadrotor orientation, the orientation of the payload (ensuring that the hook can be attached to it), and the final payload orientation.

To provide a smooth transition between the specified yaw setpoints, we design fifth-order polynomial trajectories at Segments 1 and 3 and prescribe constant reference at the remaining segments. Given that the time duration of each segment is obtained from the solution of optimization (8), the initial and final values of the polynomial trajectories are known, and by prescribing zero initial and final angular velocity and angular acceleration, the coefficients of the polynomial trajectories can be directly obtained from the solution of a linear equation. A schematic trajectory is depicted in Fig. 4 to illustrate the polynomial segments.

F. Hyperparameter Optimization

The temporal trajectory optimization problem (8) enables to prescribe constraints for the velocity, acceleration, and differential acceleration. However, the exact formulation of the actuator constraints would introduce a set of nonlinear equations at the acceleration level, hence convexity of the optimization would be lost. To address this issue, we treat

the constraint bounds and the cost function weight in (6c) as hyperparameters and propose a simple optimization program to find the best parameters that result in successful grasping and minimal execution time. By defining the hyperparameter vector as $\gamma = [v_{\max} \ a_{\max} \ \lambda_{\max} \ w]^T \in \Gamma \subset \mathbb{R}^4$, the optimization problem is formulated as

$$\begin{aligned} \gamma^* = \arg \min_{\gamma \in \Gamma} \sum_{i=1}^5 T_i(\gamma) \\ \text{s.t. } \|r_b^L(T_C)\| \leq 0.5d_h \end{aligned} \quad (16)$$

where $r_b^L = r + Rr_b^1 - r_L$, $\|\cdot\|$ is the Euclidean vector norm, and d_h is the diameter of the hook. In (16), the inequality constraint ensures that the distance between the hook and the payload is less than the radius of the hook at point C, implying that the grasping is successful. To set up the optimization, Γ has to be specified by the user based on the specific quadcopter platform, the available flight area, and the intended agility of the maneuver. The cost is evaluated by drawing random initial conditions and simulating the trajectories of the system. The objective function and the constraint are difficult to express in closed form, therefore gradient-free global optimizer methods are applied to compute the solution, such as grid search, Bayesian optimization, or particle swarm optimization (PSO). The hyperparameters are calculated offline and only once for a given quadrotor platform and flight environment, thus it is feasible to use gradient-free methods despite their high computational cost.

IV. MOTION CONTROL

A. Overall Concept

We design a dual-mode motion control approach to execute the designed trajectory and ensure the successful realization of the transportation task. To follow the designed motion trajectory of the quadrotor in terms of the reference position r_d and the reference yaw angle ψ_d , we treat the hook-based manipulator as a disturbance acting on the quadrotor dynamics and employ a robust geometric control method. Moreover, to compensate for the gravitational force of the payload, we augment the controller with a feedforward (FF) term that is activated after attaching the hook to the payload. Robust geometric control can stabilize the quadcopter under bounded uncertainty, however, it does not control the payload swing, which makes precise positioning difficult. To solve this problem, we fix the reference at point D and switch to a linear-quadratic payload controller to stabilize the payload at the bottom equilibrium point. After a fixed time, the LQR is switched off and the geometric controller is reactivated to finish the maneuver.

The main benefits of the proposed control approach come from the complementary properties of the two controllers: the geometric controller ensures precise trajectory tracking and closed-loop stability while the system is in motion, and the LQR, which is only switched on during payload placement and release, stabilizes the payload swing, enabling accurate placement of the payload. The block diagram of the control scheme is displayed in Fig. 5. In the sequel, we introduce

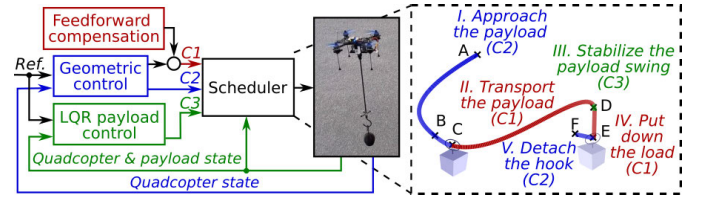


Fig. 5. Block diagram of the closed-loop control scheme with scheduling of the controllers along the motion trajectory (C1: geom. with FF, C2: geom., C3: LQR).

the control methods and provide mathematical guarantees that the motion remains stable during the whole maneuver, despite the controller switching.

B. Nonlinear Robust Geometric Control

For precise and reliable trajectory tracking, we use a nonlinear robust geometric controller based on [20], which can follow a reference trajectory in SE(3) in terms of position $r_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]^T \in \mathbb{R}^3$ and orientation $R_d(t) \in \text{SO}(3)$, where SE(3) denotes the special Euclidean group that comprises arbitrary translations and rotations of a rigid body in 3-D space. The reference orientation $R_d(t)$ is fully determined by the reference position $r_d(t)$ and the yaw angle $\psi_d(t)$. To synthesize the control law, the quadrotor dynamics given by (1) are augmented by additive state-dependent disturbances

$$m\ddot{r} = -mge_3 + FRe_3 + \Delta_r(\chi) \quad (17a)$$

$$J\dot{\omega} = \tau - \omega \times J\omega + \Delta_R(\chi) \quad (17b)$$

where $\chi = [\xi^T \ \dot{\xi}^T]^T$, and $\Delta_r(\chi), \Delta_R(\chi) \in \mathbb{R}^3$ comprise the disturbance caused by the payload motion and the hook-based manipulator on the quadrotor, bounded by $\|\Delta_r(\chi)\| \leq \delta_r$, $\|\Delta_R(\chi)\| \leq \delta_R$, where $\delta_r, \delta_R \in \mathbb{R}$ are fixed constants. Force and torque inputs are computed based on the following control law:

$$F = (-k_r e_r - k_v e_v + mge_3 + m\ddot{r}_d + \mu_r)^T R e_3 \quad (18a)$$

$$\begin{aligned} \tau = & -k_R e_R - k_\omega e_\omega + \omega \times J\omega \\ & - J(\hat{\omega} R^T R_d \omega_d - R^T R_d \dot{\omega}_d + \mu_R) \end{aligned} \quad (18b)$$

where $k_r, k_v, k_R, k_\omega \in \mathbb{R}$ are the control gains. The tracking errors $e_r(t), e_v(t), e_R(t), e_\omega(t) \in \mathbb{R}^3$ and the robust control inputs $\mu_r(t), \mu_R(t) \in \mathbb{R}^3$ are defined as follows:

$$\begin{aligned} e_r &= r - r_d, \quad e_v = \dot{r} - \dot{r}_d, \\ e_R &= \frac{1}{2}(R_d^T R - R^T R_d)^\vee, \quad e_\omega = \omega - R^T R_d \omega_d \\ \mu_r &= -\frac{\delta_r^{\kappa+2} e_B \|e_B\|^\kappa}{\delta_r^{\kappa+1} \|e_B\|^{\kappa+1} + \epsilon_r^{\kappa+1}}, \quad e_B = e_v + \frac{c_1}{m} e_r \\ \mu_R &= -\frac{\delta_R^2 e_A}{\delta_R \|e_A\| + \epsilon_R}, \quad e_A = e_\omega + c_2 J^{-1} e_R \end{aligned} \quad (19)$$

where $c_1, c_2, \epsilon_r, \epsilon_R, \kappa$ are positive constants, $\kappa > 2$, and the *vee operator* $(\cdot)^\vee : \text{SO}(3) \rightarrow \mathbb{R}^3$ is the inverse of the hat operator $(\hat{\cdot})$ introduced in Section II. It is proved in [20] that the tracking errors $(e_r, e_v, e_R, e_\omega)$ are uniformly ultimately bounded, using the control law (18), when the control gains are chosen according to the stability conditions in [20].

To obtain the uncertainty bounds δ_r, δ_R , we maximize the norm of the difference between the full manipulator model (3) and the quadcopter model (1) over the operating domain. For this, we transform (3) using $\omega = Q\dot{\lambda}$, as follows:

$$m\ddot{r} = m \underbrace{\begin{bmatrix} I & 0_{3 \times 3} & 0_{3 \times 1} \end{bmatrix} f_{\text{full}}(\chi, u)}_{f_{\text{full}}^r} \quad (20a)$$

$$J\dot{\omega} = J \underbrace{\begin{bmatrix} 0_{3 \times 3} & Q & 0_{3 \times 1} \end{bmatrix} f_{\text{full}}(\chi, u)}_{f_{\text{full}}^\omega} + \dot{Q}Q^{-1}\omega \quad (20b)$$

where $f_{\text{full}}(\chi, u)$ is expressed by rearranging (3) such that $\ddot{\xi} = f_{\text{full}}(\chi, u)$. Then, we formulate the following optimization problems to find δ_r, δ_R :

$$\delta_r = \max_{\substack{\chi \in \mathcal{X} \\ u \in \mathcal{U}}} \left\{ \left\| f_{\text{full}}^r - f_{\text{quad}}^r \right\| \right\} \quad (21a)$$

$$\delta_R = \max_{\substack{\chi \in \mathcal{X} \\ u \in \mathcal{U}}} \left\{ \left\| f_{\text{full}}^\omega - f_{\text{quad}}^\omega \right\| \right\} \quad (21b)$$

where $\mathcal{X}$ is the operating region and $\mathcal{U}$ is the set of admissible control inputs. Optimization (21) is solved using CasADi with IPOPT, which is an efficient nonlinear optimizer [25].

Before payload grasping, that is, between points A and C of the trajectory illustrated in Fig. 3, we use the control law (18). However, the payload changes the mass of the system significantly, therefore we augment the controller with an FF thrust to compensate for the gravitational force

$$F_L = F + m_L g e_3^\top R e_3 = F + F_{\text{ff}}. \quad (22)$$

The resulting thrust given by (22) can compensate for the vertical tracking error imposed by the transported payload, if the orientation of the quadrotor is near hovering, otherwise the robustness of the control law ensures that the quadcopter body remains stable under the bounded disturbance caused by the swinging of the payload.

C. Linear-Quadratic Payload Regulator

The geometric controller renders the quadcopter able to perform fast, agile maneuvers, however, it does not control the payload swing, which makes precise positioning of the payload difficult. To make sure that the payload is stabilized at the equilibrium point before detaching the hook, we use a linear-quadratic regulator [26].

We formulate the local LTI dynamics based on the first-order Taylor approximation of the nonlinear equations of motion described by (3) as follows:

$$\dot{\chi} \approx \frac{\partial f(\chi, u)}{\partial \chi} \bigg|_{\chi_0, u_0} (\chi - \chi_0) + \frac{\partial f(\chi, u)}{\partial u} \bigg|_{\chi_0, u_0} (u - u_0) \quad (23)$$

where $\chi = [\xi^\top \dot{\xi}^\top]^\top$, $u = [F \ \tau^\top]^\top$, $\chi_0 = 0_{14 \times 1}$, $u_0 = [(m + m_L)g \ 0_{1 \times 3}]^\top$, and $f = [\ddot{\xi}^\top f_{\text{full}}(\chi, u)^\top]^\top$. The control input provided by the LQR controller is calculated as $u = u_0 - K(\chi - \chi_0)$, where $K \in \mathbb{R}^{4 \times 14}$ is the feedback gain matrix that minimizes the cost function $\int_0^\infty ((\chi - \chi_0)^\top W_Q (\chi - \chi_0) + (u - u_0)^\top W_R (u - u_0)) dt$ w.r.t. the LTI dynamics described by (23).

D. Stability Analysis of Controller Switching

During the execution of the transportation task, two controller switches occur: from geometric control to LQR after Segment 3 and back to geometric control before Segment 4. In this section, we analyze the stability of the closed-loop system at these two switching points. We conduct the stability analysis based on the nominal system model, and we assume that it represents the real dynamics accurately. Although the proposed algorithm could be extended to account for model uncertainties, robust analysis is beyond the scope of this article.

Stability analysis of nonlinear systems is often performed by Lyapunov's method [27], however, constructing a Lyapunov function for a high-dimensional system is challenging, in general. Hence, we employ a data-driven technique based on the scenario approach [28] to analyze the stability of the closed-loop system $f_{\text{cl}}(\chi) = f_{\text{full}}(\chi, u_0 - K(\chi - \chi_0))$ governed by the LQR controller. For this, first we select the operating region of the system as follows: $\mathcal{X} = \{\chi \mid |\chi_i - \chi_{0,i}| \leq a_i, a_i \in \mathbb{R}^+, i = 1, \dots, 14\}$, where a contains the bounds on the absolute value of each element of $\chi - \chi_0$. Then, we intend to prove that any trajectory starting in $\mathcal{X}$ converges to zero, which certifies that $\mathcal{X}$ is a subset of the region of attraction (ROA) of the nonlinear system [27].

To prove our hypothesis, N number of independent, identically distributed samples are drawn from $\mathcal{X}$, considered as initial states, and the convergence of the corresponding trajectories are checked in simulation. Using the terminology of [28], each simulation is interpreted as a *scenario*, and the convergence of the corresponding trajectory is the *decision* $\theta \in \Theta$, $\Theta = \{0, 1\}$, where 1 corresponds to stability and 0 to instability. Hence, the *scenario decision* θ_N^* is also either 1 if all trajectories converge or 0 otherwise. The *violation probability* of a given decision $\theta \in \Theta$ is defined as $V(\theta) := P\{\chi^0 \in \mathcal{X} : \theta = 0\}$, where χ^0 is a sampled initial state, and P denotes probability. In [28], it is proved that, given a *confidence parameter* β and number of drawn samples N , it holds that $P\{V(\theta_N^*) > \varepsilon(s_N^*)\} \leq \beta$, where s_N^* is the *cardinality* of the scenarios, while $\varepsilon(k) = 1 - \sqrt[N-1]{\beta N^{-1} \binom{N}{k}^{-1}}$ for $k \in \mathbb{Z}$, $1 \leq k < N$ and $\varepsilon(N) = 1$. In general, s_N^* and thus $\varepsilon(s_N^*)$ cannot be computed a priori, because they depend on the sample points. However, in our case, $s_N^* = 1$ if all trajectories converge, then the scenario decision is $\theta_N^* = 1$, while a single divergence supports $\theta_N^* = 0$. Hence, the general results of [28] can be simplified, as follows:

$$P\{V(\theta_N^*) > \varepsilon(1)\} \leq \beta \quad (24a)$$

$$\varepsilon(1) = 1 - \sqrt[N-1]{\beta N^{-2}}. \quad (24b)$$

Based on (24b), it is possible to compute the minimum number of samples to satisfy (24a) for given β and $\varepsilon(1)$. For example, if we select $\beta = 10^{-6}$ and $\varepsilon(1) = 0.001$, then $N = 35\,000$ by (24b). If this many samples are drawn, and all trajectories converge, then, based on [28], $\mathcal{X}$ is proved to be a subset of the ROA with $1 - \varepsilon(1) = 99.9\%$ probability.

After the setpoint stabilization of the system at point D, the geometric controller is reactivated to finish the transportation

TABLE I

PARAMETERS OF THE MANIPULATOR AND THE GEOMETRIC CONTROLLER

Drone mass	m	$6.05 \cdot 10^{-1}$	kg	k_r	6.0
x axis inertia	J_x	$1.5 \cdot 10^{-3}$	kgm ²	k_v	3.0
y axis inertia	J_y	$1.45 \cdot 10^{-3}$	kgm ²	k_R	1.0
z axis inertia	J_z	$2.66 \cdot 10^{-3}$	kgm ²	k_ω	0.2
Rotor distance	l	$8.3 \cdot 10^{-2}$	m	c_1	1.0
Pole length	L	$4.0 \cdot 10^{-1}$	m	c_2	10^{-3}
Hook mass	m_h	$1.0 \cdot 10^{-2}$	kg	ϵ_r, ϵ_R	10^{-4}
Hook diameter	d_h	$4.0 \cdot 10^{-2}$	m	κ	3

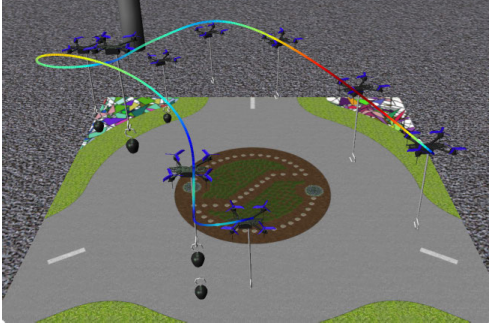


Fig. 6. Simulated transportation scenario using the digital twin model of the hook-based manipulator and the test environment. Note the close resemblance to Fig. 1, where the real-world experiments are illustrated.

task by detaching the hook. In this case, the ROA of the closed-loop system is described in [20] in terms of the physical parameters and the control gains. Therefore, at the beginning of Segment 4, we check the conditions of [20] to obtain whether the closed loop is stable and the switching can be executed safely.

V. SIMULATION STUDY

A. Simulation Environment

First, we analyze the proposed trajectory planning and control methods in simulated real-world scenarios using a high-fidelity physics simulator. The physical parameters of our custom-made quadrotor-hook manipulator platform are shown in Table I. To construct realistic simulations, we have implemented the digital twin model of the manipulator and the test environment, illustrated in Fig. 6. The dynamics are simulated by MuJoCo [29], an open-source physics engine that makes it possible to efficiently simulate complex dynamic systems with collisions and contacts. The source code of the simulations is available at GitHub.³

B. Trajectory Planner Design

To optimize the hyperparameters of the trajectory planner, (16) has been solved by grid search on the region $\Gamma = \{\gamma \mid v_{\max} \in [0.7, 3] \text{ m/s}, a_{\max} \in [0.1, 0.5] \text{ m/s}^2, \lambda_{\max} \in [0, 0.2], w \in [0.01, 0.9]\}$, in which ten uniformly distributed values for $v_{\max}$, $a_{\max}$, $\lambda_{\max}$ and four values for w have been considered as grid points (4000 cases). To test the feasibility and execution time of each parameter combination, six different scenarios of the grasping maneuver have been simulated in terms of the initial and final configuration of the

hook-based manipulator. The obtained numerical values are $v_{\max}^* = 1.46 \text{ m/s}$, $a_{\max}^* = 0.2 \text{ m/s}^2$, $\lambda_{\max}^* = 0.1$, $w^* = 0.5$.

Besides the optimized variables, there are other parameters, namely the degree n_r and the number of coefficients n_c of the B-spline curve representing the spatial trajectory, the number of grid points K by the temporal trajectory, and the control input weight ρ in optimization (8). The numerical values of these parameters have been chosen to be $n_r = 5$, $n_c = 12$, $K = 15$, and $\rho = 0.1$ based on the guidelines derived in [24] for similar trajectory optimization problems. Finally, the numerical value of the safety height defined by (13) is chosen to $z_D = 0.35 \text{ m}$ based on preliminary simulations.

For each of the five segments detailed in Section III-A, a quadratic program (QP) has been constructed for the spatial trajectory, and a second-order cone program (SOCP) has been set up for the temporal trajectory. To solve these problems, we have used Mosek,⁴ a large-scale, efficient optimization toolbox. On a Windows notebook with Intel i5 CPU and 8 GB RAM, the average computation time of the QP is $T_{QP} = 3.30 \text{ ms}$, while, for the SOCP, is $T_{SOCP} = 42.7 \text{ ms}$. These values correspond to one of the five segments, therefore the full optimization time for grasping, transportation and release is $T_{\text{opt,full}} = 5(T_{QP} + T_{SOCP}) = 230 \text{ ms}$ in average. However, the construction of constraint matrices and other matrix operations are computationally demanding, therefore the full computation time of the trajectory planning is significantly larger than the pure optimization times, it is $T_{\text{full}} = 1.54 \text{ s}$ on average. Although this computation time is already acceptable for real-world applications, it can be greatly reduced by embedded implementation.

C. Motion Control Design

The numerical values of the parameters of the robust geometric control are provided in Table I, which we have chosen based on the physical parameters of the quadcopter platform and the stability conditions of [20]. By solving optimization (21) with these parameters, the result yields $\delta_r = 0.28 \text{ N}$ and $\delta_R = 0.011 \text{ N}\cdot\text{m}$. These values are similar to the disturbance magnitudes considered in [20] relative to the physical properties of the quadcopter platform.

The weights of the LQR controller and the bounds of the operating region $\mathcal{X}$ have been chosen, as follows: $W_Q = \text{diag}(10, 10, 100, 0.1, 0.1, 0.1, 0.01, 1, 1, 10, 1, 1, 1, 0.05)$, $W_R = \text{diag}(5, 20, 20, 20)$, $a_i = 0.2$ for $i \in \{7, 14\}$, and $a_i = 0.1$ otherwise. For the stability analysis, $N = 35\,000$ uniformly distributed samples have been generated within $\mathcal{X}$. All simulated trajectories have converged, therefore under $\beta = 10^{-6}$, we conclude that with 99.9% certainty the closed-loop system is stable after controller switching, given that the geometric controller brings all states within $\mathcal{X}$ by the end of Segment 3. The latter condition is satisfied in all of the presented simulations, and in the real-world experiments, as well.

D. Results

To challenge the proposed methods, four transportation scenarios have been simulated in MuJoCo. The results are

³<https://www.github.com/AIMotionLab-SZTAKI/quadrotor-hook-1DoF/>

⁴<https://www.mosek.com/>

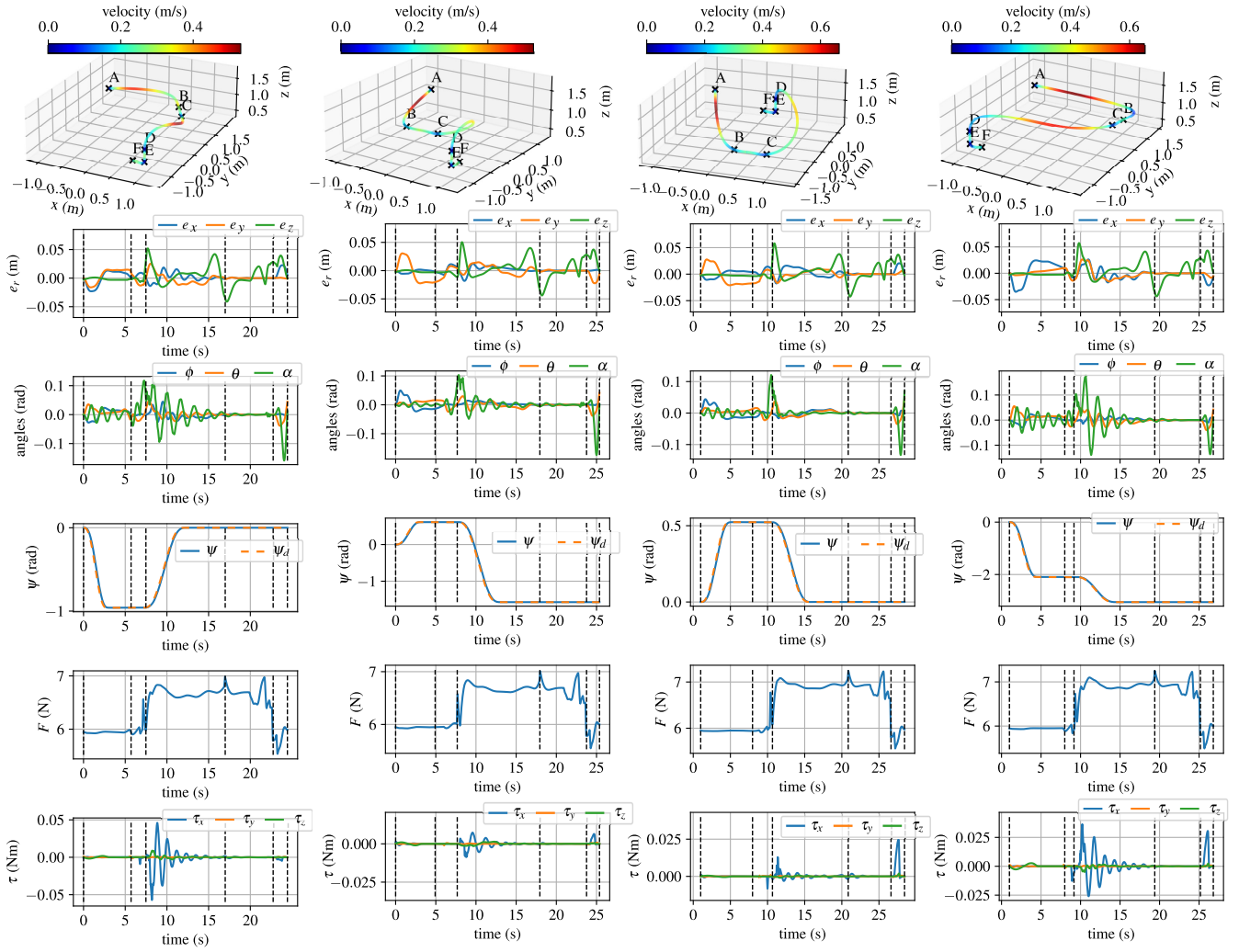


Fig. 7. Simulated scenarios of payload transportation: the 3-D plots display the reference trajectories, while the 2-D plots show the state and error trajectories. The payload mass has been set to 75 g at the first and second scenarios, and to 100 g at the third and fourth. The waypoints A, B, C, D, E, and F are indicated by dashed lines.

shown in Fig. 7, where each column corresponds to a specific scenario. The first row shows the reference position and velocity, proving that smooth trajectories can be generated using the proposed planning method to rapidly grasp the payload, take it to the target location, and release it.

To evaluate the effect of different payloads, the payload mass has been set to 75 g at the first two scenarios, and 100 g at the other two simulations. The root mean square error of the position tracking for each scenario has been $e_{r, \text{RMSE}} = \{2.1, 2.1, 2.1, 2.3\}$ cm, the peak absolute error has been $e_{r, \text{max}} = \{5.5, 5.3, 5.9, 5.9\}$ cm, and the final position error of each payload has been less than 1 mm. The trajectory of the swing angle α shows a maximum of 0.18 rad that appears after grasping the last payload. After attaching the hook, the swing of each payload becomes significant, however, the LQR can dampen payload oscillations in all cases to achieve precise positioning when detaching the hook. The yaw angle tracking is also accurate, the difference between the reference and state trajectories is insignificant. To conclude, the simulations show excellent control performance both in terms of reference tracking and robustness against the variation of the payload mass.

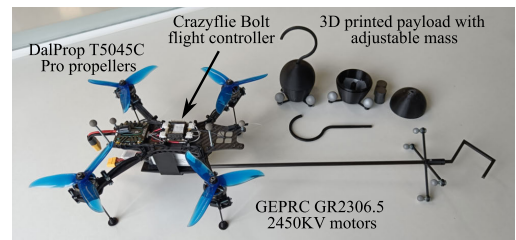


Fig. 8. Quadrotor with a hook-based manipulator and a 3-D printed payload used for the experiments.

VI. FLIGHT EXPERIMENTS

A. Experimental Setup

Flight experiments have been conducted with the proposed method and implemented on a custom-made quadrotor platform equipped with a 3-D printed hook and revolute joint, shown in Fig. 8. The onboard computation (including state estimation and control input calculation at 500 Hz) runs on a Crazyflie Bolt flight controller,⁵ including an IMU and two microcontroller units. An Optitrack motion capture system is used to provide high-precision pose information. A more

⁵<https://www.bitcraze.io/products/crazyflie-bolt-1-1/>

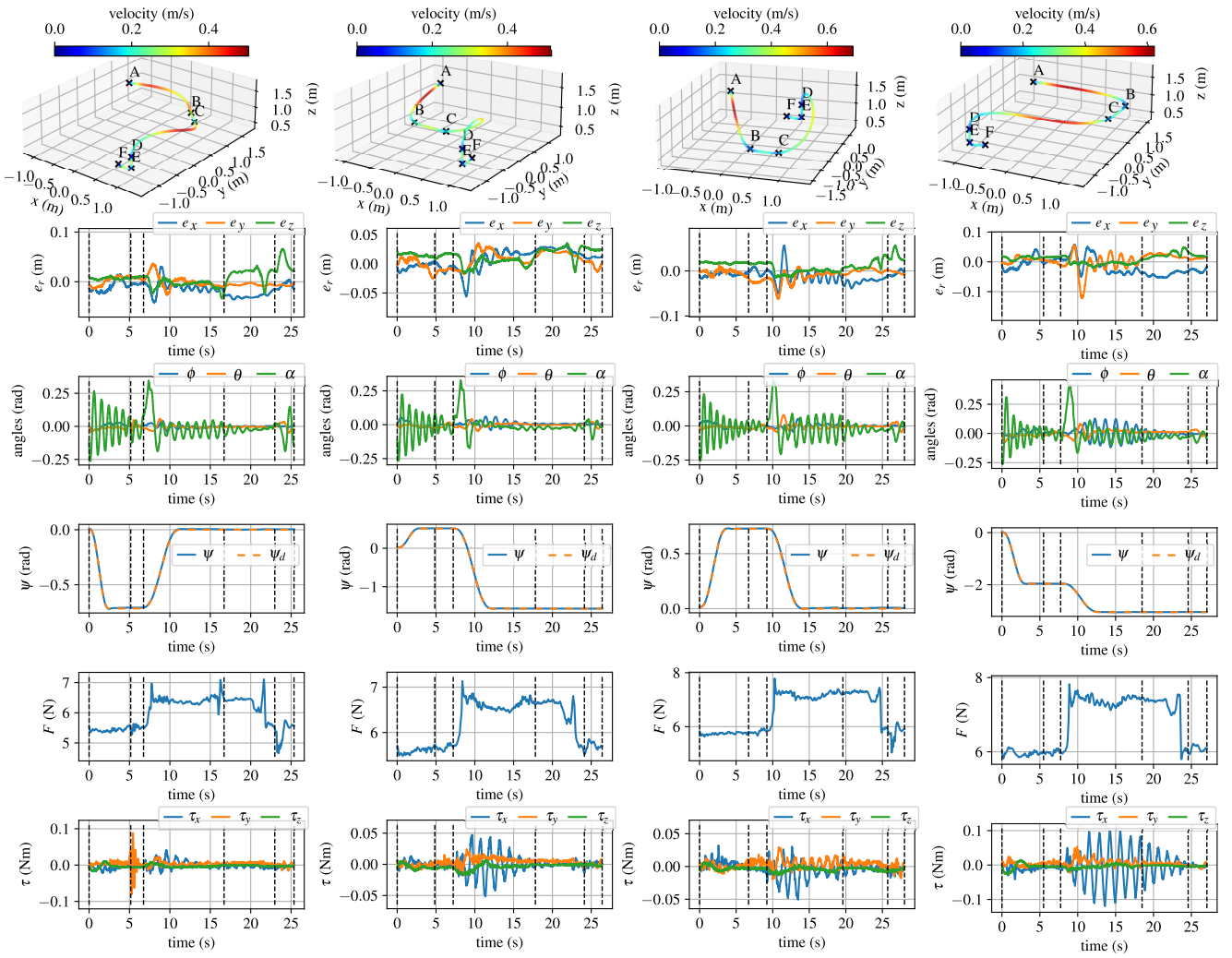


Fig. 9. Real-world transportation experiment: the reference trajectories, states, inputs, and waypoints are indicated identically to the simulation results.

detailed specification of the test environment is available in [30], and the physical parameters of the hook-based manipulator are shown in Table I.

We use a 3-D printed, drop-shaped payload for the experiments that is shown in Fig. 8. The height of the payload without the hook is 8 cm and the width is 6.5 cm. The mass can be configured by putting small weights inside the payload. Similar to the simulation study, we use a 75- and a 100-g payload for the flight experiments.

To complete payload grasping and transportation in real-world experiments, only the target position and orientation of the payload need to be specified by the user. The trajectory planner then automatically detects the pose of the quadcopter and the payload based on the motion capture system, runs the planning algorithm, and sends the trajectory to the vehicle. After the transportation is completed, a descending trajectory is computed and sent to the quadcopter. This way, the quadcopter can safely land with the attached manipulator.

B. Results

Measurement results for the flight experiments are displayed in Fig. 9, where the reference trajectories, states, and control inputs are shown similar to the simulations in Fig. 7. The root

mean square error of the position tracking for each scenario has been $e_{r, \text{RMSE}} = \{2.8, 2.9, 2.9, 4.3\}$ cm, and the final position error of each payload is $e_{r, \text{fin}} = \{1.8, 1.7, 1.0, 3.1\}$ cm, respectively. The trajectory of the swing angle α shows a maximum of 0.34 rad that appears after grasping the last payload with the highest mass. However, it is also shown experimentally that the LQR can dampen payload oscillations to achieve precise positioning when detaching the hook. Similar to the simulations, the yaw angle tracking error is insignificant in real experiments as well. Based on the measurement results, we conclude that the control performance is robust against the payload mass variation, and it is excellent in all examined scenarios, therefore the proposed methods have been validated successfully in real-world deployment.

It is important to note that unmodeled aerodynamics, more specifically drag, downwash, and ground effects can have a significant impact on the behavior of the system in real flight (see [31], [32]). For flat and light objects that have large aerodynamic resistance, the airflow generated by the propellers can cause the payload to swing on the hook, which can be challenging to compensate by the controller. Therefore, the modeling of this aerodynamic effect and compensation of it in the control design is an important task for future research.

Video footage of the flight experiments is available at <https://www.youtube.com/watch?v=FDqSrf4k9kw>.

VII. CONCLUSION

In this work, we have presented a complete methodology for payload grasping, transportation, and release using a quadrotor with a hook-based manipulator. To validate the proposed method, we have deployed the algorithms on a hardware platform and performed flight experiments. The demonstrated results show that the proposed trajectory planning and control methods are time efficient, robust, and can be applied in real flights.

ACKNOWLEDGMENT

The authors would like to thank Botond Gaál for his help during the flight experiments.

REFERENCES

- [1] O. Maghazei and T. Netland, "Drones in manufacturing: Exploring opportunities for research and practice," *J. Manuf. Technol. Manage.*, vol. 31, no. 6, pp. 1237–1259, Sep. 2019.
- [2] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5196–5202.
- [3] H. Hua, Y. Fang, X. Zhang, and S. Hu, "A time-optimal trajectory planning strategy for an aircraft with a suspended payload via optimization and learning approaches," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 6, pp. 2333–2343, Nov. 2022.
- [4] G. Li, A. Tunchez, and G. Loianno, "PCMP: Perception-constrained model predictive control for quadrotors with suspended loads using a single camera and IMU," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2012–2018.
- [5] H. Li, H. Wang, C. Feng, F. Gao, B. Zhou, and S. Shen, "AutoTrans: A complete planning and control framework for autonomous UAV payload transportation," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6859–6866, Oct. 2023.
- [6] H. Alkomy and J. Shan, "Vibration reduction of a quadrotor with a cable-suspended payload using polynomial trajectories," *Nonlinear Dyn.*, vol. 104, no. 4, pp. 3713–3735, 2021.
- [7] M. E. Guerrero-Sánchez, D. A. Mercado-Ravell, R. Lozano, and C. D. García-Beltrán, "Swing-attenuation for a quadrotor transporting a cable-suspended payload," *ISA Trans.*, vol. 68, pp. 433–449, May 2017.
- [8] B. Yu, K. Gamagedara, S. Kim, T. Lee, and J. Suk, "Geometric control and experimental validation for a quadrotor UAV transporting a payload," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 201–207.
- [9] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load—A differentially-flat hybrid system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4888–4895.
- [10] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra, and S. Levine, "Model-based meta-reinforcement learning for flight with suspended payloads," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1471–1478, Apr. 2021.
- [11] 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.
- [12] 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.
- [13] J. Saunders, S. Saeedi, and W. Li, "Autonomous aerial robotics for package delivery: A technical review," *J. Field Robot.*, vol. 41, no. 1, pp. 3–49, Jan. 2024.
- [14] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration Biomimetics*, vol. 9, no. 2, May 2014, Art. no. 025010.
- [15] F. Augugliaro et al., "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Syst. Mag.*, vol. 34, no. 4, pp. 46–64, Aug. 2014.
- [16] G. Zhang et al., "Grasp a moving target from the air: System & control of an aerial manipulator," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 1681–1687.
- [17] W. Luo, H. Eschmann, and P. Eberhard, "Gaussian process regression-augmented nonlinear model predictive control for quadrotor object grasping," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Dubrovnik, Croatia, Jun. 2022, pp. 11–19.
- [18] 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.
- [19] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.
- [20] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013.
- [21] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2520–2525.
- [22] M. J. Van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *Int. J. Robust Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, Sep. 1998.
- [23] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a UAV with a robotic arm," in *Proc. 10th Int. IFAC Symp. Robot Control*, vol. 45, Jan. 2012, pp. 704–709.
- [24] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal time allocation for quadrotor trajectory generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Jun. 2018, pp. 4715–4722.
- [25] J. A. E. 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, no. 1, pp. 1–36, Mar. 2019.
- [26] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. Hoboken, NJ, USA: Wiley, 1972.
- [27] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [28] M. C. Campi, S. Garatti, and F. A. Ramponi, "A general scenario theory for nonconvex optimization and decision making," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4067–4078, Dec. 2018.
- [29] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.
- [30] P. Antal, T. Péni, and R. Tóth, "Backflipping with miniature quadcopters by Gaussian-process-based control and planning," *IEEE Trans. Control Syst. Technol.*, vol. 32, no. 1, pp. 3–14, Jan. 2024.
- [31] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, "NeuroBEM: Hybrid aerodynamic quadrotor model," in *Proc. Robot. Sci. Syst.*, Jun. 2021, pp. 1–11.
- [32] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Experimental Robotics*. Cham, Switzerland: Springer, 2013, pp. 289–302.