

# AutoTrans: A Complete Planning and Control Framework for Autonomous UAV Payload Transportation

Haojia Li<sup>2</sup>, Haokun Wang<sup>2</sup>, Chen Feng<sup>2</sup>, Fei Gao<sup>\*3,4</sup>, Boyu Zhou<sup>\*1</sup>, and Shaojie Shen<sup>2</sup>

**Abstract**—The robotics community is increasingly interested in autonomous aerial transportation. Unmanned aerial vehicles with suspended payloads have advantages over other systems, including mechanical simplicity and agility, but pose great challenges in planning and control. To realize fully autonomous aerial transportation, this paper presents a systematic solution to address these difficulties. First, we present a real-time planning method that generates smooth trajectories considering the time-varying shape and non-linear dynamics of the system, ensuring whole-body safety and dynamic feasibility. Additionally, an adaptive NMPC with a hierarchical disturbance compensation strategy is designed to overcome unknown external perturbations and inaccurate model parameters. Extensive experiments show that our method is capable of generating high-quality trajectories online, even in highly constrained environments, and tracking aggressive flight trajectories accurately, even under significant uncertainty. We plan to release our code to benefit the community.

**Index Terms**—Aerial Systems; Applications; Motion and Path Planning; Robust/Adaptive Control

## I. INTRODUCTION

RECENTLY, robotics researchers have shown a growing interest in aerial transportation, driven by the need for efficient, safe, and flexible transportation in extensive scenarios, e.g., disaster relief, logistics, and agriculture. A typical aerial transportation system employs an active manipulator [1] or a cable [2] attached to the aerial vehicle. Compared to active manipulators, unmanned aerial vehicles with suspended payload systems (UAV payload systems) are preferred as it needs no additional actuation, which reduces mechanical complexity and weight, allowing a higher load budget [3]. Furthermore, the UAV's attitude dynamics are less affected by the payload than a manipulator, allowing improved attitude control and potentially more agile maneuvers [4].

Although UAV payload systems offer several benefits, they also pose significant challenges. One major issue is the lack of

Manuscript received: April, 24, 2023; Revised: July, 13, 2023; Accepted: August, 9, 2023. This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by The Research Grants Council General Research Fund (RGC GRF) project RMGS20EG20

<sup>1</sup>School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China.

<sup>2</sup>Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

<sup>3</sup>State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China.

<sup>4</sup>Huzhou Institute, Zhejiang University, Huzhou, China.

\*Corresponding authors: Boyu Zhou, Fei Gao

Email: hlied@ust.hk, zhoubu23@mail.sysu.edu.cn

Digital Object Identifier (DOI): see top of this page.

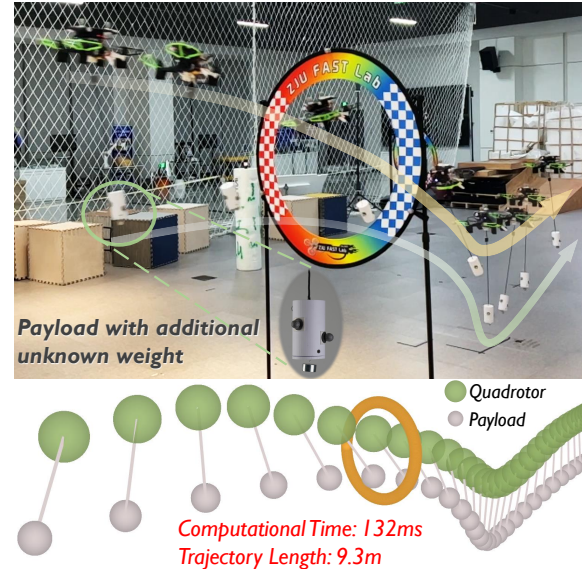


Fig. 1. A quadrotor with a suspended payload system is passing a narrow circle. An unknown weight (50g) hangs below the payload. The trajectory generation time is 132ms with 9.3m length on quadrotor's onboard computer.

a motion planner that meets three essential criteria: real-time planning, dynamic feasibility, and whole-body safety. Particularly, the algorithm must efficiently account for the system's time-varying shape and enable safe yet agile maneuvers in complex unknown environments. Some existing methods [3], [5], [6] avoid collisions by using the system's bounding sphere or simplified obstacle shapes, which are either conservative or impractical in complex scenes. Although some works consider the exact shape of the system [7], they are not real-time capable due to the complex system model.

Another challenge is the lack of a robust and adaptive controller that accurately tracks trajectories under significant uncertainty. External disturbances such as wind or rotor drag, along with inaccuracies in modeling (e.g., assuming that the cable is attached to UAV's center of mass), can all affect control accuracy. Moreover, extra uncertainty would be introduced for more flexible transportation. Specifically, it is desirable that the system can transport various payloads adaptively without knowing the precise weight of a payload. This enables efficient load-and-fly transportation but also introduces uncertain parameters into the system model.

To address these challenges, this paper presents a systematic solution for fully autonomous UAV payload transportation. Firstly, we introduce a real-time planning solution to gen-

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

erate smooth trajectories. We exploit the differential flatness property of UAV payload system to derive the system's states, thereby converting the whole-body safety and dynamic feasibility constraints into continuous and differentiable penalty functions. The problem is then formulated as an unconstrained optimization problem that can be efficiently solved by a spatial-temporal trajectory optimizer. To deal with strong nonlinearity, we further adopt a kinodynamic search front-end to find promising initial guesses. Secondly, to overcome unknown external perturbations as well as inaccurate model parameters, we propose an adaptive NMPC with a hierarchical disturbance compensation strategy. An online external force estimator is designed, which compensates for disturbance forces acting on both UAV and payload. We further apply Incremental Nonlinear Dynamic Inversion (INDI) to the angular velocity controller, compensating unknown torque. The proposed method can generate high-quality trajectories online even in highly constrained environments, such as the one in Figure 1. Meanwhile, our controller tracks trajectories robustly and accurately, even with additional unknown payloads.

In summary, our work has three contributions:

- 1) A real-time differential flatness-based spatio-temporal planning method, which generates smooth trajectories that ensure whole-body safety and dynamic feasibility.
- 2) A robust NMPC controller with a hierarchical disturbance compensation strategy, resisting perturbations and inaccurate model parameters.
- 3) Extensive simulation and real-world experiments for validating our methods. We release our code, including planning, controller, and simulation<sup>1</sup>.

## II. RELATED WORK

Numerous planning and control methods have been proposed to address the unique challenges presented by UAVs with suspended payloads. As for trajectory generation, [2], [8], [9] generated minimum swing trajectory, but did not consider obstacle avoidance. [10] used RRT, which can not ensure energy efficiency. In [11], mixed integer quadratic programming was used to generate narrow-gap-passing trajectories and demonstrated load throwing. [3], [7] achieved maneuvers with complementary constraint. While the referred work demonstrated promising results, it was hard to generate collision-free trajectories in real time. Alternatively, [5], [6], [12] achieved real-time trajectory generation, but they all conservatively assumed that obstacles are ellipsoid, which may not generalize well to complex and unstructured environments.

As for controllers, some researchers [4], [13] proposed geometric controllers, which employ a coordinate-free dynamic model derived from the properties of Lie groups with a cascade structure. However, this type of controller was unable to track both the UAV and the suspended payload simultaneously. Some approaches, such as [14]–[16], focus on minimizing the swing angle of the suspended payload to achieve stabilization. However, these approaches tend to restrict the range of motion of the payload, thereby limiting its ability to perform high-speed agile maneuvers. All the above

methods did not consider the disturbance or the uncertainty, which is essential for executing tasks outdoors, e.g., medical rescue. [17] did consider wind disturbances with a complex model, but this approach had strong assumptions about the environment and did not demonstrate its performance in real-world scenarios.

Despite the significant progress, there are still several challenges that need to be addressed to apply UAVs with suspended payloads in the real world. These include developing methods that can handle complex and cluttered environments, improving the robustness of control methods to disturbances and uncertainties, and reducing the computational complexity of planning and control methods to enable real-time operation.

## III. PRELIMINARIES

This section introduces the dynamics model and differential flatness for a quadrotor with a cable-suspended payload.

### A. Nominal System Dynamics

Quadrotor with cable-suspended payload has two modes depending on the cable tension. One is the taut mode, and the other one is the slack mode. In transportation tasks, we desire the UAV payload system to be energy-efficient. However, the energy consumption of hybrid slack and taut mode is typically higher than always-taut mode due to the large acceleration required to compensate when the cable becomes taut again [3]. Therefore, we assume the cable is always taut in this paper. The whole system can be considered as a rigid body, and the cable can be simplified as a massless rob.

The system's configuration is illustrated as Figure 3. Similar to [4], [18], the system dynamics is formulated as

$$(m_Q + m_L)(\ddot{\mathbf{x}}_L + g\mathbf{e}_z^w) = (\boldsymbol{\rho} \cdot f\mathbf{R}\mathbf{e}_z^w - m_Q l(\dot{\boldsymbol{\rho}} \cdot \dot{\boldsymbol{\rho}}))\boldsymbol{\rho}, \quad (1)$$

$$m_Q l \ddot{\boldsymbol{\rho}} + m_Q l(\dot{\boldsymbol{\rho}} \cdot \dot{\boldsymbol{\rho}})\boldsymbol{\rho} = \boldsymbol{\rho} \times (\boldsymbol{\rho} \times f\mathbf{R}\mathbf{e}_z^w), \quad (2)$$

$$\mathbf{x}_Q = \mathbf{x}_L - l\boldsymbol{\rho}, \quad (3)$$

$$\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}, \quad \boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}, \quad (4)$$

where  $\mathbf{x}_L$  and  $\mathbf{x}_Q$  are center position of the payload and quadrotor.  $\ddot{\mathbf{x}}_L$  is payload's acceleration.  $\boldsymbol{\rho}$  is cable direction (a unit vector from quadrotor to payload center).  $l$  is cable length.  $\mathbf{J}$  is the quadrotor's inertia matrix.  $\boldsymbol{\omega}$  is angular velocity of the quadrotor.  $f$  is quadrotor thrust.  $\mathbf{R}$  is rotation of quadrotor.  $\boldsymbol{\tau}$  is moment vector of quadrotor.  $\mathbf{e}_z^w$  is the z-axis of world frame.

### B. Differential Flatness

Differential flatness allows us to express the system's state and control inputs in terms of these flat outputs, reducing the optimization problem's dimension and enabling efficient trajectory planning and control [19]. In the case of a quadrotor with a cable-suspended payload, this property holds when the cable is taut. The flat outputs  $\mathcal{Z} = [\mathbf{x}_L, \psi]$  for this system are payload's position  $\mathbf{x}_L$  and the quadrotor's yaw angle  $\psi$  [2].

$$m_L(\ddot{\mathbf{x}}_L + g\mathbf{e}_z^w) = -f_c\boldsymbol{\rho}, \quad \boldsymbol{\rho} = -\frac{(\ddot{\mathbf{x}}_L + g\mathbf{e}_z^w)}{\|(\ddot{\mathbf{x}}_L + g\mathbf{e}_z^w)\|_2^2} \quad (5)$$

$$f\mathbf{R}\mathbf{e}_z^w = m_Q(\ddot{\mathbf{x}}_Q + g\mathbf{e}_z^w) - f_c\boldsymbol{\rho}. \quad (6)$$

<sup>1</sup><https://github.com/SYSU-STAR/AutoTrans>

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

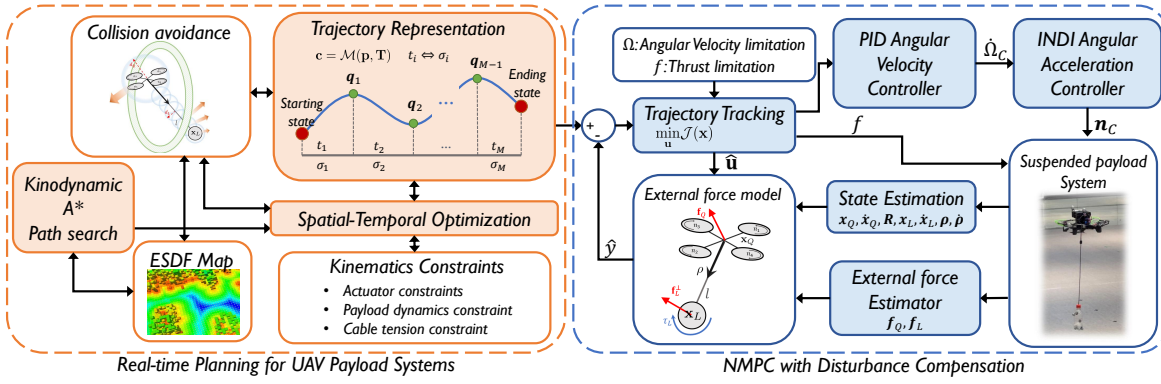


Fig. 2. System Overview. Kinodynamic A\* obtains the initial path, and spatio-temporal optimization generates a smooth, feasible, collision-free trajectory. NMPC with force estimator and INDI handles disturbances during tracking.

The position  $\mathbf{x}_Q$  and its derivatives for the quadrotor can be determined using equation (3). The z-axis of the quadrotor can be obtained from equation (5) and equation (6). As  $\mathbf{x}_Q$  and  $\psi$  are known along with their derivatives, the remaining states and control inputs can be obtained using the conventional quadrotor method, as described in [20].

### C. System Overview

The overall architecture is depicted in Figure 2. The kinodynamic A\* obtains a collision-free path as promising initial guesses. Then, a spatio-temporal trajectory optimization method is utilized to generate a smooth, dynamically feasible, and collision-free trajectory. The generated polynomial trajectory is tracked by a nonlinear model predictive controller (NMPC) incorporating a hierarchical disturbance compensation strategy that includes INDI control for compensating angular accelerations and a force estimator for compensating external forces.

## IV. REAL-TIME PLANNING FOR UAV PAYLOAD SYSTEMS

In this section, we introduce the differential flatness-based spatio-temporal optimization formulation.

### A. Nonlinear Optimization Construction

We utilize  $M$  pieces of  $D$ -dimension polynomial with degree  $N = 2s - 1$  to represent the whole trajectory. Each piece is expressed as  $\mathcal{Z}_i(t) = \mathbf{c}_i^T \beta(t)$ ,  $\beta(t) = [1, t_i, \dots, t_i^{2s-1}]^T$ . In our case, we choose  $D = 3$  and  $N = 7$  to represent the payload position  $\mathbf{x}_L$  in flat output space.

The objective function minimizes the control energy involving a time regularization written as (7a), where  $Q_J$  and  $\lambda_T$  are the weight of energy and time penalty, respectively.

$$\min_{\mathbf{c}, \mathbf{T}} J(\mathbf{c}, \mathbf{T}) = \underbrace{\sum_{i=1}^M \int_0^{t_i} \mathcal{Z}_i^{(s)}(t) Q_J \mathcal{Z}_i^{(s)}(t) dt}_{\text{Energy Cost}} + \underbrace{\lambda_T \|\mathbf{T}\|}_{\text{Time Cost}} \quad (7a)$$

$$s.t. \quad \mathcal{Z}_1^{[s-1]}(0) = \mathbf{z}_0, \quad \mathcal{Z}_M^{[s-1]}(t_M) = \mathbf{z}_f, \quad (7b)$$

$$\mathcal{Z}_i^{[s-1]}(t_i) = \mathcal{Z}_{i+1}^{[s-1]}(0), \quad (7c)$$

$$t_i > 0, t_i \in \mathbf{T}, \quad (7d)$$

$$C_b(\mathcal{Z}_i(t), \mathcal{Z}_i^{(1)}(t), \dots, \mathcal{Z}_i^{(s)}(t)) \leq 0, b \in \mathcal{G}. \quad (7e)$$

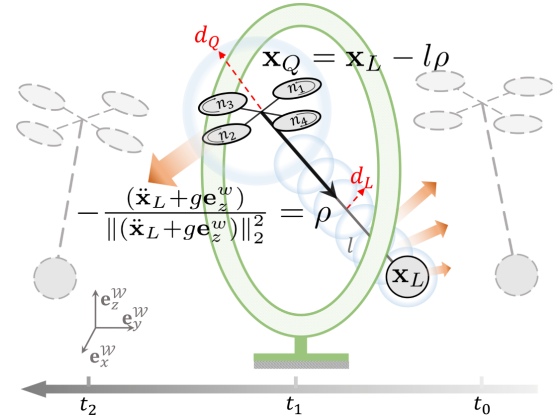


Fig. 3. Configuration of the UAV payload system. The system shape is simplified as a bunch of bubbles. When bubbles collide with obstacles, the ESDF field pulls the bubbles away from the obstacles.

The dynamics constraints and collision avoidance are expressed as violation function  $C_b$  (7e) where  $\mathcal{G}$  is the set of all constraints. The equation (7b) and (7c) denote boundary conditions and continuous constraints of trajectory separately where  $\mathcal{Z}_i^{(s)}$  is  $s$  order derivative on flat output  $\mathcal{Z}_i$ , and  $\mathcal{Z}_i^{[s-1]}$  is all derivative from 0-order to  $s - 1$  order.

To deform the spatio-temporal trajectory, we utilize the MINCO representation [19]. This approach allows us to optimize the time and shape of the trajectory simultaneously while naturally satisfying the boundary conditions (7b) and continuous constraints (7c). Instead of direct parametrization by coefficients,  $\mathbf{c}$ , MINCO represents a polynomial spline with initial, terminal states, intermediate points  $\mathbf{p}$  and time allocation  $\mathbf{T}$ . The spline guarantees to pass through  $\mathbf{p}$  within a specific time  $\mathbf{T}$ . We transfer the optimization space by  $\mathbf{c} = \mathcal{M}(\mathbf{p}, \mathbf{T})$  to reduce the optimization variables and improve numerical stability, where  $\mathcal{M}$  expresses a smooth and linear-complexity map. Furthermore,  $\mathcal{M}$  can be treated as a differentiable function [19].

### B. Whole-body Collision Avoidance

The payload transportation system comprises multiple rigid bodies, leading to variations in its shape with changing state. Representing the entire system as a single point reduces the solution space, making it difficult for the system to pass through narrow gaps. To overcome this, we approximate the

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

shape of the payload transportation system using a set of spherical bubbles (as shown in Figure 3). The system position and shape are decided by the quadrotor and payload position. Based on the differential flatness property, (3) and (5), the system position and shape are determined by payload position  $\mathbf{x}_L$  and acceleration  $\ddot{\mathbf{x}}_L$ . We ensure that the distance between these bubbles and obstacles is greater than the safe distance  $d_s$ . We formulate collision avoidance constraints, as (9),

$$d(l_s) = d_s - \mathbf{E}(\mathbf{x}_L - l_s \boldsymbol{\rho}), \quad (8)$$

$$C_{bc} = \frac{1}{N} \sum_{j=0}^N \max \left[ \left( d \left( \frac{j}{N} l \right) \right)^3, 0 \right] \leq 0, \quad (9)$$

where  $N$  is the number of bubbles. The  $d_s$  can be varying according to system shape with different  $j$ . The safe distance of payload and cable is  $d_L$ . The  $d_Q$  represents the quadrotor's safe distance. We maintain a discretization euclidean signed distance field (ESDF)  $\mathbf{E}$  for querying the shortest distance to obstacles with [21]. When the quadrotor and payload are close to the obstacle, the gradient will push the shape out of the obstacle like the orange arrows in Figure 3. In practice, trilinear interpolation is utilized to improve the precision of distance and gradient.

### C. Dynamic Constraints

To ensure the feasibility of the trajectory, we impose dynamic constraints on the system using differential flatness.

1) *Quadrotor actuator constraints:* We consider hardware limitations by limiting the maximum thrust and tilt angle of the quadrotor. From the flatness theorem in section III-B, the thrust and the quadrotor z-axis is written as (10),

$$f \mathbf{R} \mathbf{e}_z^w = (m_Q + m_L) (\ddot{\mathbf{x}}_L + g \mathbf{e}_z^w) - m_Q l \dot{\boldsymbol{\rho}}. \quad (10)$$

We limits the thrust within the boundaries  $f_l$  and  $f_u$  by

$$C_{bf}(\mathbf{z}) = \left( f - \frac{f_u + f_l}{2} \right)^2 - \left( \frac{f_u - f_l}{2} \right)^2 \leq 0. \quad (11)$$

Limiting the thrust benefits quadrotor attitude stabilization and avoiding unfeasible trajectory.

The tilt angle of the quadrotor should not exceed the limitation to ensure dynamic feasibility and prevent the quadrotor from flipping. We limit the tilt angle of the quadrotor to  $\theta_{max} = 60^\circ$ . The tilt angle constraint is defined as equation (12),

$$C_{b\theta}(\mathbf{z}) = \cos(\theta_{max}) - \mathbf{e}_z^w T \mathbf{R} \mathbf{e}_z^w \leq 0. \quad (12)$$

2) *Payload dynamics constraint:* We restrict the payload velocity and acceleration in a reasonable magnitude by (13),

$$C_{bp}(\mathbf{z}) = \begin{cases} \|\dot{\mathbf{x}}_L\|_2^2 - v_{max}^2 \leq 0 \\ \|\ddot{\mathbf{x}}_L\|_2^2 - a_{max}^2 \leq 0 \end{cases}. \quad (13)$$

3) *Cable tension constraint:* As discussed in III-A, We assume the cable is always taut. From equation (5), the cable tension is expressed as  $f_c = \|m_L (\ddot{\mathbf{x}}_L + g \mathbf{e}_z^w)\|_2$ . As we project the force on the z-axis, the cable tension constraint can be written as equation (14),

$$C_{bf_c}(\mathbf{z}) = \epsilon - \ddot{\mathbf{x}}_L \mathbf{e}_z^w - g \leq 0, \quad (14)$$

where  $\epsilon$  is a small positive number to avoid the numerical precision problem.

### D. Unconstrained Optimization

To speed up optimization, we hope to eliminate all the constraints and transfer the original problem into an unconstrained problem. Considering the time of each segment, it should be strictly positive as (7d). We introduce an unconstrained virtual time  $\sigma = [\sigma_1, \dots, \sigma_i, \dots, \sigma_n]$  and use a diffeomorphism map [19] to map  $\sigma_i \in \mathbb{R}$  to the real duration  $t_i \in \mathbb{R}^+$ .

We adopt the integration to compute the constraint violation to reduce the infinite number of constraints to a finite number,

$$S = \sum_{i=1}^M \int_0^{t_i} L_1(C_b(\mathbf{z})) dt, \quad (15)$$

where  $L_1(\cdot)$  is smooth penalty function [22]. In summary, the trajectory optimization problem can be redefined as an unconstrained optimization problem shown as equation (16),

$$\min_{\mathbf{q}, \sigma} J(\mathbf{q}, \sigma) = \sum_{i=1}^M \int_0^{t_i} \mathcal{Z}_i^{(s)}(t) Q_J \mathcal{Z}_i^{(s)}(t) dt + \lambda_T \|\mathbf{T}\| + \lambda_s S, \quad (16)$$

where  $\lambda_s$  are the weight of constraint penalty.

### E. Kinodynamic A\* for initial guesses

Directly optimizing the trajectory of a UAV payload system is challenging due to its nonlinear dynamics and time-variant shape, which impose several nonlinear constraints. Therefore, having a promising initial guess is of vital importance to prevent getting stuck in infeasible local minima and achieve faster convergence during optimization. Our planning method originates from kinodynamic hybrid A\* [23]. Different from the original method regarding the quadrotor as a point, we consider the robot system's geometry shape varying in different states leveraging the differential flatness.

For hybrid A\*, we use the payload acceleration  $\ddot{\mathbf{x}}_L$  as the control input  $u$ , and represent the trajectories using 2-order time-parameterized polynomials as primitives. Each primitive is evaluated and only the collision-free ones are added to the open list. The collision checking method is similar to IV-B, where we check bubbles in the grid map.

The cost function is defined as,

$$C(u, T) = \int_0^T \|u\|_2^2 dt + \lambda T. \quad (17)$$

$u := \ddot{\mathbf{x}}_L$ ,  $\lambda$  is the weight of time, and  $T$  is path time. The heuristic function is the same with [23].

## V. NMPC WITH DISTURBANCE COMPENSATION

In order to improve the tracking accuracy and enhance the robustness, an external force compensation strategy is employed in the Nonlinear Model Predictive Control (NMPC) for UAV payload systems to resist external forces and model uncertainties. All the perturbations can be seen as forces and torques acting on the UAV and payload. We design a force estimator cooperating with the NMPC to cancel the

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

disturbance forces. To implicitly compensate for the torques, an Incremental Nonlinear Dynamic Inversion (INDI) method is adopted on the angular velocity controller. Our compensation mechanism can resist various disturbances, such as wind, aerodynamic drag, and even weight changes.

**A. NMPC with External Force**

We extend the nominal system dynamics presented in the previous section (III-A) to include an external force term acting on the UAV-payload system. The system dynamics with external force is given by the following:

$$\ddot{\mathbf{x}}_Q = f_c \boldsymbol{\rho} + \frac{f \mathbf{R} \mathbf{e}_z^w + \mathbf{f}_Q}{m_Q} - g \mathbf{e}_z^w, \quad (18)$$

$$\ddot{\mathbf{x}}_L = -f_c \boldsymbol{\rho} + \frac{\mathbf{f}_L}{m_L} - g \mathbf{e}_z^w, \quad (19)$$

$$f_c = \frac{m_L l (\dot{\boldsymbol{\rho}} \cdot \dot{\boldsymbol{\rho}})}{(m_Q + m_L)} + \frac{\boldsymbol{\rho} \cdot (m_Q \mathbf{f}_L - m_L (f \mathbf{R} \mathbf{e}_z^w + \mathbf{f}_Q))}{m_Q (m_Q + m_L)}, \quad (20)$$

$$\ddot{\boldsymbol{\rho}} = \boldsymbol{\rho} \times \left( \boldsymbol{\rho} \times \left( \frac{f \mathbf{R} \mathbf{e}_z^w + \mathbf{f}_Q}{m_Q l} - \frac{\mathbf{f}_L}{m_L l} \right) \right) - (\dot{\boldsymbol{\rho}} \cdot \dot{\boldsymbol{\rho}}) \boldsymbol{\rho}, \quad (21)$$

$$\dot{\boldsymbol{\rho}} = \frac{d\boldsymbol{\rho}}{dt}, \quad \dot{\mathbf{x}}_L = \frac{d\mathbf{x}_L}{dt}, \quad \dot{\mathbf{x}}_Q = \frac{d\mathbf{x}_Q}{dt}, \quad \dot{\mathbf{R}} = \mathbf{R} \hat{\boldsymbol{\omega}}, \quad (22)$$

where the  $f_Q$  and  $f_L$  denote the external force acting on the UAV and payload, respectively.  $f_c$  expressed by (20) is the tension on the cable and  $\boldsymbol{\rho}$  is the cable direction. The UAV and payload dynamics are given by (18) and (19).

The NMPC problem is written as,

$$\min_{\mathbf{u}} \mathcal{J} = \frac{1}{2} \sum_{k=1}^N \left( \|\tilde{\mathbf{x}}(k)\|_{\mathbf{Q}_k}^2 + \|\tilde{\mathbf{u}}(k)\|_{\mathbf{H}_k}^2 \right) + \frac{1}{2} \|\tilde{\mathbf{x}}(N+1)\|_{\mathbf{Q}_e}^2, \quad (23a)$$

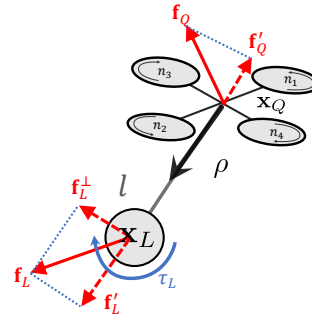
$$s.t. \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \quad (23b)$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}, \quad \mathbf{x}(N+1) \in X_f, \quad (23c)$$

where  $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \mathbf{x}_r(k)$ ,  $\tilde{\mathbf{u}}(k) = \mathbf{u}(k) - \mathbf{u}_r(k)$ .  $\mathbf{x}_r$  and  $\mathbf{u}_r$  are the reference state and control input derived from trajectories. Control input  $\mathbf{u} = [f_c, \boldsymbol{\omega}_c]^T$  includes command thrust  $f_c$  and command angular velocity  $\boldsymbol{\omega}_c$ . System state is  $\mathbf{x} = [\mathbf{x}_Q, \mathbf{x}_L, \dot{\mathbf{x}}_Q, \dot{\mathbf{x}}_L, \boldsymbol{\rho}, \dot{\boldsymbol{\rho}}, \mathbf{R}]^T$ .  $\mathbf{u}_{min}$  and  $\mathbf{u}_{max}$  are constraints of control inputs imposed by the physical and operational limitations. The function  $f(\mathbf{x}(k), \mathbf{u}(k))$  is a discrete version of system dynamics (equation (18)-(22)).  $N$  is the prediction horizon.  $\mathbf{Q}_k$  and  $\mathbf{H}_k$  are the state and control input weighting matrices at  $k$  step.  $\mathbf{Q}_e$  is the terminal state weighting matrix. The  $\mathbf{x}(N+1) \in X_f$  is the terminal constraint. Note that, we apply exponential decay on the weight matrix  $\mathbf{Q}$  and  $\mathbf{H}$ , written as (24), to improve the controller robustness, avoid ill condition [24]. Intuitively, the decay weight prioritizes the error at the current time over future times [24], [25], reducing the impact of the estimation noise and model uncertainty. The  $b_x$  and  $b_u$  decide the decay speed. To solve the NMPC problem, we use the real-time iteration (RTI) scheme with Gauss-Newton hessian approximation. We implement the code with ACADO [26].

$$\mathbf{Q}_k = \exp\left(-\frac{k}{N} b_x\right) \mathbf{Q}, \quad \mathbf{H}_k = \exp\left(-\frac{k}{N} b_u\right) \mathbf{H}. \quad (24)$$

**B. External Force Estimator**



(a) External force model (b) Mount point mismatch

Fig. 4. (a) shows the external forces acting on the system. The component force along the direction of the cable  $\mathbf{f}'_Q$  and  $\mathbf{f}'_L$  would cancel each other. Another force  $\mathbf{f}_L^\perp$  can cause the moment  $\tau_L$  acting on the cable. (b) shows the mount point mismatch. In practice, It is hard to satisfy that the mount point is exactly the same as the center of mass.

Obtaining accurate external force is key to the proposed controller. We propose an effective online estimator to obtain the external force.

The forces are illustrated in Figure. 4(a). Adding equation (18) to (19), we can obtain

$$\mathbf{f}_Q + \mathbf{f}_L = m_Q (g \mathbf{e}_z^w + \ddot{\mathbf{x}}_Q) + m_L (g \mathbf{e}_z^w + \ddot{\mathbf{x}}_L), \quad (25)$$

where the  $\ddot{\mathbf{x}}_Q$  and  $\ddot{\mathbf{x}}_L$  are the acceleration of UAV and payload which can be easily measured by IMU.

From (18), we have

$$\boldsymbol{\rho} \times \mathbf{f}_Q = \boldsymbol{\rho} \times (f \mathbf{R} \mathbf{e}_z^w - m_Q g \mathbf{e}_z^w - m_Q \ddot{\mathbf{x}}_Q). \quad (26)$$

The  $f \mathbf{R} \mathbf{e}_z^w$  is the thrust of the UAV in the world frame. The thrust  $f$  can be obtained through the angular velocity of each propeller,  $\mathbf{n}$  (measured by bi-directional Dshot ESC), by the propeller model of quadrotor like

$$\mathbf{G}_1 = \begin{bmatrix} k_f & k_f & k_f & k_f \\ a_y k_f & -a_y k_f & -a_y k_f & a_y k_f \\ -a_x k_f & -a_x k_f & a_x k_f & a_x k_f \\ -k_m & k_m & -k_m & k_m \end{bmatrix}, \quad (27)$$

$$\mathbf{G}_2 = \begin{bmatrix} \mathbf{0}_{3 \times 4} & \begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} \\ I_p & -I_p \end{bmatrix}, \quad \begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{G}_1 \mathbf{n}^{\circ 2} + \mathbf{G}_2 \dot{\mathbf{n}}, \quad (28)$$

where  $a_x = a \cos \beta$ ,  $a_y = a \sin \beta$ ,  $a$  is the arm length of the quadrotor,  $\beta$  is the angle between the quadrotor  $x$  axis and the propeller,  $k_f$  and  $k_m$  indicate thrust coefficient and torque coefficient respectively,  $I_p$  is the inertia of the propeller,  $\boldsymbol{\tau}$  is the torque of the quadrotor, and  $\circ$  represents the Hadamard power.

It should be noted that when the payload is kept at a fixed angle to the UAV, such as the hovering state, equations (18) and (19) are underdetermined for  $\mathbf{f}_Q$  and  $\mathbf{f}_L$ , leading to infinite solutions. Physically,  $\mathbf{f}_L$  and  $\mathbf{f}_Q$  can be resolved into components along and perpendicular to the cable. However, the component force along the cable cancels out, leading to infinite solutions. To avoid this, a nominal combination of forces for  $\mathbf{f}_L$  and  $\mathbf{f}_Q$  is found. We assume that  $\mathbf{f}_L$  is

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

perpendicular to the cable and creates a moment  $\tau_L$  on it. Therefore, we have  $\mathbf{f}_L \cdot \boldsymbol{\rho} = 0$  and  $\boldsymbol{\tau}_L = \boldsymbol{\rho} \times \mathbf{f}_L$ .

We formulate the estimation problem as a nonlinear least squares problem. Residuals are defined as

$$r_1^i = \mathbf{f}_Q^i + \mathbf{f}_L^i - m_Q (g\mathbf{e}_z^w + \ddot{\mathbf{x}}_Q^i) + m_L (g\mathbf{e}_z^w + \ddot{\mathbf{x}}_L^i), \quad (29)$$

$$r_2^i = \boldsymbol{\rho}^i \times \mathbf{f}_Q^i - \boldsymbol{\rho}^i \times (f^i \mathbf{R}^i \mathbf{e}_z^w - m_Q g\mathbf{e}_z^w - m_Q \ddot{\mathbf{x}}_Q^i), \quad (30)$$

$$r_3^i = \mathbf{f}_L^i \cdot \boldsymbol{\rho}^i, \quad (31)$$

where the  $i$  denotes each measurement. For robustness, we utilize a sliding window to estimate the forces in our implementation. Assuming that external forces vary insignificantly within the sliding window, we add a variance regularization term. The objective function is expressed as

$$\min_{\mathbf{f}_Q, \mathbf{f}_L} \sum_{i=1}^N (L_\delta(r^i) + k_r L_\delta(\mathbf{f}_Q^i - \overline{\mathbf{f}}_Q) + k_r L_\delta(\mathbf{f}_L^i - \overline{\mathbf{f}}_L)), \quad (32)$$

where  $L_\delta$  is Huber loss,  $k_r$  is the regularization parameter,  $\overline{\mathbf{f}}_Q$  and  $\overline{\mathbf{f}}_L$  are the mean of the forces in the sliding window. We solve this problem by L-BFGS [27].

To incorporate with NMPC, we assume the external forces  $\mathbf{f}_Q$  and  $\mathbf{f}_L$  in system dynamics (equation (18)-(22)) keep constant during the prediction horizon. The external forces are defined as online parameters of the system dynamics model. Our estimator updates the  $\mathbf{f}_Q$  and  $\mathbf{f}_L$  at a high frequency (100Hz), allowing the controller to continuously adapt to the changing system dynamics and external conditions.

### C. INDI Inner Loop Controller

In the previous sections, we always assume the payload is attached to the UAV mass center. However, in practice, this assumption is hard to satisfy, as shown in Figure. 4(b). This leads to an additional torque on the UAV. Measuring the distance between CoM and attached point and modeling the associated torque is non-trivial. Inspired by [28], [29], we adopt an Incremental Nonlinear Dynamic Inversion-based inner loop controller to eliminate this torque implicitly.

After getting the angular velocity commands  $\boldsymbol{\omega}_r$  from previous NMPC controller, The desired angular acceleration  $\dot{\boldsymbol{\omega}}_c$  is calculated as

$$\dot{\boldsymbol{\omega}}_c = K_\omega(\boldsymbol{\omega}_r - \boldsymbol{\omega}_f) + \dot{\boldsymbol{\omega}}_r, \quad (33)$$

where  $\boldsymbol{\omega}_f$  is the filtered feedback angular velocity from quadrotor IMU,  $\dot{\boldsymbol{\omega}}_r$  denotes reference angular acceleration. The INDI control law [28] is then applied to calculate the control moment  $\boldsymbol{\tau}_c$ .

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_f + \mathbf{J}(\dot{\boldsymbol{\omega}}_c - \dot{\boldsymbol{\omega}}_f), \quad (34)$$

where the  $\boldsymbol{\tau}_f$  is the filtered feedback angular moment obtained through the angular velocity of each propeller based on the propeller model (equation (28)) and  $\dot{\boldsymbol{\omega}}_f$  is the feedback angular acceleration. The term  $\boldsymbol{\tau}_f - \mathbf{J}\dot{\boldsymbol{\omega}}_f$  in (34) can implicitly compensate the unknown torque on the UAV [28]. Then we solve the following equation to obtain each desired rotor angular velocity  $n_c$ .

$$\begin{bmatrix} f_c \\ \boldsymbol{\tau}_c \end{bmatrix} = \mathbf{G}_1 \mathbf{n}_c^2 + \Delta t_m^{-1} \mathbf{G}_2 (\mathbf{n}_c - \mathbf{n}_f), \quad (35)$$

where  $\Delta t_m$  is the motor time constant and  $\mathbf{n}_f$  is the current rotor angular velocity. In the implement,  $\boldsymbol{\omega}_f$  and  $\mathbf{n}_f$  are filtered with the second-order Butterworth filter with a same cut-off frequency (10Hz) to ensure that they have a similar phase and delay. This is important for sensors' synchronization. The  $\dot{\boldsymbol{\omega}}_f$  is obtained from noise-robust numerical derivative to avoid IMU's noise being amplified.

## VI. EXPERIMENT AND EVALUATION

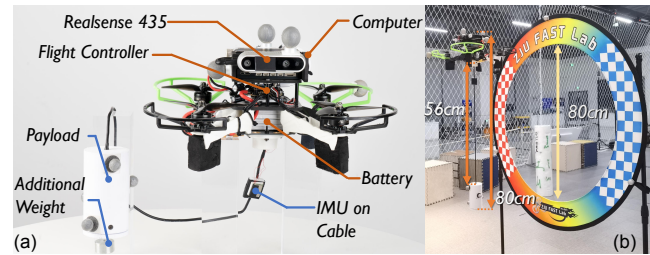


Fig. 5. (a) Quadrotor with suspended payload platform. Additional weight is hooked on the payload. (b) Gate in aggressive flight experiment.

We evaluate the performance of the proposed planning and control methods through adequate simulation and real-world experiments. In this section, we aim to answer three questions. i) Can the planning module generate dynamic feasible, and collision-free trajectories in clutter environments while meeting the real-time requirement? (See VI-A.) ii) Can the controller resist the disturbance and the model parameter inaccurate, achieving "load-and-fly"? (See VI-B.) iii) What is the performance of the whole framework in the real world? (See VI-C, VI-D and our video.)

We set up our experiments with a customized quadrotor platform, shown in Figure 5 (a). The platform consists of an Ardupilot flight controller (Kakute H7 stack), an onboard computer (DJI Manifold 2C), and an Intel RealSense D435 camera. The payload has a mass of 285g, with the capability to add additional weight. The quadrotor's propeller angular velocities are obtained through bi-directional dshot ESC. Accurate positioning information for quadrotor and suspended payload is provided by a motion capture system (MoCap). We fuse the data from IMU on the cable and MoCap to estimate the direction of the cable and payload's position, velocity by extend kalman filter. The IMU data is used for prediction, while the MoCap provides high-precision correction. All the modules, including planning, controller, and perception, run on the onboard computer.

### A. Planning Performance

To answer the first question, we evaluate the planning module in three simulation scenarios, like Figure 6, and compare its average computational time and successful rate against other planning methods [3], [12]. Due to no source code available, we implement other methods with CasADi [30] by ourselves. The first scene is a simple 3D environment containing 12 random squares. The second scene is passing a random gap. The third scene is a complex and cluttered environment. We created 10 simulation environments for each scenario to measure the success rate. It should be noted that

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.**

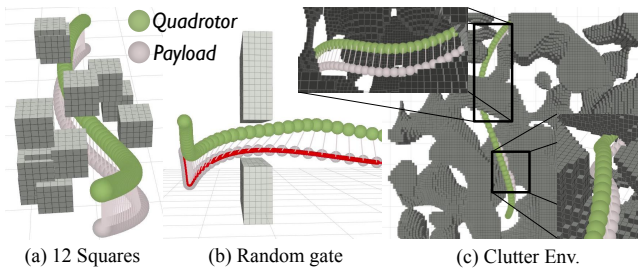


Fig. 6. Planning experiments in three different scenarios.

TABLE I  
PLANNING PERFORMANCE COMPARISON

	12 Squares		Random Gap		Clutter Env.	
	Time[s]	Succ.	Time	Succ.	Time	Succ.
Ours	<b>0.021</b>	<b>100%</b>	<b>0.017</b>	<b>100%</b>	<b>0.063</b>	<b>90%</b>
Foehn [3]	17.1	90%	15.8	80%	54.8	40%
Son [12]	0.437	80%	2.2	50%	-	-

Son's method [12] does not support non-convex environments, so we only tested it in the first two scenarios.

Table I illustrates our method achieved the highest success rate of 97.5% while offering a significant improvement in computation time compared to other methods, as our method performed hundreds of times faster.

### B. NMPC Ablation Study

To answer the second question, we evaluate the proposed controller on the same eight trajectory like Figure 7. We evaluate the controller's basic performance with different speed profiles and assess its performance against parameter uncertainty by introducing two unknown weights (50g and 200g). To validate its performance in unfamiliar conditions, we create two different wind environments using a fan, with wind speeds of 4.5 m/s and 9 m/s. The performance is measured as RMSE/MAX in Table II.

The results illustrate that NMPC with force compensation and INDI has the best performance. Overall, the NMPC with force compensation and INDI can reduce about 35% tracking error. As Figure 7, when we hung an additional 200g weight on the payload, the NMPC with force compensation and INDI can still keep the similar performance with no additional weight while without the force compensation and INDI, the tracking error increases significantly.

Furthermore, We add a 50g additional weight when the quadrotor is flying to verify the force estimator's performance. As shown in Figure 8, when we load the weight, the z-axis of  $f_Q$  changes from 0N to about -0.49N. This enables load-and-fly operation where the payload can be attached to the quadrotor without prior knowledge of its weight.

### C. Aggressive Flight Experiment

We evaluate the whole framework in an aggressive flight experiment, where the system has to pass through a gate, shown as Figure 5 (b). In this experiment, we aim to demonstrate the ability of our planning method to generate dynamically feasible trajectories in real time, even in challenging scenarios. The controller can resist unknown weight added to the payload, even with maneuvers.

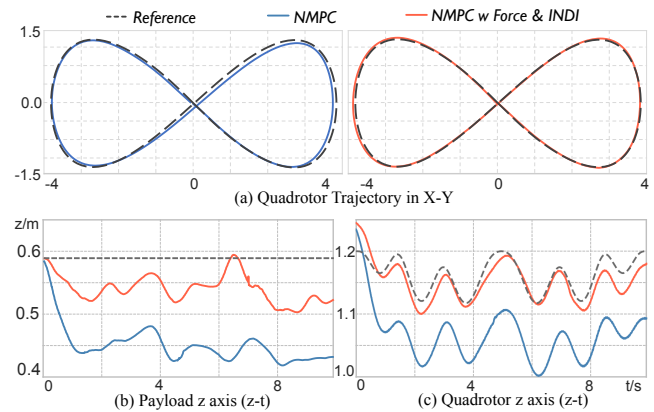


Fig. 7. Position Tracking Comparison with 200g unknown weight. (a) shows quadrotor tracking error in X-Y plane. (b),(c) show the payload and quadrotor tracking error in Z axis. NMPC with force compensation and INDI shows better tracking performance than only NMPC.

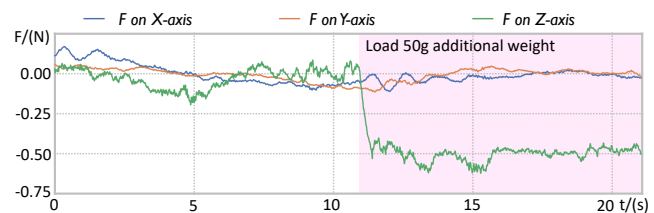


Fig. 8. Force estimation results ( $f_Q$ ) when we load a 50g additional weight.

In the experiment, The UAV payload system, even with an unknown weight (50g), passed through the gate multiple times, with the gate's position varied for each repetition. The maximum velocity and acceleration of the payload were measured at 5.2m/s and 7.3m/s<sup>2</sup>, respectively. The system exhibited the capability to navigate through the gate in real time while performing the necessary maneuvers.

### D. Autonomous Navigation in Unknown Environments

To further validate the robustness of our method, we conduct autonomous navigation experiments in complex unknown environments. The UAV payload system was equipped with an onboard camera for environment perception. The motion planning module was challenged by the complex surroundings and limited perception range, requiring rapid and continual trajectory re-planning in response to sudden obstacles to ensure safety. Three different experiment cases were set up, and some shots are presented in Figure 9. The full experiments can be seen in the accompanying video.

## VII. CONCLUSION

This paper proposes a practical solution for autonomous UAV payload system navigation. We proposed an efficient trajectory generation formulation meeting safe and real-time requirements and an NMPC controller with a hierarchical disturbance compensation strategy to resist disturbances. The sufficient experiments demonstrate the effectiveness and superiority of our method. Currently, we assume the external forces remain constant for a short time, which may result in a mismatch. Our future work will focus on incorporating a small neural network to predict external disturbances.

TABLE II  
NMPC ABLATION STUDY IN REAL ENVIRONMENT

Trajectory & Environment			Quadrotor Position Tracking Error (RMSE/MAX) [cm]				Payload Position Tracking Error (RMSE/MAX) [cm]			
$v_{max}$ [m/s]	$a_{max}$ [m/s <sup>2</sup> ]	Condition	NMPC	w INDI	w Force	w Force & INDI	NMPC	w INDI	w Force	w Force & INDI
1.5	2.1	None	5.2/9.1	4.9/7.9	3.4/6.5	<b>2.4/5.0</b>	8.4/13.7	9.3/14.1	6.8/10.9	<b>4.4/8.8</b>
2.5	3.7		7.1/11.5	7.1/11.7	4.4/6.9	<b>3.9/6.5</b>	10.0/21.7	9.9/21.1	8.7/13.1	<b>5.6/10.9</b>
4.0	5.7		7.7/12.8	7.6/13.3	5.1/9.9	<b>4.9/9.7</b>	10.2/22.1	11.9/25.5	9.4/19.7	<b>7.7/12.1</b>
5.9	9.0		9.0/14.5	8.7/15.3	7.9/17.3	<b>7.1/15.0</b>	14.6/27.2	14.6/26.7	12.1/27.1	<b>11.5/22.0</b>
4.0	5.7	Weight +50g	8.0/12.6	6.8/11.4	5.8/10.2	<b>5.4/10.5</b>	13.3/24.3	12.2/21.1	10.2/19.9	<b>9.5/15.3</b>
		Weight +200g	12.4/15.5	11.8/15.2	6.4/11.5	<b>5.5/11.0</b>	19.5/27.0	18.0/25.0	<b>10.8/17.1</b>	10.9/17.1
		Wind 4.5m/s	8.0/14.3	7.3/11.8	6.0/11.3	<b>5.4/11.1</b>	11.1/23.3	10.5/18.1	9.7/19.9	<b>9.0/15.5</b>
		Wind 9m/s	8.2/13.6	7.2/11.6	6.8/11.7	<b>5.5/10.0</b>	11.9/22.0	10.5/18.2	9.8/21.3	<b>9.2/20.3</b>

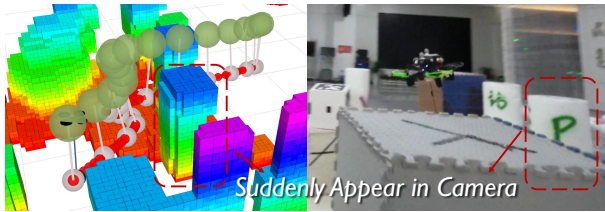


Fig. 9. The UAV-payload system performs autonomous navigation in complex unknown environments. It can fast modify the trajectory to handle the sudden appearance of new obstacles due to the limited perception range.

REFERENCES

[1] J. Fishman, S. Ubellacker, N. Hughes, and L. Carlone, “Dynamic Grasping with a “Soft” Drone: From Theory to Practice,” in *IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (IROS)*, 2021, pp. 4214–4221.

[2] 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 Intl. Conf. on Robot. and Autom. (ICRA)*, 2013, pp. 4888–4895.

[3] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, “Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload,” in *Proc. of Robot.: Sci. and Syst. (RSS)*, vol. 13, 2017.

[4] K. Sreenath and V. Kumar, “Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load,” in *IEEE Control and Decision Conf. (CDC)*, 2013, pp. 2269–2274.

[5] C. Y. Son, D. Jang, H. Seo, T. Kim, H. Lee, and H. J. Kim, “Real-time optimal planning and model predictive control of a multi-rotor with a suspended load,” *IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, pp. 5665–5671, 2019.

[6] N. D. Potdar, G. C. H. E. de Croon, and J. Alonso-Mora, “Online trajectory planning and control of a MAV payload system in dynamic environments,” *Auton. Robots*, vol. 44, no. 6, pp. 1065–1089, 2020.

[7] J. Zeng, P. Kotaru, M. W. Mueller, and K. Sreenath, “Differential Flatness Based Path Planning with Direct Collocation on Hybrid Modes for a Quadrotor with a Cable-Suspended Payload,” *IEEE Robot. Autom. Lett. (RA-L)*, vol. 5, no. 2, pp. 3074–3081, 2020.

[8] 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 Intl. Conf. on Robot. and Autom. (ICRA)*, 2012, pp. 2691–2697.

[9] I. Palunko, P. Cruz, and R. Fierro, “Agile Load Transportation : Safe and Efficient Load Manipulation with Aerial Robots,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 69–79, Sept. 2012.

[10] J. Silveira, S. N. Givigi, E. O. Freire, L. Molina, and E. Carvalho, “Aggressive Motion Planning for a Quadrotor System with Slung Load Based on RRT,” in *IEEE Intl. Systems Conf.*, 2020, pp. 1–7.

[11] S. Tang and V. Kumar, “Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload,” *IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, pp. 2216–2222, 2015.

[12] C. Y. Son, H. Seo, D. Jang, and H. J. Kim, “Real-time optimal trajectory generation and control of a multi-rotor with a suspended load for obstacle avoidance,” *IEEE Robot. Autom. Lett. (RA-L)*, vol. 5, no. 2, pp. 1915–1922, Apr. 2020.

[13] F. A. Goodarzi, D. Lee, and T. Lee, “Geometric stabilization of a quadrotor UAV with a payload connected by flexible cable,” in *American Control Conference*, 2014, pp. 4925–4930.

[14] 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 Control and Decision Conf. (CDC)*, Dec. 2015, pp. 6718–6723.

[15] 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, 2017.

[16] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, “A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots,” in *IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2013, pp. 4896–4901.

[17] P. Pereira, J. Cortes, and D. V. Dimarogonas, “Aerial Slung-Load Position Tracking Under Unknown Wind Forces,” *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 3952–3968, Sept. 2021.

[18] G. Li, A. Tunchez, and G. Loianno, “PCMPC: Perception-Constrained Model Predictive Control for Quadrotors with Suspended Loads using a Single Camera and IMU,” in *IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2021, pp. 2012–2018.

[19] Z. Wang, X. Zhou, C. Xu, and F. Gao, “Geometrically Constrained Trajectory Optimization for Multicopters,” *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, Oct. 2022.

[20] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2011, pp. 2520–2525.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, “Distance transforms of sampled functions,” *Theory of computing*, vol. 8, pp. 415–428, 2012.

[22] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, and F. Gao, “Differential Flatness-Based Trajectory Planning for Autonomous Vehicles,” Aug. 2022.

[23] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight,” *IEEE Robot. Autom. Lett. (RA-L)*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.

[24] L. Wang, “Use of exponential data weighting in model predictive control design,” in *IEEE Control and Decision Conf. (CDC)*, 2001, pp. 4857–4862.

[25] T. Yoon and D. Clarke, “Receding-horizon predictive control with exponential weighting,” *International Journal of Systems Science*, vol. 24, no. 9, pp. 1745–1757, 1993.

[26] B. Houska, H. Ferreau, and M. Diehl, “An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range,” *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[27] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[28] E. Tal and S. Karaman, “Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness,” *IEEE transactions on control systems technology*, vol. 29, no. 3, pp. 1203–1218, 2021.

[29] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, “A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight,” *IEEE Transactions on Robotics*, pp. 1–17, 2022.

[30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: A software framework for nonlinear optimization and optimal control,” *Mathematical Prog. Comput.*, vol. 11, pp. 1–36, 2019.