**IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France**

Differential Flatness based Path Planning with Direct Collocation on Hybrid Modes for a Quadrotor with a Cable-Suspended Payload

Jun Zeng, Prasanth Kotaru, Mark W. Mueller and Koushil Sreenath

***Abstract*— Generating agile maneuvers for a quadrotor with a cable-suspended load is a challenging problem. State-of-the-art approaches often need significant computation time and complex parameter tuning. We use a coordinate-free geometric formulation and exploit a differential flatness based hybrid model of a quadrotor with a cable-suspended payload. We perform direct collocation on the differentially-flat hybrid system, and use complementarity constraints to avoid specifying hybrid mode sequences. The non-differentiable obstacle avoidance constraints are reformulated using dual variables, resulting in smooth constraints. We show that our approach has lower computational time than the state-of-the-art and guarantees feasibility of the trajectory with respect to both the system dynamics and input constraints without the need to tune lots of parameters. We validate our approach on a variety of tasks in both simulations and experiments, including navigation through waypoints and obstacle avoidance.**

I. Introduction

A. Motivation

Aerial manipulation with Unmanned Aerial Vehicles (UAVs) has wide applicability in aerial package delivery. Early aerial manipulation research employed a payload that was either rigidly attached to the quadrotor or attached to an actuated manipulator arm, see [8], [16]. An alternative method is to use a cable suspension system to replace the actuated manipulator. While this preserves the agility of the UAV, the resulting system is underactuated and hybrid, due to the multiple dynamical models depending on whether the cable is slack or taut. The nonlinear underacutated and hybrid model makes fast trajectory optimization for this system challenging. We address this challenge by directly doing direct collocation on the differentially-flat hybrid system, using complementarity constraints to avoid specifying hybrid mode sequences, using a coordinate-free geometric formulation to avoid singularties, and reformulating the obstacle avoidance constraint as a dual problem, resulting in faster optimization than prior work.

B. Related Work

1) Controllers: The state of the art for controllers in cable-suspended aerial manipulation can be generally classified into two approaches. The first approach is to use a feedback controller to minimize the swinging behavior of the transported load [4], [6], [11]. This method is not energetically efficient and the quadrotor has to counteract the swinging motion of the payload. Therefore, it does not exploit the system dynamics and prevents high-speed agile maneuvers. The other approach is to make use of the swing agility of the system and design the geometric controller for a quadrotor with a suspended load system [9], [12]–[14], [17], [19].

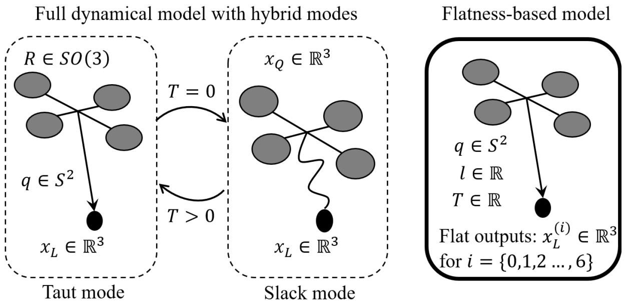


Fig. 1: Full dynamical model with hybrid modes and flatness-based model description and its generalized coordinates. Full dynamics (left) represents the hybrid dynamical model while the flatness-based one (right) describes the hybrid behavior of the cable-payload system through the enforcement of joint-limits on the prismatic joint l. The generalized coordinates of the flatness-based model also contain high order derivatives of load position $x_L^{(i)}$, load attitude q and cable tension T. From cable tension T and the prismatic joint l, we can identify whether the cable is slack or taut.

2) Motion Planning: Path planning for agile navigation leveraging the swinging dynamics of the system has been studied in [3], [5], [15]. Obstacles are represented using mixed integers to capture hard constraints of the obstacle faces in [15], while a conservative signed distance was proposed in [5]. Next, tasks such as waypoint navigation or obstacle avoidance often require a large number of parameters and must be tuned carefully in [3], [15]. Moreover, the exact order for switching hybrid modes are required to be predefined in [3], [15]. Furthermore, in [5], when constructing the optimization problem, the full dynamics was used as constraints, increasing the computational complexity. Some recent work [2], [18] on obstacle avoidance for aerial manipulation use different manipulator structures, which is quite innovative but less energy efficient and make the system more difficult to control.

C. Contributions

The contributions of the paper with respect to prior work are:

- We model the quadrotor-load system using states as high order derivatives of load position, load attitude, cable tension and the distance between quadrotor and payload,

All the authors are with the Department of Mechanical Engineering, University of California, Berkeley, California {zengjunsjtu, prasanth.kotaru, mwm, koushils}@berkeley.edu

This work was supported in part by NSF Grant CMMI-1840219 and in part by Berkeley Deep Drive.

Experimental videos are at https://youtu.be/e09RZOx_nZk.

presented in Figure 1. These states fully describe the system since the quadrotor-load system is differentially-flat with load position as flat outputs.

- We use direct collocation and differential-flatness to speed up the optimization and reduce the number of nonlinear constraints. Complementarity constraints are also used to represent the hybrid modes, so that mode sequences don't have to be predefined.
- The non-differentiable collision-free constraints between quadrotor-load system and obstacles are reformulated into smooth (differentiable) nonlinear constraints using strong duality of optimization. This formulation leads us to compute *energy-efficient*, *dynamically feasible*, *but safe* trajectories for a variety of tasks.

D. Paper Structure

This paper is organized as follows. Section II describes the differential flatness based model for quadrotor-load system with complementarity variables representing the hybrid modes. Section III presents path planning optimization formulation using direct collocation. The specifications for a variety of tasks are also illustrated. Section IV and V present numerical simulations and experiments. In Section VI, the benchmark of computation time and further discussions are presented and Section VII provides concluding remarks.

II. Model and System Variables

In the paper, we model the quadrotor as a rigid body and the payload as a point mass, controlled by a massless cable. The system comprising of a quadrotor with a cable suspended load is proved in [14] to be a differentially-flat hybrid system with a set of flat outputs (x_L, ψ) or (x_Q, ψ) when the cable is taut or not, respectively, where x_L, x_Q, ψ represent the load position, quadrotor position and quadrotor's yaw angle. The two hybrid modes are presented in Figure 1 and the list of variables used in this paper are presented in Table I. Recent research on aerial manipulation with cable-suspended payloads model the system using this hybrid approach. However, this renders trajectory optimization a complex problem, since it requires computing hybrid sequences subject to non-convex constraints on the system states.

$\mathscr{B}$, $\mathscr{W}$	Body and world frame
$m_Q \in \mathbb{R}$	Mass of quadrotor
$m_L \in \mathbb{R}$	Mass of suspended load
$R \in SO(3)$	Rotation matrix from $\mathscr{B}$ to $\mathscr{W}$
$x_Q, v_Q \in \mathbb{R}^3$	Position and velocity vectors of the quadrotor in $\mathscr{W}$
$x_L, v_L \in \mathbb{R}^3$	Position and velocity vectors of the payload in $\mathscr{W}$
$f \in \mathbb{R}$	Magnitude of the thrust for the quadrotor
$M \in \mathbb{R}^3$	Moment vector for the quadrotor in the $\mathscr{B}$
$q \in S^2$	Unit vector from quadrotor to the load
$l_0 \in \mathbb{R}$	cable's length when taut

TABLE I: Various symbols used in the paper.

We introduce a differential flatness based parameterization of the quadrotor-load system similar to [5] for path planning. Figure 1 shows the model description and the generalized coordinates. In this model, we describe the quadrotor's center of mass and the payload connected through a prismatic joint of length l, which is less than or equal to the original length of cable l_0 when the cable is slack. Instead of using two angles to represent the load attitude, in this paper, the load attitude is represented directly by the unit vector from quadrotor to load $q \in S^2$. This representation helps to reduce the computation time with the lower complexity for the auto-derivatives in the optimization solver. Notice that no matter if the cable is taut or slack, we always have

$$x_Q = x_L - l \cdot q. \tag{1}$$

Our generalized coordinates of the system include the flat outputs $x_L^{(i)}$ representing the derivatives of order i for payload position, where $i \in \{0,1,...,6\}$. Thus, we have,

$$\frac{dx_L^{(i)}}{dt} = x_L^{(i+1)} \quad \text{for } i \in \{0,1,...,5\}. \tag{2}$$

The optimization will search for the quantities $x_L^{(i)}$ subject to the dynamic constraints given in (2).

To fully describe the system's hybrid dynamics, the collocation relation described in (2) is not sufficient. We also introduce the scalar magnitude of the cable tension $T \in \mathbb{R}^+$ to describe the system dynamics,

$$-Tq = m_L \ddot{x}_L + m_L g e_3, \tag{3}$$

$$T(l - l_0) = 0, \tag{4}$$

where (3) and (4) represent the payload dynamics and the complementarity constraint, respectively. With (2), (3), (4), we can fully describe the system dynamics using the differential-flatness property.

Remark 1: Although the quadrotor attitude $R \in SO(3)$ is excluded from the generalized coordinates for path planning, we can still calculate it based on the optimized values of flat outputs after the optimization, when cable is taut.

III. Collocation-based Path Planning

We now propose the path planning optimization as a nonlinear optimization problem on N collocation states. We first note that (2) represents the payload collocation constraints and (3), (4) represent complementarity constraints. The optimization problem then is,

$$\min_{x_L^{(i)},T,l,q} \quad J(x_L^{(i)}, T, l, q) \tag{5a}$$

$$\text{s.t. } x_L^{(i)}(t_{k+1}) - x_L^{(i)}(t_k) = \frac{\Delta t_k}{2}(x_L^{(i+1)}(t_{k+1}) + x_L^{(i+1)}(t_k)), \tag{5b}$$

$$m_L \ddot{x}_L(t_k) = -T(t_k)q(t_k) - m_L g e_3, \tag{5c}$$

$$T \geq 0, \tag{5d}$$

$$T\,(l(t_k) - l_0) = 0, \tag{5e}$$

$$0 \leq l(t_k) \leq l_0, \tag{5f}$$

where (5b) represents the direct trapezoidal collocation [7] between two neighboring nodes with $k \in \{1,2,3,...,N-1\}$ and $i \in \{0,1,...,5\}$. The complementarity constraint is implemented in (5d), (5e) and we limit the prismatic distance between the quadrotor and the payload in (5f). The trapezoidal collocation presented in (5b) results in a second order approximation, enabling the computational efficiency.

In the following, we describe the cost function and other constraints for different scenarios.

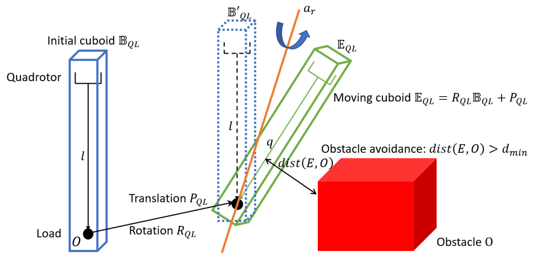

Initial cuboid $\mathbb{B}_{QL}$
Quadrotor
l
Load
O
$\mathbb{B}'_{QL}$
a_r
$\mathbb{E}_{QL}$
Moving cuboid $\mathbb{E}_{QL} = R_{QL}\mathbb{B}_{QL} + P_{QL}$
l
q
Obstacle avoidance: $dist(E,O) > d_{min}$
$dist(E,O)$
Translation P_{QL}
Rotation R_{QL}
Obstacle O

Fig. 2: Obstacle avoidance formulation. Notice that the quadrotor-load system is modelled as a controlled object with a square cuboid shape, denoted as $\mathbb{E}_{QL}(t)$. The load attitude is represented through the unit vector q and the cable length is l, which could be different from the nominal cable length l_0. The initial cuboid is first translated and then rotated based on the current configuration of the system. This translated and rotated cuboid $\mathbb{E}_{QL}(t)$ is then used for the obstacle avoidance constraint, $dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min}$.

1) Cost Function Formulation: We use a cost function J as below,

$$J = S_f t_f + \sum_{k=1}^{N} \left[x_L^{(6)}(t_k)^T Q x_L^{(6)}(t_k) + \bar{R}_0 T(t_k) + \bar{R}_1(l_0 - l(t_k))\Delta t_k + \bar{R}_2(l(t_k) - l(t_{k-1}))^2 \Delta t_k \right] \tag{6}$$

where $t_f = \sum_{k=1}^{N} \Delta t_k$ and S_f represent the final time and the terminal cost parameter, respectively. $Q \in \mathbb{R}^{3\times3}$ is a diagonal positive-definite matrix and $\bar{R}_0, \bar{R}_1, \bar{R}_2$ are three positive scalars. By adding a terminal time cost, the optimization will find the optimal time while also considering energy efficiency.

Remark 2: Besides the terminal time cost, there are four other parts of the cost function. We use $\sum_{k=1}^{N} x_L^{(6)}(t_k)^T Q x_L^{(6)}(t_k)\Delta t_k$ to minimize the energy consumption, since the torque of the quadrotor depends on the 6^{th} derivative of the payload position through differential-flatness property, as proved in [13]. Additionally, we have $\sum_{k=1}^{N} \bar{R}_0 T(t_k)\Delta t_k$ and $\sum_{k=1}^{N} \bar{R}_1(l_0 - l(t_k))\Delta t_k$ representing the cost on the taut and slack modes, respectively. From our intuitions, when $\bar{R}_1$ becomes dominant, almost all nodes along the trajectory will stay in the taut mode. Inversely, when $\bar{R}_0$ becomes larger, the solution will tend to have more slack nodes. Thus, with these two parts of cost function and the complementarity constraints, the hybrid modes need not be predefined, with the optimization finding the appropriate modes. Another advantage of having the term $\sum_{k=1}^{N} \bar{R}_1(l_0 - l(t_k))\Delta t_k$ is to avoid having excessive slack phases. The term $\bar{R}_2(l(t_k) - l(t_{k-1}))^2\Delta t_k$ is required to help make the distance between quadrotor and payload transition smoother between the taut and the slack modes.

2) Input Constraints: According to the physical restrictions of the actuated system, we need to add constraints on the optimization variables. For example, the cable tension should have an upper bound T_{max}, where we have

$$0 <= T(t_k) <= T_{max}. \tag{7}$$

We could estimate the maximum of cable tension from the maximum thrust of the quadrotor.

3) Geometric constraints: We could constrain the load attitude $q(t_k)$ to prevent the load swinging over the quadrotor and prevent collision between the payload and the propellers,

$$-q \cdot e_3 \geq cos(\alpha_{max}) \tag{8}$$

where the α_{max} represents the maximum swing angle for load attitude.

4) Obstacle Avoidance: Here we present our novel implementation of transferring the obstacle avoidance constraints into smooth nonlinear constraints using strong duality of optimization, where the quadrotor-load system is considered as as a full-dimensional controlled object. Figure 2 illustrates the obstacle avoidance formulation.

We model the quadrotor-load system as a controlled object $\mathbb{E}_{QL}(t)$ with square cuboid shape, whose the central axis represents the prismatic joint $l(t)$. Although the quadrotor attitude R is excluded from the generalized coordinates for path planning, we actually don't simply consider the quadrotor as a point mass for obstacle avoidance. Instead, we use ε_x, ε_y (the cross section of the cuboid) and ε_z (the additional maximum prismatic length), all of which are determined by the size of quadrotor and load. This moving square cuboid at each timestamp can be modelled through the rotation $R_{QL}(t)$ and the translation $t_{QL}(t)$ from an initial convex set $\mathbb{B}_{QL}(t) = \{y : Gy \leq g\}$, where $G \in \mathbb{R}^{6\times3}$ and $g \in \mathbb{R}^{6\times1}$. In our implementation, the initial convex set $\mathbb{B}_{QL}(t)$ represents the vertical connection between the quadrotor and the payload through the prismatic joint. We suppose the transformation relation from convex set $\mathbb{B}_{QL}$ to $\mathbb{E}_{QL}$ could be written as,

$$\mathbb{E}_{QL}(t) = R_{QL}(t)\mathbb{B}_{QL} + P_{QL}(t), \tag{9}$$

where R_{QL} and P_{QL} are defined later in (15) and (14) respectively. The initial convex set $\mathbb{B}_{QL}(t)$ is defined as,

$$\mathbb{B}_{QL}(t) = \{y = (y_1, y_2, y_3) : -\varepsilon_x \leq y_1 \leq \varepsilon_x, -\varepsilon_y \leq y_2 \leq \varepsilon_y, 0 \leq y_3 \leq \varepsilon_z + l(t)\}. \tag{10}$$

We also assume that obstacles are convex compact sets with non-empty relative interior, such that each of them can be represented as

$$\mathbb{O} = \{y : Ay \leq b\}. \tag{11}$$

Then the collision-free trajectory generation with $d_{min} > 0$ as a desired safety margin can be formulated as

$$\begin{aligned} dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min} \Leftrightarrow \exists \lambda, \mu > 0 : \\ -g^T\mu + (AP_{QL}(x) - b)^T\lambda > d_{min}, \\ G^T\mu + R_{QL}(x)^T A^T \lambda = 0, \quad ||A^T\lambda|| < 1, \end{aligned} \tag{12}$$

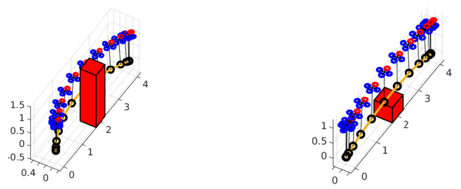
where λ and μ represent the dual variables associated with the obstacle $\mathbb{O}$. The detailed proof of (12) can be found [20, Theorem 1].

We could also use a slack variable s to represent the penetration distance if the safety margin is not needed to be strictly guaranteed, then the obstacle avoidance with minimum penetration could be formulated as

$$\begin{aligned} dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min} \Leftrightarrow \exists \lambda, \mu > 0, s \geq 0 : \\ -g^T\mu + (AP_{QL}(x) - b)^T\lambda > d_{min} - s, \\ G^T\mu + R_{QL}(x)^T A^T \lambda = 0, \quad ||A^T\lambda|| < 1. \end{aligned} \tag{13}$$

Introducing the slack variable s as the penetration distance in (13) would make this nonlinear optimization smoother. To minimize the penetration distance, we also need to add a term Ks with a large positive scalar K into the cost function (6).

Remark 3: The constraint $dist(\mathbb{E}_{QL}(t), \mathbb{O}) > d_{min}$ is non-differentiable, while the dual variables in the formulation



(a)

(b)

Fig. 3: Snapshots of simulation for obstacle avoidance task while navigating from the origin $[0,0,0]^T$ to a target payload position $[4,0,0]^T$. There is a cubic obstacle centered at $[2,0,0]^T$ with side length equal to $1m$ and no waypoints positions are predefined. The system avoids the obstacle laterally if the height of the obstacle is relatively high and vertically if the obstacle's height is smaller. Notice that the desired quadrotor attitude is generated based on the differential flatness.

(12) or (13) could make the constraint become differentiable and smooth, so that a variety of nonlinear optimization solvers could tackle this kind of optimization problem. Here we show that that if the controlled object (quadrotor-load system) and the obstacles are described by convex sets such as polytopes or ellipsoids (or can be decomposed into a finite union of such convex sets), then the collision avoidance constraints can be exactly and non-conservatively reformulated as a set of smooth non-convex constraints in (12).

The final question is how to express the rotation matrix R_{QL} and translation vector P_{QL} as functions of load position and load attitude. It's straight forward that,

$$P_{QL}(t) = x_L(t), \tag{14}$$

where the translation vector equals to the load position. After a pure translation $P_{QL}(t)$, our cuboid moves from $\mathbb{B}_{QL}(t)$ to $\mathbb{B}'_{QL}(t)$, presented in Figure 2. The rotation matrix R_{QL} represents a θ_r rotation along the axis a_r, colored as orange in Figure 2, where the rotation matrix could be expressed through the Rodrigues' rotation formula,

$$R_{QL} = I_3 + \sin(\theta_r)\hat{a}_r + (1-\cos(\theta_r))\hat{a}_r^2, \tag{15}$$

where $\hat{\cdot}$ represents the hat operator such that $\forall x, y \in \mathbb{R}^3, \hat{x}y = x \times y$, further, $\theta_r = \pi$, $a_r = \dfrac{q_0 + q}{||q_0 + q||}$ and $q_0 = [0,0,-1]^T$ represents the vertical load attitude of cuboid $\mathbb{B}'_{QL}(t)$.

5) Special Tasks: In this section, we show how our general formulation of the optimization problem can be particularized for specific tasks, such as waypoint navigation, payload throwing towards a desired target and passing through narrow windows.

a) Waypoint navigation: The waypoint navigation can be specified either on load position or quadrotor position. For the robustness of the algorithm, we do not use equality constraints but allow a small tolerance ε around the waypoint:

$$\begin{aligned} p_{wL,i} - \varepsilon \leq & x_L(t) \leq p_{wL,i} + \varepsilon \text{ (payload waypoints)}, \\ p_{wQ,i} - \varepsilon \leq & x_Q(t) \leq p_{wQ,i} + \varepsilon \text{ (quadrotor waypoints)}, \end{aligned} \tag{16}$$

where $x_Q(t) = x_L(t) - l(t)q(t)$. Moreover, it would also be possible to set the position of the load or the quadrotor for specific node, since it is fully defined in the system state.

b) Payload throwing: A payload throwing task can be achieved through a final release of payload after quadrotor-load navigation so that it free falls towards a given target. The

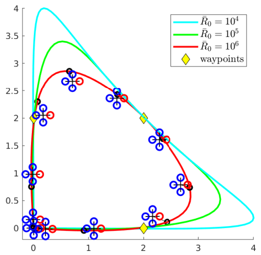


Fig. 4: Top-view of the desired payload position trajectories for navigating three waypoints without position error tolerance. In both trajectories, the quadrotor starts from hover at the origin and returns to the same position after flying through the payload waypoints. The three waypoints are $[0,2,0]^T$, $[2,2,0]^T$ and $[2,0,0]^T$, marked with diamonds. The three payload trajectories colored in blue, green and red are optimized with $\bar{R}_0 = 10^4, 10^5, 10^6$, respectively. Note that the scalar $\bar{R}_0$ is the weight parameter in the cost function in (6).

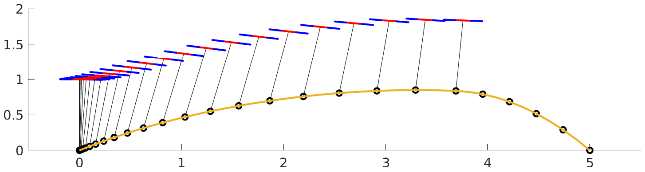


Fig. 5: x-z plane view of the desired payload trajectory for throwing task. According to the optimization results, the payload is released after being transported from the origin $[0,0,0]^T$ and reaches the desired target of $[5,0,0]^T$ after a projectile trajectory. Note that after the payload is released, only the snapshots of the payload are shown.

ballistic trajectory is well defined as function of the velocity at the moment of release. Assume x_r be the position, v_r the velocity of the payload in the world frame at the moment of release and x_t the target position. The released payload arriving at the given target x_t can be written as a constraint,

$$x_r + v_r t_r + [0,0,-\frac{1}{2}gt_r^2]^T = x_t, \tag{17}$$

where t_r represents the travel time from the release moment to the target position. In the payload throwing task, We don't need to preset the travel time t_r and the position of release x_r, but let the optimization find the most appropriate values.

c) Window task: In this task, the quadrotor and the load have to move through a window of height being smaller than the cable length, so that the quadrotor is forced to swing up the load to make it pass through the window, or to have hybrid mode transitions to achieve aggressive window avoidance maneuvers. Since a window-shaped obstacle can be considered as the combination of four obstacles together, the window task could be considered as a task to avoid multiple obstacles.

IV. Numerical Simulations

In this section, we present the numerical simulations of our path planning algorithm. Navigation of quadrotor-load system through waypoints, obstacle avoidance or other specifications.

A. Navigation with Obstacle Avoidance

Our algorithm is able to find dynamically-feasible trajectories for navigation task with obstacle avoidance. We select 50 nodes for this path planning formulation and the

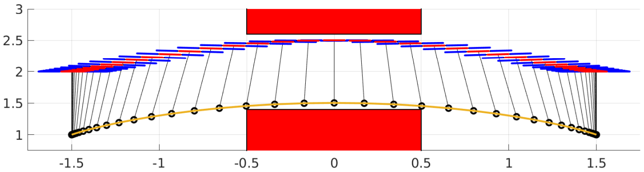


Fig. 6: Snapshots of maneuvering through window-like obstacles. With the definition of a signed distance, our quadrotor-load system could execute the window task. In this numerical simulation, the window's height and width are $1.2m$ and $1.0m$ respectively. Although the quadrotor-load system moves aggressively in this scenario, the optimization still generates an obstacle-free trajectory. Notice that the cable is always taut from the result of the optimization. Two lateral obstacles representing the window are excluded in the plotting but considered in the optimization formulation.

example shown in Figure 3a demonstrates path planning for obstacle avoidance. Snapshots show that the system initially hovers at the origin and navigates to another target payload position by avoiding the obstacle between them. Moreover, since our path planning problem formulation considers the energy efficiency, the quadrotor-load system will avoid the obstacle with different poses based on it. For example, when the height of this cuboid is decreased to $0.5m$, the quadrotor-load system will navigate this open space by avoiding the obstacle vertically instead of laterally, shown in Figure 3b.

B. *Waypoint Navigation*

The algorithm is able to find solutions for the waypoint navigation task with computational efficiency. Firstly, the quadrotor-load system is required to maneuver from hover at origin and back to the origin, while navigating through three waypoints without any tolerance ($\varepsilon = 0$). We have 100 nodes along the trajectories and 25 nodes are assigned for each segment of the trajectory. We show in Figure 4 that waypoint navigation task is well executed.

C. *Payload Throwing*

In order to throw the payload towards the target position after the navigation, the system needs to swing it up and move towards the target position. In addition, the quadrotor might need to lift the payload and make the horizontal velocity at release positive, to ensure the payload has sufficient time during the free fall projection. We don't need additional constraints to specify the movement, while letting the optimization find the appropriate time for payload release. In the simulation presented in Figure 5, the quadrotor lifts the payload a little bit from the initial position and swings it to pump energy before releasing the payload.

D. *Window Tasks*

1) Non-Guided Window Task: For a window whose height is bigger than the length of cable, the optimization could generate a dynamically feasible trajectory to pass through this kind of window, presented in Figure 6.

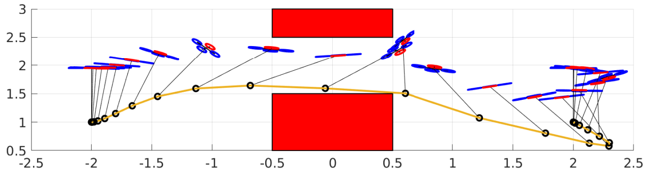
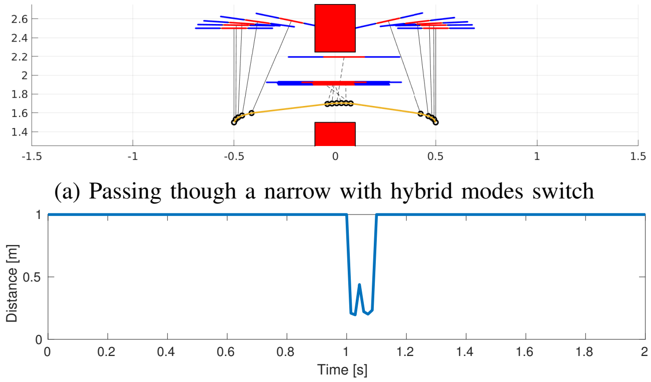


Fig. 7: X-Z view of snapshots of passing through a window with waypoint guidance. By defining one waypoint in the window area, our system could navigate through a narrow window. We only put a constraint that waypoint should lay inside the open space surrounded by the window. Notice that the cable is always taut from the result of the optimization. Two lateral obstacles representing the window are excluded in the plotting but considered in the optimization formulation.



(a) Passing though a narrow with hybrid modes switch

(b) Distance between quadrotor and payload

Fig. 8: Snapshots of passing through a window with hybrid modes. By tuning cost parameters, the optimization generates the hybrid modes to pass through the window. In Figure 8b, we could also easily observe the three hybrid modes during this maneuver, where the cables are marked dotted when they become slack. Two lateral obstacles representing the window are excluded in the plotting but considered in the optimization formulation.

2) State-Guided Window Task: For the narrow window, in order to decrease the computation time, we could add constraints at the specific node during the optimization. In Figure 7, we show that our system could navigate through a narrow window while additionally defining a waypoint in the open space surrounded by the window obstacle as guidance for the optimization. Precisely, we put 30 nodes for this path planning problem and put these additional constraints at node number 15: (i) load position is inside the open space $\mathbb{S} = \{y = (y_1, y_2, y_3) : y_1 \in [-0.5, 0.5], y_2 \in [-1, 1], y_3 \in [1.5, 2.5]\}$; (ii) load attitude angle α is bigger than α_{min}, where $\alpha_{min} = \pi/4$. Here load attitude angle α represents the angle between load attitude and vertical direction,

$$\cos(\alpha) = [0, 0, -1]^T \cdot q$$

3) Mode-Guided Window Task: An alternative method to pass through a narrow window is to exploit the hybrid modes of the system. To exploit it, we could increase the $\bar{R}_0$ to be relatively large values at several middle nodes and this leads the optimization to generate motions where the cable will be taut, slack and taut when passing through the narrow window. The snapshots of maneuvering though a narrow window are presented in Figure 8, where we can easily observe that instead of an aggressive swing in Figure 7, the quadrotor moves down to decrease the distance between itself and the suspended payload to avoid obstacles, where the cable becomes slack.

Description	Notation	Value
mass of the quadrotor	m_Q	0.825Kg
mass of the payload	m_L	0.065Kg
length of cable	l_0	1.097m
load attitude angle limitation	α_{max}	$\pi/2$
quadrotor dimension	$2\varepsilon_x/2\varepsilon_y$	0.255/0.255m
safety distance	d_{min}	5cm

TABLE II: Parameters for experiments

V. Experimental Results

A. Setup

A custom build quadrotor with a Raspberry Pi 3 based Navio-2 is used to test the trajectories generated using our planner. A ROS node on Raspberry Pi 3 runs the on-board attitude control at 500Hz. The ground station consists of a MacBook Pro running Ubuntu 18.04 and ROS melodic. A ROS node runs the position control at 200Hz on the ground control and communicates with the onboard control through WiFi. A 3D printed load is suspended using a string from the quadrotor. Parameters for the hardware setup are listed in Table II. Optitrack motion capture system is used to estimate the pose, velocity of the quadrotor and load. An Inertial Measurement Unit (IMU) on the Navio2 is used to estimate the body-attitude and body-rates.

The optimal trajectories generated using the techniques described in Sections III & IV are implemented in the experiments using the following control architecture. The optimal trajectory outputs the load trajectory and its derivatives $x_L^{(i)}$ and the corresponding quadrotor trajectory is computed through differential flatness. The control-loop is closed around this trajectory, i.e, we implement trajectory tracking control on the quadrotor. Note, this is due to our limitations in estimating the load-cable attitude and its angular velocity accurately, as the load attitude states are vital to achieve load position control [13].

For the path planning optimization problems, we use open source solver IPOPT [1] in Matlab with modeling language Yalmip [10]. In each experiment, we first generated an offline trajectory on a laptop, and this trajectory is sent to the ground control using a Python interface.

Finally, experimental snapshots for some scenarios are shown in Figure 9 and some experiments are demonstrated in the video at `https://youtu.be/e09RZOx_nZk`. The experiments illustrate obstacle avoidance with and without waypoint guidance. In the video, we also show more complex cases, such as generating a s-shaped trajectory to avoid wall-shaped obstacles in a closed space.

B. Computation Time

To verify the computation time advantage of our approach, we time the path planning optimization for each scenario. A MacBook Pro with Intel Core i7 (CPU 2.6GHz base speed) running Ubuntu 18.04 was used for the offline path planning computation. For each scenario mentioned in Section IV, the computation time are measured 10 times. For obstacle avoidance formulation, we firstly use the safe margin distance constraints in (12). The results are listed in Table III. Our algorithm outperforms state-of-the-art approaches and is over 10 times faster in hover to hover with no obstacles, payload throwing and waypoint navigation, where it takes generally less than 1s with our algorithm. In [5] authors need 10s to 30s for payload throwing and around 50s for triangle flight for waypoint navigation and in [15] more than 3000s are needed for some complex trajectories.

In Section III, we noted that the minimum-penetration formulation in (13) could be used as obstacle avoidance constraints, which makes the problem smoother. In this case, the computation time is further reduced about 10-20% when the obstacles are crowded. The results of computation time are also listed in Table III.

C. Initial Guess

In the results so far, we have not used any initial guess for solving our optimization problem. Our optimization formulation in (5) is a non-convex optimization problem, and hence computationally challenging to solve in general. It is well-known that the solution quality critically depends on the initial guess that is provided to the solvers, and that different initial guesses can lead to different (local) optima. Unfortunately, computing a good initial guess is often difficult and highly problem dependent; ideally, the initial guess should be obstacle-free and approximately satisfy the system dynamics. We propose the use of A* to get initial guesses for x_L and also its high order derivatives with collocation constraints in (5b). The initial guess of cable tension T and load attitude q could be calculated from the load acceleration and the distance $l(t)$ between the quadrotor and the payload. The initial guess of penetration distance s is assumed as the biggest dimension of the obstacle. The results with minimum-penetration method and A* as initial guess are also listed in Table III.

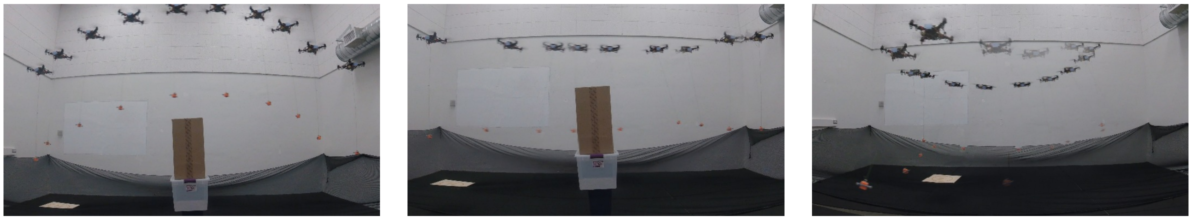
VI. Results and Discussion

A. Advantages

Our approach generates energy efficient, dynamically feasible trajectories, which satisfy the constraints of obstacle avoidance, waypoint navigation, and hybrid mode transitions. The algorithm could handle almost all the cases for the system containing a quadrotor with a suspended payload, without the need of excessive cost tuning. The guidance in the form of specifying waypoints is optional. Our problem formulation exploits the differential-flatness property with complementarity constraints for representing either slack or taut for the cable. As we don't use full dynamics as collocation constraints, and instead use differential flatness, this reduces the number of nonlinear constraints in the optimization problem and the computation time of motion planning is far more optimized compared to prior work.

B. Complexity and Robustness

Assume we have m nodes for our optimization and n obstacles (simply assumed as cuboids, then matrix A in (11) becomes 6×3) need to be considered in the collision-free trajectory generation, then we have $(3\times7+1+3+1)n+(6+3)nm = 26n+9mn$ variables to optimize, where $21n$ variables



(a) Obstacle avoidance without waypoint guidance

(b) Obstacle avoidance with lateral waypoint guidance

(c) Payload waypoint navigation with horizontal waypoint guidance

Fig. 9: Experimental snapshots of the quadrotor-with-load system executing successful obstacle avoidance and waypoint navigation maneuvers. The numbers of provided waypoints in Figure 9a, 9b, and 9c equal to zero, one and three, respectively.

Algorithm types		safety margin distance without initial guess		minimum-penetration without initial guess		minimum-penetration with A* for initial guess	
Experiments	nodes	mean [s]	std [s]	mean [s]	std [s]	mean [s]	std [s]
Hover to hover	50	0.46	0.22	0.47	0.19	0.53	0.21
Vertical obstacle avoidance	50	9.8	5.3	9.6	4.9	8.1	5.7
Horizontal obstacle avoidance	50	14.3	4.7	14.5	4.5	11.8	3.7
Guided obstacle avoidance	50	6.2	1.1	6.6	1.3	5.9	0.9
Payload throwing	50	1.4	1.1	1.3	1.0	1.3	0.6
Triangle flight for quadrotor waypoints	100	0.39	0.14	0.37	0.16	0.38	0.19
Triangle flight for payload waypoints	100	0.42	0.18	0.45	0.21	0.44	0.17
No-guided for window task	50	103.1	11.3	94	9.2	88.4	9.1
State-guided for window task*	30	470.2	47.0	442.6	41.9	390.4	38.2
Mode-guided for window task*	30	723.9	82.7	669.1	74.2	557.2	78.2

TABLE III: Computation time. The time metrics of using constraints with safety margin distance in (12) or with minimum-penetration distance in (13) for different tasks mentioned in Section IV are presented, respectively. The dual infeasibility tolerance for solver IPOPT needs to be set higher for state-guided and window-guided for window task. The standard deviation of computation time comes from fluctuations of CPU performance. We could observe that computation time is decreased when the obstacles become crowded when using minimum-penetration as obstacle avoidance constraints and initial guesses are also helpful to the even faster optimization. Notice that the computation time of A* is excluded in this table.

are for payload positions and their derivatives, $5n$ ones for load attitude, cable tension, quadrotor-load distance and $9mn$ ones for dual variables λ and μ in (12).

The problem becomes challenging if the environment becomes crowded with obstacles. We test the scalability of our planning algorithm with different numbers of obstacles. The environment setup and scalability results are presented in Table IV and Figure 10.

C. Limitations and Future Work

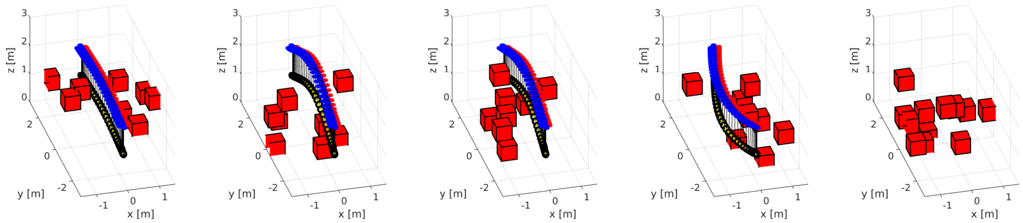
One limitation of our approach is that it is quite difficult to set physical limitations of quadrotor movement directly. For example, in order to set the maximum of thrust f or moment M of the quadrotor, a nonlinear constraint need to be formulated with the differential-flatness property. Without physical limitations on the quadrotor, it might make our trajectory too aggressive so that our controller could be unable to experimentally track it. A possible method to improve that is to assign some additional upper or lower bounds on the payload jerk, snap or other high order derivatives (which are linear constraints). Since we focus on purely optimization-based planning approach with collocation for quadrotor-load system, the complexity of our algorithm increases with the number of obstacles and may no longer be efficient when there are too many obstacles. This motivates us to combine the sampling based and optimization based methods together in the future work, which could reduce the computation time in the environments crowded with obstacles and keep the trajectories dynamically feasible.

VII. Conclusion

In this work, we proposed a model for parametrizing the quadrotor with a suspended payload using the differential-flatness property with complementarity constraints. With additional constraints from robot, task, environment and customized guidance, the path planning was formulated as a nonlinear optimization problem on collocation points. Our contributions are mostly three parts: We (i) model the system using high order derivatives of load position, load attitude, cable tension, and distance between quadrotor and payload, which fully describes the quadrotor-load system as it's differentially-flat with load position as flat outputs; (ii) formulate the path planning problem as a nonlinear optimization problem with direct collocation constraints and use complementarity constraints to represent the hybrid modes; (iii) propose a novel method for reformulating the obstacle avoidance from non-convex constraints into smooth ones with dual variables, by considering the quadrotor-load system as as a full-dimensional controlled object. Our trajectory generation problem outperforms the state-of-the-art in the computation time. We numerically and experimentally verified our approach, showing that our method could easily generalize to a variety of tasks, such as waypoint navigation, obstacle avoidance, payload throwing and window tasks. We also experimentally validated the majority of planning tasks.

number of obstacles	1	2	3	4	5	6	7	8	9	10	11	12
mean [s]	5.7	17.0	23.5	72.1	115.2	120.5	153.1	165.2	218.7	252.3	292.1	342.9
std [s]	1.5	8.3	9.2	45.0	60.1	53.7	62.1	86.5	79.7	83.8	102.5	136.7
min [s]	3.5	7.2	16.8	25.6	33.2	61.9	88.5	82.4	141.4	136.6	158.0	191.2
max [s]	8.1	32.0	40.5	172.3	223.5	226.4	272.8	289.9	332.0	428.9	569.1	652.7
failing cases	0	0	0	0	0	0	1	1	0	2	5	10

TABLE IV: Complexity and robustness analysis for path planning algorithm in a crowded environment with obstacles. The planning problem is setup with $[0,-2.5,1.0]$, $[0,2.5,1.0]$ are payload initial and final positions and no waypoints are predefined in the planner. In the closed space $\mathbb{S} = \{y = (y_1, y_2, y_3) : y_1 \in [-1.5, 1.5], y_2 \in [-1.5, 1.5], y_3 \in [0, 2]\}$, we randomly generate cubes with dimensions 0.5m by 0.5 by 0.5m as obstacles Moreover, we limit the payload system movement in the closed space $\mathbb{W} = \{y = (y_1, y_2, y_3) : y_1 \in [-1.5, 1.5], y_2 \in [-3, 3], y_3 \in [0, 2]\}$, which forces the system to navigate between obstacles and ensure the initial and the final configurations of the system are collision-free. For each trail, we sample the designed number of obstacles and test the path planning algorithm and we have 20 trails for each number of obstacles. Initial guesses with A* are used and minimum-penetration for obstacle avoidance is applied for optimization formulation. The failing cases comes from the number of iterations exceeding due to an almost infeasible environment to navigate as there are too many big obstacles and we have limited the movement of quadrotor-payload system in the closed space $\mathbb{W}$.



(a) (b) (c) (d) (e)

Fig. 10: Complexity and robustness analysis of planning algorithm: navigation task in a closed crowded environment $\mathbb{W}$ with payload position from $[0,-2.5,1.0]$ to $[0,2.5,1.0]$. The details about additional constraints and obstacle sampling are presented in the caption of Table IV. In Figure 10a, 10b, 10c, 10d, we show four successful trails with eight random obstacles, where the difficulty of obstacle avoidance could vary depending on their locations and that's why the computation time holds a relatively large standard deviation, presented in Table IV. We also visualize the case when the path planning algorithm fails, where twelve random obstacles are randomly sampled, presented in Figure 10e. From our observation, this failing case comes from the environment becomes too crowded in this case and there's no feasible trajectories in the closed environment.

References

[1] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[2] A. Caballero, M. Bejar, A. Rodriguez-Castaño, and A. Ollero, "Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms," *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, p. 1729881418770525, 2018.

[3] C. de Crousaz, F. Farshidian, and J. Buchli, "Aggressive optimal control for agile flight with a slung load," in *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Machine Learning in Planning and Control of Robot Motion*, 2014.

[4] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia., "Learning swing-free trajectories for uavs with a sus- pended load," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 4902-4909.

[5] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Robotics: Science and Systems*, 2017.

[6] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. García, "Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing," in *IEEE Conference on Decision and Control*, 2015, pp. 6718–6723.

[7] M. Kelly, "An introduction to trajectory optimization: how to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[8] S. Kim, S. Choi, and H. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4990–4995.

[9] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor uavs with a suspended payload," in *IEEE Conference on Decision and Control*, 2013, pp. 5510–5515.

[10] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[11] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 2691–2697.

[12] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems*, 2013.

[13] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *IEEE International Conference on Decision and Control*, 2013, pp. 2269-2274.

[14] 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 *IEEE International Conference on Robotics and Automation*, 2013, pp. 4873-4880.

[15] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE International Conference on Robotics and Automation*, 2015, 2216-2222.

[16] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired dynamic grasping and perching," *Bioinspiration & Biomimetics*, vol. 9, no. 2, pp. 025 010–025 024, June 2014.

[17] G. Wu and K. Sreenath, "Geometric control of multiple quadrotors transporting a rigid-body load," in *IEEE International Conference on Decision and Control*, 2014, pp. 6141-6148.

[18] J. Zeng, P. Kotaru, and K. Sreenath, "Geometric control and differential flatness of a quadrotor uav with load suspended from a pulley," in *American Control Conference*. IEEE, 2019, pp. 2420–2427.

[19] J. Zeng and K. Sreenath, "Geometric control of a quadrotor with a load suspended from an offset," in *American Control Conference*. IEEE, 2019, pp. 3044–3050.

[20] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4327–4332.