

# Towards Fully Onboard State Estimation and Trajectory Tracking for UAVs with Suspended Payloads

Martin Jiroušek<sup>a</sup>, Tomáš Báča<sup>b</sup> and Martin Saska<sup>c</sup>

*Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic*

**Keywords:** Unmanned Aerial Vehicle, Suspended Payload, Autonomous Aerial Transportation, Onboard Estimation, Model Predictive Control.

**Abstract:** This paper addresses the problem of tracking the position of a cable-suspended payload carried by an unmanned aerial vehicle, with a focus on real-world deployment and minimal hardware requirements. In contrast to many existing approaches that rely on motion-capture systems, additional onboard cameras, or instrumented payloads, we propose a framework that uses only standard onboard sensors—specifically, real-time kinematic global navigation satellite system measurements and data from the onboard inertial measurement unit—to estimate and control the payload’s position. The system models the full coupled dynamics of the aerial vehicle and payload, and integrates a linear Kalman filter for state estimation, a model predictive contouring control planner, and an incremental model predictive controller. The control architecture is designed to remain effective despite sensing limitations and estimation uncertainty. Extensive simulations demonstrate that the proposed system achieves performance comparable to control based on ground-truth measurements, with only minor degradation ( $< 6\%$ ). The system also shows strong robustness to variations in payload parameters. Field experiments further validate the framework, confirming its practical applicability and reliable performance in outdoor environments using only off-the-shelf aerial vehicle hardware.

## 1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are playing an increasingly vital role across a wide range of applications, from aerial mapping (Colomina and Molina, 2014), infrastructure inspection (Sikora et al., 2023), and precision agriculture (Gode et al., 2024), to last-mile delivery (Murray and Chu, 2015) and environmental monitoring (C. et al., 2024). As their use becomes more widespread, there is a growing demand for advanced control strategies that ensure reliable operation in challenging, real-world conditions.

In this paper, we address the problem of tracking the position of a payload suspended from a UAV. Our focus lies in enabling reliable deployment in field conditions while minimizing hardware requirements. We propose a solution that operates using only standard UAV hardware—specifically, an RTK GNSS receiver and the onboard sensors available in most flight controllers (e.g., IMU, barometer, magnetometer...).

Based on these inputs, we estimate and control the position of the suspended payload along a predefined reference trajectory.

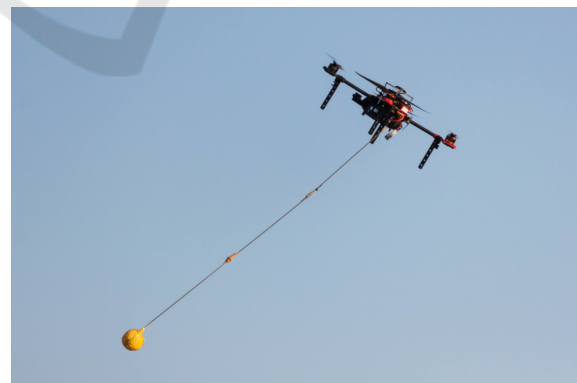


Figure 1: UAV carrying a cable suspended payload.

In many practical applications, it is the position of the payload—rather than the UAV itself—that must be accurately controlled. For example, in agile pickup-and-delivery tasks, the end of the cable can reach the target before the UAV, exploiting the com-

<sup>a</sup> <https://orcid.org/0009-0003-4552-9995>

<sup>b</sup> <https://orcid.org/0000-0001-9649-8277>

<sup>c</sup> <https://orcid.org/0000-0001-7106-3816>

plex system dynamics. Similarly, in applications such as deploying sensors into hazardous or constrained environments, contacting surfaces with the payload, or precise mechanical interaction with the environment, controlling the payload's position is of primary importance. Reducing the hardware and instrumentation requirements is key to simplifying deployment and lowering system cost—motivating our reliance solely on RTK GNSS and standard flight-controller sensors.

This problem presents significant challenges. The UAV-payload system is highly underactuated and exhibits complex, nonlinear coupled dynamics. Compounding the difficulty, the position and orientation of the payload are not directly measured in our setup. Existing state-of-the-art solutions either rely on adaptive controllers that treat the payload dynamics as disturbances (Li et al., 2023), (Wang et al., 2024), or on perception-constrained control frameworks that use downward-facing cameras to track the payload (Li et al., 2021), (Recalde et al., 2025), (Sarvaiya et al., 2025). However, vision-based approaches introduce substantial complexity, cost, and sensitivity to environmental conditions, as well as constraints on maintaining payload visibility within the camera's field of view. A key limitation across many prior works is the lack of robust, fully onboard estimation tightly coupled with control.

We address this gap by designing and validating a lightweight, onboard-only state estimation and control framework capable of real-world payload tracking using generic UAV hardware. Our solution consists of a linear Kalman filter (LKF)-based estimator, an incremental model predictive controller (MPC), and a motion-planning component based on model predictive contouring control (MPCC). The incremental MPC formulation is inherently robust to zero-mean noise, which allows us to tune the estimator for low bias at the cost of tolerable noise. Meanwhile, the MPCC planner enables smooth transitions between reference waypoints and allows flexible trajectory shaping.

Experimental results show that the proposed system performs comparably to an idealized setup using ground-truth payload position, with only marginal performance degradation. The framework also demonstrates robustness to variations in payload parameters and has been successfully deployed in real-world outdoor environments, achieving results consistent with simulation. These findings highlight the viability of practical, low-cost UAV-based suspended payload tracking using only standard onboard sensors.

## 2 RELATED WORK

Early research on UAVs with suspended payloads primarily focused on generating swing-free trajectories and stabilizing hover under load. A pioneering contribution (Palunko et al., 2012) demonstrated offline planning of swing-free maneuvers using dynamic programming and input-shaping techniques. Subsequent work (Faust et al., 2013) applied reinforcement learning to synthesize swing-minimized trajectories under uncertain dynamics. Later approaches, such as (Sreenath et al., 2013), leveraged differential flatness and geometric control to enable simultaneous tracking of both UAV and payload trajectories. The field has since expanded to include hybrid control frameworks (Wang et al., 2024), disturbance-robust planning (Li et al., 2023), and cooperative multi-robot transportation strategies (Zhang et al., 2023). However, accurate payload position tracking—particularly under onboard-only sensing—remains an open challenge.

Many works have focused exclusively on planning and control algorithms, typically evaluated in simulation or under motion-capture conditions. For instance, (Li et al., 2023) proposed a real-time NMPC framework with whole-body safety guarantees, but evaluated it only in simulation or controlled laboratory environments. In (Zhang et al., 2023), cooperative multi-UAV planning using insetting-formation methods was explored, though it relied heavily on external sensing and did not address onboard-only execution. Similarly, (Tang and Kumar, 2015) presented MIQP-based trajectory planning with hybrid dynamics and obstacle avoidance, assuming accurate state feedback. Overall, realistic onboard state estimation has often been abstracted or oversimplified in these control-centric studies.

A prominent line of research aims at estimating payload state—typically angles or position—to enable payload-aware control. Early methods employed inertial measurements or encoder-based observers to infer payload swing online, but required dedicated sensors (Rego and Raffo, 2016). More recent approaches have leveraged perception-driven estimation. For example, PCMPC (Li et al., 2021) proposed a perception-constrained MPC that fuses monocular camera and IMU data to estimate the cable direction and payload dynamics. Later, hybrid perception-aware frameworks such as HPA-MPC (Sarvaiya et al., 2025) and ES-HPC-MPC (Recalde et al., 2025) introduced advanced control and estimation strategies capable of handling slack-taut transitions, while enforcing visibility constraints via onboard cameras. Although these perception-enhanced MPC methods

demonstrate high performance, they require reliable visual feedback and incur significant computational overhead. In contrast, fully onboard estimation using only standard UAV sensors—such as the flight controller IMU and RTK GNSS—is rarely addressed, leaving a gap in lightweight, deployable solutions for field-ready platforms.

To the best of our knowledge, no prior work has demonstrated robust payload position tracking and control in fully outdoor conditions using only GNSS (specifically RTK) and standard flight-controller sensors—without relying on motion capture, external cameras, or payload instrumentation. Our work addresses this gap by designing and validating a lightweight, onboard-only state estimation and MPC framework capable of tracking suspended payloads in real-world outdoor settings using standard UAV hardware. This enables practical deployment across a broad range of platforms and outdoor environments.

### 3 MATHEMATICAL MODEL

The modeling of aerial systems transporting suspended payloads is a well-established problem in aerial robotics, with one of the most influential formulations introduced in (Palunko et al., 2012). Their model, based on a Lagrangian derivation under the assumption of a taut, massless cable, captures the coupled motion between the unmanned aerial vehicle and the payload with sufficient fidelity for control design, while remaining analytically tractable. In this work, we adopt the core structure of Palunko’s model and extend it to account for aerodynamic damping effects on both the aerial vehicle and the payload. The inclusion of linear air-drag terms for each body provides a more realistic description of energy dissipation in outdoor environments, improving the model’s suitability for real-time state estimation and control under realistic conditions. This mid-level formulation maintains a balance between physical fidelity and computational efficiency, enabling onboard implementation without sacrificing robustness to unmodeled effects.

#### 3.1 Assumptions

To ensure analytical tractability without significantly compromising realism, several simplifying assumptions are introduced. The cable linking the UAV and the payload is modeled as massless and always taut, with attachment points located precisely at the centers of mass (CoG) of both bodies. This eliminates any torque contributions or additional inertial effects due to the cable. The payload is treated as a rigid body

of known mass  $m_l$  and suspended at a fixed length  $l$ . Aerodynamic drag is included for both the UAV and payload, modeled as linear with respect to translational velocity. The payload is further assumed to exhibit symmetric aerodynamic characteristics, excluding rotational drag and shape-induced effects. Friction at the cable connection is neglected, thereby allowing the payload to rotate freely about the suspension point. Environmental disturbances such as wind are completely omitted to improve trackability of the model. These modeling choices align with established literature (Palunko et al., 2012) and yield a manageable yet sufficiently expressive dynamic formulation.

#### 3.2 System Description

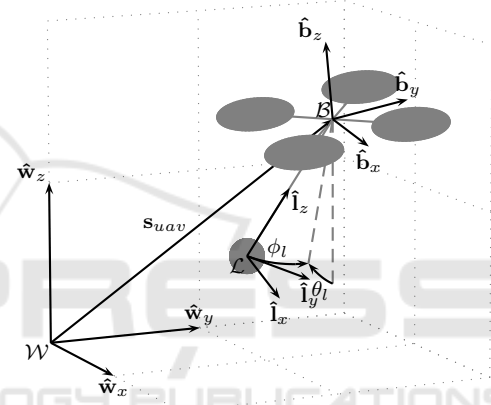


Figure 2: Coordinate systems and generalized coordinates of the UAV-payload system.

The system configuration is described using three right-handed Cartesian coordinate frames, as illustrated in Fig. 2. The world frame  $\mathcal{W}$  is an inertial frame fixed to the Earth, with  $\hat{\mathbf{w}}_z$  oriented upward. The body frame  $\mathcal{B}$  is rigidly attached to the UAV, centered at its CoG, with  $\hat{\mathbf{b}}_z$  aligned with the thrust direction. The load frame  $\mathcal{L}$  is centered at the payload’s CoG, with  $\hat{\mathbf{l}}_z$  aligned along the cable, and  $\hat{\mathbf{l}}_x, \hat{\mathbf{l}}_y$  parallel to the corresponding body frame axes.

The UAV’s generalized coordinates in the world frame are defined by:

$$\mathbf{q}_{uav} = [x \ y \ z \ \theta \ \phi \ \psi]^\top, \quad (1)$$

where  $\mathbf{s}_{uav} = [x \ y \ z]^\top$  denotes the UAV’s position, and  $\Gamma = [\theta \ \phi \ \psi]^\top$  corresponds to its orientation, expressed via Tait-Bryan angles. The rotation from body to world frame is given by:

$$\mathbf{R}_{\mathcal{B} \rightarrow \mathcal{W}}(\phi, \theta, \psi) = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi). \quad (2)$$

Under the assumption of a taut cable, the payload’s relative position is parameterized by two an-

gles:

$$\mathbf{q}_l = [\theta_l \ \phi_l]^\top, \quad (3)$$

which define the load-to-world rotation matrix as:

$$\mathbf{R}_{\mathcal{L} \rightarrow \mathcal{W}}(\phi_l, \theta_l) = \mathbf{R}_x(\phi_l) \mathbf{R}_y(\theta_l). \quad (4)$$

Using this formulation, the payload's absolute position becomes:

$$\mathbf{s}_l = \mathbf{R}_{\mathcal{L} \rightarrow \mathcal{W}} \begin{bmatrix} 0 & 0 & -l \end{bmatrix}^\top + \mathbf{s}_{uav}. \quad (5)$$

### 3.3 Equations of Motion

Since the system's potential and kinetic energies do not depend on UAV attitude, the dynamics are formulated using the reduced generalized coordinate vector:

$$\mathbf{q} = [\mathbf{s}_{uav} \ \mathbf{q}_l]^\top. \quad (6)$$

The Lagrangian is constructed in the standard form:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\dot{\mathbf{q}}) - \mathcal{V}(\mathbf{q}), \quad (7)$$

where the potential energy is expressed as:

$$\mathcal{V}(\mathbf{q}) = g \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top (\mathbf{m}_{uav} \mathbf{s}_{uav} + \mathbf{m}_l \mathbf{s}_l), \quad (8)$$

and the kinetic energy is given by:

$$\mathcal{T}(\dot{\mathbf{q}}) = \frac{1}{2} (\mathbf{m}_{uav} \|\dot{\mathbf{s}}_{uav}\|^2 + \mathbf{m}_l \|\dot{\mathbf{s}}_l\|^2). \quad (9)$$

The conservative force in the system originates from UAV thrust, modeled as:

$$\mathbf{f}_{con}(\Gamma, F) = \begin{bmatrix} \mathbf{R}_{\mathcal{B} \rightarrow \mathcal{W}}(\Gamma) \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}, \quad (10)$$

where  $F$  denotes the total collective thrust.

Aerodynamic drag represents the primary dissipative force and is incorporated as follows. The payload drag, projected onto angular coordinates, is modeled by:

$$\mathbf{f}_{dis}^l(\dot{\mathbf{q}}, \mathbf{q}) = -d_l \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \mathbf{R}_{\mathcal{L} \rightarrow \mathcal{W}}^{-1}(\theta_l, \phi_l) \dot{\mathbf{s}}_l^{\mathcal{W}}, \quad (11)$$

while the radial drag component acts on the UAV as:

$$\begin{aligned} \mathbf{f}_{dis}^{uav}(\dot{\mathbf{q}}, \mathbf{q}) &= -d_{uav} \dot{\mathbf{s}}_{uav} \\ -d_l \mathbf{R}_{\mathcal{L} \rightarrow \mathcal{W}}(\theta_l, \phi_l) &\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_{\mathcal{L} \rightarrow \mathcal{W}}^{-1}(\theta_l, \phi_l) \dot{\mathbf{s}}_l^{\mathcal{W}}. \end{aligned} \quad (12)$$

The complete expression for the dissipative force becomes:

$$\mathbf{f}_{dis}(\dot{\mathbf{q}}, \mathbf{q}) = [\mathbf{f}_{dis}^{uav} \ \mathbf{f}_{dis}^l]^\top = -\mathbf{D}(\phi_l, \theta_l) \dot{\mathbf{q}}. \quad (13)$$

Combining these elements and applying the Euler-Lagrange formalism yields the governing equations of motion:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + (\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{D}(\mathbf{q})) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{f}_{con}(\Gamma, F). \quad (14)$$

### 3.4 Flight Controller Model

The UAV's onboard flight controller (FCU) is abstracted as a set of decoupled first-order systems corresponding to each actuation channel. It tracks reference inputs  $\mathbf{u}$  via internal state variables  $\mathbf{x}_a = [\theta \ \phi \ \psi \ F]^\top$ , governed by the following dynamics:

$$\dot{\mathbf{x}}_a = \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a \mathbf{u}, \quad (15)$$

where  $\mathbf{A}_a$  and  $\mathbf{B}_a$  are diagonal matrices given by:

$$\begin{aligned} \mathbf{A}_a &= \text{diag} \left( -\frac{1}{\tau_1}, \dots, -\frac{1}{\tau_4} \right), \\ \mathbf{B}_a &= \text{diag} \left( \frac{K_1}{\tau_1}, \dots, \frac{K_4}{\tau_4} \right), \end{aligned} \quad (16)$$

with  $\tau_i$  and  $K_i$  denoting the time constants and gains, respectively, for each control channel.

### 3.5 State Space Model

The full nonlinear model, incorporating both physical dynamics and the flight controller, is expressed as:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= -\mathbf{M}^{-1}(\mathbf{q}) \left( (\mathbf{C}(\mathbf{v}, \mathbf{q}) + \mathbf{D}(\mathbf{q})) \mathbf{v} + \mathbf{g}(\mathbf{q}) - \mathbf{f}_{con}(\mathbf{x}_a) \right), \\ \dot{\mathbf{x}}_a &= \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a \mathbf{u}. \end{aligned} \quad (17)$$

It is noted that the mass matrix  $\mathbf{M}$  becomes singular at  $\theta_l = \frac{\pi}{2}$ , imposing a constraint on the domain of explicit solutions.

For the purposes of control and estimation design, the system is linearized about the hover equilibrium with a motionless payload:  $\theta_l = \phi_l = 0 \text{ rad}$ ,  $\dot{\theta}_l = \dot{\phi}_l = 0 \text{ rad s}^{-1}$ . The resulting linear time-invariant (LTI) model is:

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}, \quad (18)$$

where the state vector is composed as:

$$\mathbf{x} = [\mathbf{q} \ \mathbf{v} \ \mathbf{x}_a]^\top. \quad (19)$$

## 4 CONTROL FRAMEWORK

The control framework consists of three components: a state estimator, a tracking controller, and a trajectory planner, as shown in Fig. 3. The state estimator reconstructs the full system state, including unmeasured variables such as the payload position, using UAV position and attitude measurements in conjunction with the system model introduced in Section 3.

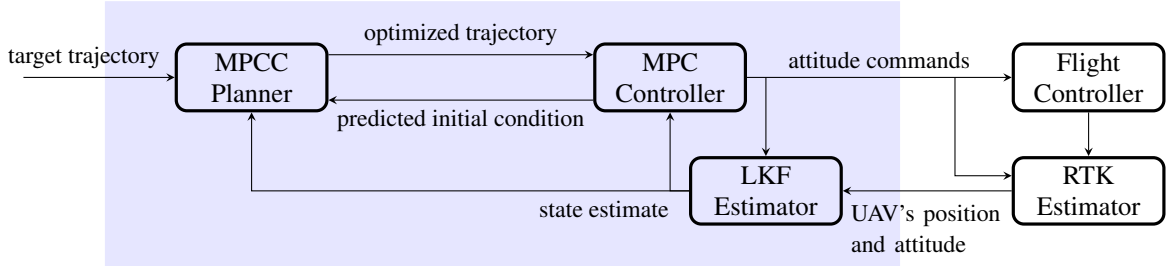


Figure 3: Architecture of the control framework. Blue block represents the proposed solution.

The tracking controller computes UAV attitude commands that ensure the payload tracks a desired trajectory by solving a constrained optimal control problem in a receding horizon fashion. Finally, the trajectory planner generates smooth, dynamically feasible reference trajectories online from sparse target states using temporally weighted optimization.

These components operate at different rates to ensure computational efficiency: the estimator and controller run at 100 Hz, while the trajectory planner executes at 1 Hz, enabling long-horizon planning without compromising real-time control performance.

#### 4.1 State Estimator

We assume the UAV's position is measured via RTK GNSS, while its attitude is provided by the onboard flight controller. Rather than relying on raw measurements, we employ the RTK estimator from the MRS UAV System (Baca et al., 2021), which provides built-in health checks and outlier rejection to ensure robust operation under field conditions.

To incorporate these measurements, the linear state-space model from equation (18) is augmented with a measurement model:

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 7} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 7} & \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \end{bmatrix} \mathbf{x}, \quad (20)$$

where  $\mathbf{I}_3$  is the  $3 \times 3$  identity matrix and  $\mathbf{0}_{a \times b}$  denotes an  $a \times b$  zero matrix.

The estimator is implemented as a Linear Kalman Filter (LKF) (Kalman, 1960) with empirically tuned noise covariances  $\mathbf{Q}$  and  $\mathbf{R}$ , which balance smoothness and responsiveness (values listed in Table 7).

To account for variability in control loop timing—caused primarily by the onboard online optimization—the estimator employs forward Euler discretization with the current time step:

$$\tilde{\mathbf{A}}(dt) = \mathbf{I} + dt\mathbf{A}, \quad \tilde{\mathbf{B}}(dt) = dt\mathbf{B}, \quad (21)$$

ensuring consistent and time-accurate state propagation even when the loop interval varies.

#### 4.2 Tracking Controller

The tracking controller generates UAV attitude commands to track reference trajectories while accounting for constraints and dynamic limitations. It is formulated as a constrained optimal control problem and solved using Model Predictive Control (MPC) based on the linearized system dynamics.

The optimization problem is posed as a quadratic program (QP) with soft constraints:

$$\begin{aligned} \min_{\substack{\tilde{\mathbf{x}}_t, \dots, \tilde{\mathbf{x}}_{t+N} \\ \tilde{\mathbf{u}}_t, \dots, \tilde{\mathbf{u}}_{t+N-1} \\ \tilde{\mathbf{s}}_t, \dots, \tilde{\mathbf{s}}_{t+N}}} & \sum_{n=t}^{t+N} \frac{1}{2} \begin{bmatrix} \tilde{\mathbf{x}}_n \\ \tilde{\mathbf{u}}_n \\ 1 \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_n & \mathbf{0} & \mathbf{q}_n \\ \mathbf{0} & \mathbf{R}_n & \mathbf{0} \\ \mathbf{q}_n^\top & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_n \\ \tilde{\mathbf{u}}_n \\ 1 \end{bmatrix} + \\ & \frac{1}{2} \begin{bmatrix} \tilde{\mathbf{s}}_n^l \\ \tilde{\mathbf{s}}_n^u \end{bmatrix}^\top \begin{bmatrix} \mathbf{Z}_n^l & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_n^u \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{s}}_n^l \\ \tilde{\mathbf{s}}_n^u \end{bmatrix}, \end{aligned} \quad (22)$$

$$\text{s.t. } \tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{A}}_n \tilde{\mathbf{x}}_n + \tilde{\mathbf{B}}_n \tilde{\mathbf{u}}_n, \quad n = t, \dots, t+N-1,$$

$$\begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\mathbf{x}}_n \end{bmatrix} \leq \begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\mathbf{x}}_n \end{bmatrix} + \tilde{\mathbf{s}}_n^l, \quad n = t, \dots, t+N,$$

$$\begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\mathbf{x}}_n \end{bmatrix} \geq \begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\mathbf{x}}_n \end{bmatrix} + \tilde{\mathbf{s}}_n^u, \quad n = t, \dots, t+N. \quad (23)$$

To gain finer control over the control actions and allow penalization of their rate, we adopt an incremental model formulation introduced in (Qin and Badgwell, 2003):

$$\begin{bmatrix} \mathbf{x}_{n+1} \\ \mathbf{u}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_n \\ \mathbf{u}_n \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{u}_n. \quad (24)$$

This formulation helps prevent the propagation of noise from the state estimate to the control action and introduces robustness against estimation errors.

The payload position is estimated from the system state using a linearized form of Equation (5), including control inputs:

$$\tilde{\mathbf{y}}_n = \begin{bmatrix} \mathbf{s}_{\text{uav},n} + l \begin{bmatrix} -\theta_{l,n} \\ \phi_{l,n} \\ -1 \end{bmatrix} \\ \mathbf{u}_n \end{bmatrix} = \tilde{\mathbf{C}} \begin{bmatrix} \tilde{\mathbf{x}}_n \\ 1 \end{bmatrix}. \quad (25)$$

Deviation from the reference is penalized through:

$$\mathbf{Q}_n = \tilde{\mathbf{C}}^\top \text{diag}(\mathbf{p}_{s_l}, \mathbf{p}_u) \tilde{\mathbf{C}}, \quad \mathbf{R}_n = \text{diag}(\mathbf{p}_{\Delta u}), \quad (26)$$

with values specified in Table 7.

Slack variables ensure feasibility under noise or disturbances, and the reference trajectory is introduced via a linear state term:

$$\mathbf{q}_n = \begin{pmatrix} \mathbf{x}_n^d \end{pmatrix}^\top \mathbf{Q}_n. \quad (27)$$

The only states constrained in the optimization are translational velocity (per-axis) and the UAV's attitude. The optimization problem is solved by the HPIPM solver (Frison and Diehl, 2020) using partial condensation (Axehill, 2015).

### 4.3 Trajectory Planner

The trajectory planner generates a dynamically feasible reference trajectory from target states using long-horizon optimization, complementing the fast controller operating at shorter time scales.

It solves an optimization problem structurally similar to the controller's formulation in (22), (23), but adopts the Model Predictive Contouring Control (MPCC) paradigm introduced in (Romero et al., 2022). The key difference lies in the time-varying weighting matrix  $\mathbf{Q}_i$  applied along the planning horizon:

$$\mathbf{Q}_i = \omega_i \mathbf{Q}, \quad i = 0, \dots, N, \quad (28)$$

where

$$\omega_i = \max_{k=1, \dots, M} \exp \left( -\frac{(t_{r,k} - t_i)^2}{2\sigma^2} \right), \quad i = 0, \dots, N. \quad (29)$$

Here,  $t_{r,k}$  denotes the timestamp of the  $k$ -th state in the reference trajectory (consisting of  $M$  states), and  $t_i$  is the timestamp of the  $i$ -th step in the planning horizon. The parameter  $\sigma$  controls the spread of the Gaussian kernels used for weighting.

This temporal weighting approach allows the planner to gradually transition between sparse reference points while maintaining smoothness and feasibility. For large values of  $\sigma$ , the weights vary slowly, encouraging adherence to the reference trajectory throughout the horizon. Conversely, small  $\sigma$  values lead to sharp peaks in the weighting, guiding the system to closely match specific reference states while allowing flexibility in between. The specific value of  $\sigma$  used is provided in Table 7.

A limitation of long-horizon MPC planners is that significant computation time may cause part of the computed trajectory to become outdated before it is applied. To mitigate this, the planner uses a predicted system state at time  $t + t_{\text{plan}}$  as its initial condition,

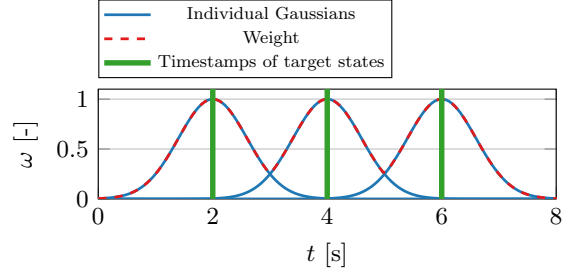


Figure 4: Illustration of the Gaussian weighting function  $\omega_i$  along the planning horizon, centered at reference times  $t_{r,k}$ .

where  $t_{\text{plan}}$  is an estimate of the planner's execution time. This predicted state is available from the controller's internal MPC prediction and ensures temporal alignment between planning and control.

### 4.4 Limitations

The proposed control framework assumes a constant UAV heading, as the payload-induced disturbances are modeled in the world coordinate frame and the system dynamics are linearized around a fixed heading. Another limitation is the assumption that the cable remains taut throughout the flight. Additionally, the framework requires prior knowledge of the payload's parameters, such as mass and rope length. The current implementation is not adaptive and relies on these parameters being known and fixed. However, we consider these limitations to be acceptable in many practical scenarios. If necessary, they could be addressed through more complex adaptive, hybrid and nonlinear methods, albeit at the cost of increased computational demands.

## 5 VERIFICATION

This section verifies the full control pipeline using the proposed metrics in both simulation and real-world experiments. While the system is capable of tracking densely sampled reference trajectories, such trajectories offer limited freedom for planning. In contrast, sparser trajectories allow for more planning optimizations and, when executed at higher speeds, better expose the dynamic properties of the suspended payload.

For instance, a slowly moving, densely sampled reference trajectory can be accurately followed using a conventional UAV position controller with a fixed vertical offset corresponding to the cable length. In such cases, the payload remains nearly stationary, and swing angles  $\theta_l$  and  $\phi_l$  are negligible. Consequently, the strengths and limitations of the proposed control

framework are most apparent when tracking agile, sparse trajectories.

## 5.1 Methodology

Tracking errors on sparse reference trajectories are often difficult to interpret due to their potential inclusion of discontinuities or infeasible transitions. As such, absolute RMSE values can vary significantly depending on the chosen waypoint discretization.

To establish a fair baseline, we introduce an *open-loop (OL) planner*, which solves the same MPCC problem (refer to Section 4.3) across the entire sparse reference to generate a dynamically feasible trajectory. These pre-planned OL trajectories, shown in Fig. 5, serve as an idealized benchmark against which closed-loop performance can be measured.

We evaluate the full closed-loop system—including onboard estimation and MPC tracking—against this baseline using the *relative RMSE degradation*:

$$\Delta_{\text{RMSE}} = \frac{\text{RMSE}(\mathbf{x}_{\text{exec}}) - \text{RMSE}(\mathbf{x}_{\text{OL}})}{\text{RMSE}(\mathbf{x}_{\text{OL}})} \times 100\%, \quad (30)$$

where  $\text{RMSE}(\mathbf{x})$  is computed over a trajectory  $\mathbf{x}$  consisting of states  $\mathbf{s}_0, \dots, \mathbf{s}_N$  as:

$$\text{RMSE}(\mathbf{x}) = \sqrt{\frac{1}{N+1} \sum_{n=0}^N \|\mathbf{s}_n^{\text{ref}} - \mathbf{s}_n\|^2},$$

using zero-order hold to define a reference state  $\mathbf{s}_n^{\text{ref}}$  at step  $n$ .

The metric  $\Delta_{\text{RMSE}}$  captures how much tracking performance deteriorates compared to the ideal OL plan. Increased degradation indicates effects of estimation noise, limited prediction horizons, and model mismatch. We report  $\Delta_{\text{RMSE}}$  across varying trajectory speeds and payload parameters to assess robustness.

Two reference trajectories are used for evaluation: a square trajectory and a complex, custom-designed one. Both are composed of 3D waypoints with uniform temporal spacing defined by a time step  $dt$ .

The *square trajectory* consists of alternating 5 m steps along the  $x$  and  $y$  axes, forming orthogonal segments to evaluate step response and speed sensitivity (Fig. 5). The *complex trajectory* includes changes in altitude, direction, and spacing, designed to challenge the controller with dynamically rich behavior (visualized in Fig. 8).

## 5.2 Simulation

Simulations were performed using the high-fidelity Gazebo environment and the MRS UAV System,

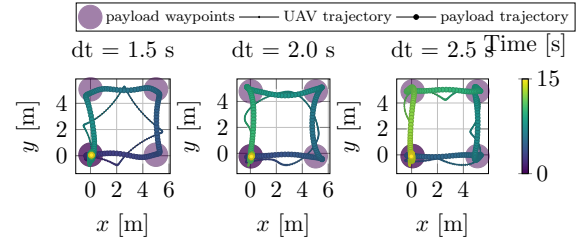


Figure 5: Open-loop planned trajectories for the square scenario, shown at different reference speeds.

known for accurate sim-to-real transfer (Baca et al., 2021). The payload is modeled as a chain of 10 rigid links connected via spherical joints. We analyze performance in terms of both trajectory speed and payload parameters.

### 5.2.1 Effect of Trajectory Speed

Simulations on the square trajectory were conducted at different speeds by varying the time step  $dt \in \{1.5, 2.0, 2.5\}$  s. Results in Table 1 confirm that faster execution increases RMSE, as expected. For  $dt = 2.5$  s, the tracking error nearly matches that of the OL baseline, while for  $dt = 1.5$  s, the degradation remains below 40%.

Table 1: Tracking RMSE degradation on square trajectory with varying speed ( $l = 2$  m,  $m_l = 1.5$  kg).

| $dt$ [s] | $\text{RMSE}(\mathbf{x}_{\text{OL}})$ [m] | $\Delta_{\text{RMSE}}^{\text{gt}}$ [%] | $\Delta_{\text{RMSE}}$ [%] |
|----------|-------------------------------------------|----------------------------------------|----------------------------|
| 1.5      | 0.873                                     | 32.45                                  | 38.03                      |
| 2.0      | 0.864                                     | 16.12                                  | 18.79                      |
| 2.5      | 0.874                                     | 1.33                                   | 4.61                       |

Table 2 shows estimation RMSEs for the swing angles and angular velocities. Estimation accuracy remains stable and correlates with trajectory excitation levels. Minimal difference between using ground-truth and estimated states ( $< 6\%$ ) demonstrates the estimator's reliability. Figure 6 illustrates a representative tracking result.

Table 2: Estimation RMSE for square trajectory with varying speed ( $l = 2$  m,  $m_l = 1.5$  kg).

| $dt$ [s] | $\theta_l$ [rad] | $\phi_l$ [rad] | $\dot{\theta}_l$ [rad/s] | $\dot{\phi}_l$ [rad/s] |
|----------|------------------|----------------|--------------------------|------------------------|
| 1.5      | 0.085            | 0.097          | 0.167                    | 0.202                  |
| 2.0      | 0.049            | 0.052          | 0.115                    | 0.126                  |
| 2.5      | 0.075            | 0.095          | 0.149                    | 0.203                  |

### 5.2.2 Effect of Payload Parameters

To evaluate robustness against different payload configurations, we simulate the complex trajectory for

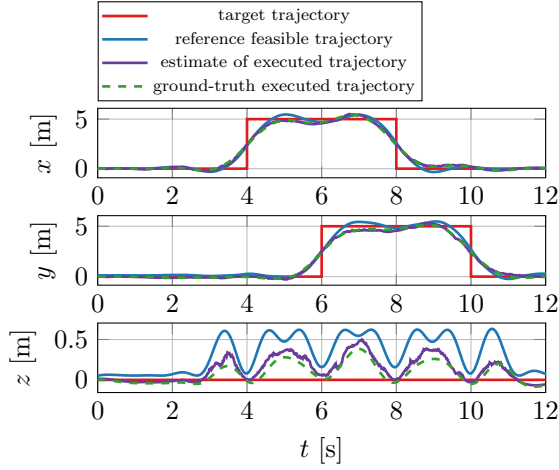


Figure 6: Closed-loop tracking on square trajectory ( $dt = 2$  s).

varying cable lengths and masses (Table 3).

Estimation RMSE decreases with increasing mass, likely due to stronger excitation improving observability. Cable length has a minor effect, though longer cables show slightly increased errors.

Table 3: Estimation RMSE [rad] of payload angles across parameters ( $dt = 2$  s).

|                | $l = 1$ m | $l = 2$ m | $l = 3$ m |
|----------------|-----------|-----------|-----------|
| $m_l = 0.5$ kg | 0.201     | 0.145     | 0.186     |
| $m_l = 1.0$ kg | 0.190     | 0.146     | 0.168     |
| $m_l = 1.5$ kg | 0.179     | 0.141     | 0.163     |

Tracking degradation (Table 4) generally decreases with payload mass. However, results reveal that  $l = 2$  m consistently leads to higher errors, suggesting that dynamic compatibility between the payload and trajectory may outweigh pure underactuation effects.

Table 4: Relative tracking RMSE degradation [%] across parameters ( $dt = 2$  s).

|                | $l = 1$ m | $l = 2$ m | $l = 3$ m |
|----------------|-----------|-----------|-----------|
| $m_l = 0.5$ kg | 4.50      | 5.76      | 5.66      |
| $m_l = 1.0$ kg | 2.29      | 4.97      | 3.37      |
| $m_l = 1.5$ kg | 1.53      | 4.26      | 1.62      |

### 5.3 Real-World Deployment

We deployed the proposed control pipeline on a Tarot T650 UAV platform equipped with an onboard Intel NUC computer, a Pixhawk 4 flight controller, and an Emlid Reach M2 RTK GNSS module to obtain global position measurements (Fig. 7(a)). A video of the real-world experiment is available at <https://mrs.fel.cvut.cz/papers/>



(a) Photo from experiment (b) Payload with RTK GNSS  
Figure 7: Experimental hardware used in the field deployment.

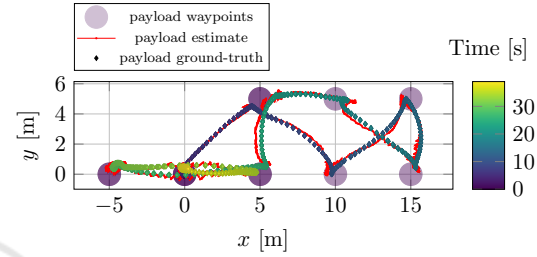


Figure 8: Executed trajectory during the real-world experiment.

uav-with-cable-suspended-payload. The payload consisted of a second Intel NUC, also equipped with an identical RTK GNSS receiver (Fig. 7(b)), enabling direct ground-truth measurements of the payload's 3D position. The payload mass was 1.5 kg and it was suspended using a 2.0 m cable. To ensure accurate data synchronization between the UAV and the payload computers, the two NUCs were physically connected via an Ethernet cable.

The payload was tasked with following a complex reference trajectory (see Fig. 8), discretized with  $dt = 3$  s. To assess the quality of the payload angle estimation, we compared the estimated angles ( $\theta_l, \phi_l$ ) with ground-truth angles reconstructed directly from the RTK GNSS position measurements. Fig. 9 illustrates the comparison across the full trajectory, while Table 5 summarizes estimation statistics for each angle.

Table 5: Estimation accuracy on real-world trajectory

|      | $\theta_l$ [rad] | $\phi_l$ [rad] |
|------|------------------|----------------|
| RMSE | 0.178            | 0.078          |
| STD  | 0.172            | 0.077          |
| Bias | 0.046            | 0.012          |

Estimation error is slightly higher in  $\theta_l$ , which corresponds to the direction of stronger excitation due to trajectory design. Importantly, both estimated angles exhibit very low bias relative to their standard deviation, indicating that the error is primarily due to zero-

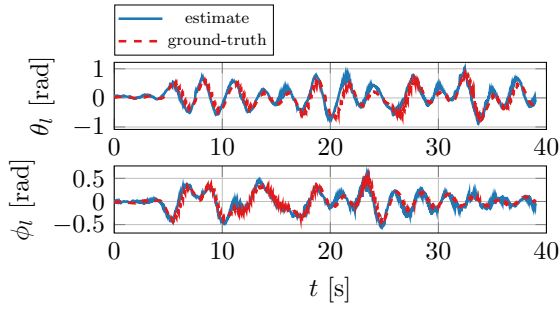


Figure 9: Comparison of estimated and ground-truth payload angles.

mean noise rather than modeling inaccuracies. This supports the validity of the dynamic model introduced in Section 3.

Using these state estimates, the UAV successfully tracked the desired trajectory. Fig. 10 shows the tracking performance over the flight. Quantitative results are given in Table 6, which also presents simulation performance on the same trajectory to allow a sim-to-real comparison.

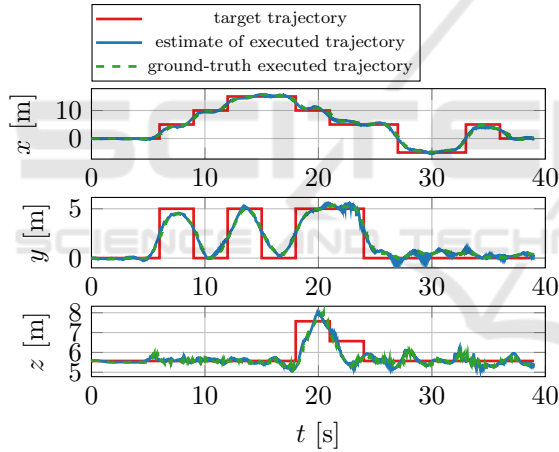


Figure 10: Trajectory tracking performance in real-world flight.

Table 6: Real-world vs. simulation performance comparison.

| Environment     | $\Delta_{\text{RMSE}}$ | Tracking RMSE [m] | Estimation RMSE [rad] |
|-----------------|------------------------|-------------------|-----------------------|
| Simulation      | 1.36                   | 1.563             | 0.134                 |
| Real-world      | 7.26                   | 1.654             | 0.194                 |
| Sim-to-real gap |                        | +6%               | +45%                  |

While estimation RMSE increases by 45% in the real-world setting, tracking RMSE increases by only 6%. This highlights the robustness of the MPC

controller to noisy state estimates. The estimator's low bias ensures that tracking errors do not accumulate, while the controller's incremental structure effectively filters out high-frequency estimation error. These results demonstrate that the full control pipeline—from sparse trajectory planning to state estimation and control—remains performant under real-world conditions, validating the proposed framework as a practical solution for agile payload transport with suspended cables.

## 6 CONCLUSIONS

We presented a complete control framework for real-time trajectory tracking of a suspended payload using an unmanned aerial vehicle equipped solely with standard onboard sensors—namely RTK GNSS and the flight controller's IMU. Unlike many prior approaches relying on external perception systems or additional payload instrumentation, our method operates in a fully onboard, field-deployable setup.

The proposed framework integrates a lightweight linear Kalman filter for state estimation, an incremental MPC for robust control under estimation noise, and a model predictive contouring control (MPCC) planner for smooth, adaptable trajectory generation. Our design leverages the robustness of incremental MPC to allow a bias-optimized estimator without sacrificing stability or tracking accuracy.

Experimental validation in simulation demonstrated that the proposed framework achieves performance close to ground-truth-based control, with tracking errors differing by only a few percent. The controller also showed strong robustness to variations in payload parameters. Furthermore, a field experiment confirmed the practical applicability of the system, achieving performance comparable to simulations and thereby validating the relevance of the simulated evaluations.

To our knowledge, this is the first demonstration of accurate and robust payload position tracking in outdoor environments using only common UAV hardware and onboard sensors. Our approach thus significantly lowers the barrier to deploying such systems in practical scenarios, paving the way for scalable applications in agile transport, sensor placement, and mechanical interaction tasks.

## ACKNOWLEDGEMENTS

This work was supported by the European Union under the project \*Robotics and Ad-

vanced Industrial Production\* (reg. no. CZ.02.01.01/00/22\_008/0004590), by the Czech Science Foundation (GAČR) under research projects no. 23-07517S and no. 24-12360S, and by the CTU grant no. SGS23/177/OHK3/3T/13.

## REFERENCES

- Axehill, D. (2015). Controlling the level of sparsity in MPC. *Systems & Control Letters*, 76:1–7.
- Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M. (2021). The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, 102(1):26.
- C., M. A., K., B., Ramsundar, and S., J. (2024). Path Planning Algorithm for UAV Based Water Quality Monitoring. In *2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, pages 1–5.
- Colomina, I. and Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92:79–97.
- Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L. (2013). Learning swing-free trajectories for UAVs with a suspended load. In *2013 IEEE International Conference on Robotics and Automation*, pages 4902–4909. ISSN: 1050-4729.
- Frison, G. and Diehl, M. (2020). HPIPM: a high-performance quadratic programming framework for model predictive control\*. *IFAC-PapersOnLine*, 53(2):6563–6569.
- Gode, K., Gharat, K., Jogi, H., Sapkal, A., Thakar, R., Vishwakarma, S., Talele, K., and Kulkarni, S. (2024). Multi-Stage UAV-Based System for Scalable and Accurate Crop Health Monitoring. In *2024 IEEE Space, Aerospace and Defence Conference (SPACE)*, pages 652–655.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.
- Li, G., Tunchez, A., and Loianno, G. (2021). PCMPC: Perception-Constrained Model Predictive Control for Quadrotors with Suspended Loads using a Single Camera and IMU. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2012–2018. ISSN: 2577-087X.
- Li, H., Wang, H., Feng, C., Gao, F., Zhou, B., and Shen, S. (2023). AutoTrans: A Complete Planning and Control Framework for Autonomous UAV Payload Transportation. *IEEE Robotics and Automation Letters*, 8(10):6859–6866. Conference Name: IEEE Robotics and Automation Letters.
- Murray, C. C. and Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109.
- Palunko, I., Fierro, R., and Cruz, P. (2012). Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *2012 IEEE International Conference on Robotics and Automation*, pages 2691–2697. ISSN: 1050-4729.
- Qin, S. J. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764.
- Recalde, L. F., Sarvaiya, M., Loianno, G., and Li, G. (2025). ES-HPC-MPC: Exponentially Stable Hybrid Perception Constrained MPC for Quadrotor with Suspended Payloads. arXiv:2504.08841 [eess] version: 1.
- Rego, B. S. and Raffo, G. V. (2016). Suspended load path tracking control based on zonotopic state estimation using a tilt-rotor UAV. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1445–1451. ISSN: 2153-0017.
- Romero, A., Sun, S., Foehn, P., and Scaramuzza, D. (2022). Model Predictive Contouring Control for Time-Optimal Quadrotor Flight. *IEEE Transactions on Robotics*, 38(6):3340–3356.
- Sarvaiya, M., Li, G., and Loianno, G. (2025). HPA-MPC: Hybrid Perception-Aware Nonlinear Model Predictive Control for Quadrotors With Suspended Loads. *IEEE Robotics and Automation Letters*, 10(1):358–365.
- Sikora, T., Markovic, L., and Bogdan, S. (2023). Towards Operating Wind Turbine Inspections using a LiDAR-equipped UAV. arXiv:2306.14637 [cs].
- Sreenath, K., Michael, N., and Kumar, V. (2013). Trajectory generation and control of a quadrotor with a cable-suspended load - A differentially-flat hybrid system. In *2013 IEEE International Conference on Robotics and Automation*, pages 4888–4895. ISSN: 1050-4729.
- Tang, S. and Kumar, V. (2015). Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2216–2222. ISSN: 1050-4729.
- Wang, H., Li, H., Zhou, B., Gao, F., and Shen, S. (2024). Impact-Aware Planning and Control for Aerial Robots With Suspended Payloads. *IEEE Transactions on Robotics*, 40:2478–2497.
- Zhang, Y., Xu, J., Zhao, C., and Dong, J. (2023). IF-Based Trajectory Planning and Cooperative Control for Transportation System of Cable Suspended Payload With Multi UAVs. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 635–642. ISSN: 2153-0866.

## APPENDIX

Table 7 summarizes all key parameters used in the experiments, including those for the Linear Kalman Filter (LKF), the Model Predictive Controllers (MPC and MPCC), solver settings, and physical parameters of the UAV and payload.

Table 7: Summary of parameters used in the framework. Parameters not specified for the MPCC planner are identical to those of the MPC controller.

| Parameter                                                                                                       | Value                                                               |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <b>Linear Kalman Filter</b>                                                                                     |                                                                     |
| Update rate                                                                                                     | 100Hz                                                               |
| Process noise covariance $Q$<br>( $s_{uav}, \mathbf{q}_l, \dot{s}_{uav}, \dot{\mathbf{q}}_l, \theta, \phi, F$ ) | $\text{diag}(1, 1, 1, 30, 30, 100, 100, 100000, 0.1, 0.1, 1, 1, 1)$ |
| Measurement noise covariance $R$<br>( $s_{uav}, \theta, \phi$ )                                                 | $\text{diag}(10, 10, 10, 10, 10)$                                   |
| <b>Incremental MPC Controller</b>                                                                               |                                                                     |
| Update rate                                                                                                     | 100Hz                                                               |
| Horizon length $N$                                                                                              | 50                                                                  |
| Sampling time $\Delta t$                                                                                        | 0.05 s                                                              |
| State penalty matrix $Q$ parameters<br>( $\mathbf{p}_{sl}, \mathbf{p}_u$ )                                      | $[10, 10, 10000, 0, 0, 0.05]^T$                                     |
| Control penalty matrix $R$<br>( $\Delta\theta_u, \Delta\phi_u, \Delta F_u$ )                                    | $\text{diag}(100, 100, 5)$                                          |
| Slack penalty matrix $R_{slack}$                                                                                | $\text{diag}(10, \dots)$                                            |
| Velocity bound                                                                                                  | $10 \text{ m s}^{-1}$                                               |
| Tilt control action bound                                                                                       | 0.75 rad                                                            |
| <b>MPCC Planner</b>                                                                                             |                                                                     |
| Replanning rate                                                                                                 | 1 Hz                                                                |
| Horizon length $N$                                                                                              | 300                                                                 |
| Control penalty matrix $R$<br>( $\Delta\theta_u, \Delta\phi_u, \Delta F_u$ )                                    | $\text{diag}(500, 500, 5)$                                          |
| Tilt control action bound                                                                                       | 0.5 rad                                                             |
| Contouring kernel variance $\sigma$                                                                             | 0.25                                                                |
| <b>Solver</b>                                                                                                   |                                                                     |
| Optimization solver                                                                                             | HPIPM                                                               |
| Formulation                                                                                                     | Partial condensing                                                  |
| Sparsity level                                                                                                  | 10                                                                  |
| <b>Model</b>                                                                                                    |                                                                     |
| UAV mass $m_u$                                                                                                  | 3.5 kg                                                              |
| Payload air drag coefficient $d_l$                                                                              | 0.1                                                                 |
| UAV air drag coefficient $d_{uav}$                                                                              | 0.1                                                                 |
| Flight Controller Unit (FCU) model gains $K_i$<br>( $\theta, \phi, F$ )                                         | $[1, 1, 1]$                                                         |
| FCU model time constants $\tau_i$<br>( $\theta, \phi, F$ )                                                      | $[0.2, 0.2, 0.05]$                                                  |