Science Robotics
AAAS

Supplementary Materials for

Agile and cooperative aerial manipulation of a cable-suspended load

Sihao Sun *et al.*

Corresponding author: Sihao Sun, sihao.sun@outlook.com

Sci. Robot. **10**, eadu8015 (2025)
DOI: 10.1126/scirobotics.adu8015

The PDF file includes:

Methods
Discussion
Figs. S1 to S5
Algorithm S1
Table S1
Legends for movies S1 to S5
References (*43–46*)

Other Supplementary Material for this manuscript includes the following:

Movies S1 to S5

Supplementary Methods

Proof of Smoothness of Quadrotor Trajectories

Proposition 1: *When the i-th cable remains taut, the trajectory of i-th quadrotor, denoted as $\boldsymbol{p}_i(t)$, is C^3 smooth if $\boldsymbol{\lambda}_i$ and $\boldsymbol{\gamma}_i$ defined in Equation 3 are bounded.*

Proof: For the i-th quadrotor, the kinematic constraint (Equation 5) holds when the corresponding cable is taut. Then we take the 3rd-order derivative of Equation 5 to obtain the jerk of the quadrotor

$$\begin{aligned}\ddot{\boldsymbol{v}}_i = \ddot{\boldsymbol{v}} + \boldsymbol{R}(\boldsymbol{q})\Big\{&\boldsymbol{\omega}\times\Big[\dot{\boldsymbol{\omega}}\times\boldsymbol{\rho}_i + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{\rho}_i)\Big]\\ &+ \ddot{\boldsymbol{\omega}}\times\boldsymbol{\rho}_i + \dot{\boldsymbol{\omega}}\times(\boldsymbol{\omega}\times\boldsymbol{\rho}_i) + \boldsymbol{\omega}\times(\dot{\boldsymbol{\omega}}\times\boldsymbol{\rho}_i)\Big\}\\ &- l_i\Big\{\ddot{\boldsymbol{r}}_i\times\boldsymbol{s}_i + 2\dot{\boldsymbol{r}}_i\times(\boldsymbol{r}_i\times\boldsymbol{s}_i) + \boldsymbol{r}_i\times(\dot{\boldsymbol{r}}_i\times\boldsymbol{s}_i)\\ &+ \boldsymbol{r}_i\times\Big[\boldsymbol{r}_i\times(\boldsymbol{r}_i\times\boldsymbol{s}_i)\Big]\Big\},\end{aligned} \tag{S1}$$

where $\dot{\boldsymbol{\omega}}$ is given in the load dynamics (Equation 2); $\ddot{\boldsymbol{v}}$ and $\ddot{\boldsymbol{\omega}}$ are obtained by taking the derivative of both sides of Equation 2:

$$\ddot{\boldsymbol{v}} = -\frac{1}{m}\sum_{i=1}^{n}\Big[\dot{t}_i\boldsymbol{s}_i + t_i(\boldsymbol{r}_i\times\boldsymbol{s}_i)\Big], \tag{S2}$$

$$\ddot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\Big\{-\dot{\boldsymbol{\omega}}\times\boldsymbol{J}\boldsymbol{\omega} - \boldsymbol{\omega}\times\boldsymbol{J}\dot{\boldsymbol{\omega}} + \sum_{i=1}^{n}\Big[\dot{t}_i\boldsymbol{R}^\top\boldsymbol{s}_i + t_i\Big(-\boldsymbol{\omega}\times\boldsymbol{R}^\top\boldsymbol{s}_i + \boldsymbol{R}^\top(\boldsymbol{r}_i\times\boldsymbol{s}_i)\Big)\Big]\times\boldsymbol{\rho}_i\Big\}, \tag{S3}$$

Judging from the load-cable dynamics (Equation 2 and 3), the continuousness of $\ddot{\boldsymbol{v}}_i$ is determined by the highest-order states $\dot{t}_i$ and $\ddot{\boldsymbol{r}}_i$. Therefore, $\ddot{\boldsymbol{v}}_i$ is continuous; namely $\boldsymbol{p}_i$ is C^3-smooth, when $\ddot{t}_i = \gamma_i$ and $\dddot{\boldsymbol{r}}_i = \boldsymbol{\lambda}_i$ are bounded. □

The OCP of the planner takes γ_i and $\boldsymbol{\lambda}_i$ as inputs, which can be bounded by setting input constraints. Hence, the generated trajectories of all quadrotors are smooth up jerk as long as the cable tautness is guaranteed. One step further, once the reference jerk is continuous, we can also obtain a smooth angular velocity reference.

Proposition 2: *The angular velocity of the i-th quadrotor expressed in the inertial frame, denoted by $\boldsymbol{\omega}_i^{\mathcal{I}} \in \mathbb{R}^3$ is C^0-smooth if $\boldsymbol{\lambda}_i$ and γ_i defined in Equation 3 are bounded, and aerodynamic drag $\boldsymbol{f}_{a,i}$ is at least C^1-smooth.*

Proof: To obtain the angular velocity reference of each quadrotor, we need to revisit the translational dynamic equation of the i-th quadrotor

$$\dot{\boldsymbol{v}}_i = \left(T_i\boldsymbol{z}_i + t_i\boldsymbol{s}_i + \boldsymbol{f}_{a,i}\right)/m_i + \boldsymbol{g} \tag{S4}$$

Taking the derivative of both sides of Equation S4, we have

$$\begin{aligned}\boldsymbol{h}_i &\triangleq \boldsymbol{\omega}_i^{\mathcal{I}} \times \boldsymbol{z}_i \\ &= \left[m_i \ddot{\boldsymbol{v}}_i - \dot{T}_i \boldsymbol{z}_i - \dot{t}_i \boldsymbol{s}_i - t_i \left(\boldsymbol{r}_i \times \boldsymbol{s}_i \right) - \dot{\boldsymbol{f}}_{a,i} \right] / T_i\end{aligned} \tag{S5}$$

Since the yaw rate references are zero for all quadrotors, $\boldsymbol{\omega}_i^{\mathcal{I}}$ is perpendicular to $\boldsymbol{z}_i$. Then we can obtain the expression of $\boldsymbol{\omega}_i^{\mathcal{I}}$ by

$$\boldsymbol{\omega}_i^{\mathcal{I}} = \boldsymbol{z}_i \times \boldsymbol{h}_i = \frac{1}{T_i} \boldsymbol{z}_i \times \left[\ddot{\boldsymbol{v}}_i - \dot{t}_i \boldsymbol{s}_i - t_i (\boldsymbol{r}_i \times \boldsymbol{s}_i) - \dot{\boldsymbol{f}}_{a,i} \right] \tag{S6}$$

According to Proposition 1 and Equation 9, $\ddot{\boldsymbol{v}}_i$ is C^0-smooth and T_i is at least C^1 smooth. When γ_i is bounded, $\dot{t}_i$ is also C^0-smooth. Hence angular velocity $\boldsymbol{\omega}_i^{\mathcal{I}}$ is also C^0-smooth when $\boldsymbol{f}_{a,i}$ is at least C^1-smooth. □

If we use Equation S6 to generate the angular velocity reference of each quadrotor, its smoothness is guaranteed through Proposition 2, if we use a smooth drag model (In this work, we assume zero drag for simplicity). The smooth angular velocity reference, used as feed-forward terms by the trajectory tracking controller onboard the quadrotor, guarantees smooth quadrotor behavior, which is particularly crucial during dynamic motions.

Incremental Nonlinear Dynamic Inversion (INDI) Low-Level Controller

The INDI low-level controller of each quadrotor generates rotor speed commands, using the collective thrust command T_{des} and angular acceleration command $\boldsymbol{\alpha}_{\text{des}}$. Here, we summarize the key equations of the INDI controller introduced in our previous work (*29*). In the following context, we denote rotor speed commands as $\boldsymbol{u}_c \in \mathbb{R}^4$, and rotor speed measurement as $\boldsymbol{u}_m \in \mathbb{R}^4$. Note that the following equations apply to a single quadrotor. Hence we omit the subscript i for readability.

The INDI low-level controller employs the following model that maps the rotor speeds to the collective thrust $T \in \mathbb{R}_{\geq 0}$ and body torque $\boldsymbol{\tau} \in \mathbb{R}^3$

$$\begin{bmatrix} T \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{G}_1 \boldsymbol{u}_m^{\circ 2} + \boldsymbol{G}_2 \dot{\boldsymbol{u}}_m \tag{S7}$$

where $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are the control effectiveness matrices with respect to the rotor speeds. Specifically, $\boldsymbol{G}_1$ depends on the shape and size of the quadrotor and aerodynamic coefficients of the propellers. $\boldsymbol{G}_2$ captures the inertial yawing torque due to the acceleration and deceleration of the rotors, which is a function of the moment of inertia of the rotors.

Therefore, once the desired collective thrust and torque $[T_{\text{des}},\ \boldsymbol{\tau}_{\text{des}}]^\top$ is computed, INDI numerically solves the following equation to obtain rotor speed command $\boldsymbol{u}_c$

$$\begin{bmatrix} T_{\text{des}} \\ \boldsymbol{\tau}_{\text{des}} \end{bmatrix} = \boldsymbol{G}_1 \boldsymbol{u}_c^{\circ 2} + \Delta t^{-1} \boldsymbol{G}_2 \left(\boldsymbol{u}_c - \boldsymbol{u}_{c,k-1} \right) \tag{S8}$$

where Δt is the sampling interval of the controller; $\boldsymbol{u}_{c,k-1}$ is the last computed rotor speed command $\boldsymbol{u}_c$. And $\boldsymbol{u}_{c,0} = \boldsymbol{u}_m$ for initialization.

Unlike conventional dynamic inversion, the INDI low-level controller defines the desired body torque in the following incremental form

$$\boldsymbol{\tau}_{\text{des}} = \boldsymbol{\tau}_f + \boldsymbol{J} \left(\boldsymbol{\alpha}_{\text{des}} - \dot{\boldsymbol{\omega}}_f \right) \tag{S9}$$

where $\dot{\boldsymbol{\omega}}_f$ is the angular acceleration obtained by numerically differentiating the filtered gyroscope measurement from the quadrotor. $\boldsymbol{\tau}_f$ is the filtered body torque, which can be calculated using rotor speed measurements leveraging Equation S7, yielding

$$\boldsymbol{\tau}_f = \left[\boldsymbol{G}_1 \boldsymbol{u}_f^{\circ 2} + \Delta t^{-1} \boldsymbol{G}_2 \left(\boldsymbol{u}_f - \boldsymbol{u}_{f,k-1} \right) \right]_{2:4} \tag{S10}$$

where $\boldsymbol{u}_f$ is the low-pass filtered rotor speed measurements. Note that the cutoff frequency of the filter for $\boldsymbol{u}_f$ and $\boldsymbol{\omega}_f$ is the same to synchronize the delay introduced by the low-pass filter on these two measurements. The INDI low-level controller leverages the sensor measurements to effectively capture and compensate for the external torques that are not captured in Equation S7, such as aerodynamic torque, CoG bias of quadrotors, etc.

Supplementary Discussions

Load Pose Estimation Performance

In the above experiments (all real-world experiments shown in this paper), we need information on load pose and twist to achieve dynamic and accurate trajectory tracking. In the state-of-the-art method that includes real-world experiments, additional sensors are required for the load pose estimation. The most commonly used approach is attaching reflective markers on the load to measure its pose from the motion capture system (*12, 16, 21*), or resorting to additional downward-facing cameras and attaching additional circular tags on the load (*11*). However, it is impractical to attach these sensors in the field for day-to-day operations.

In comparison, our algorithm does not need to put any additional sensors on the load, nor does it make any modifications to the quadrotors. In the experiments, we demonstrated that by simply leveraging the IMU on each quadrotor and the dynamic model of the multi-lifting system can provide a sufficiently accurate load pose and twist estimate as well as the cable states to achieve agile pose control. Fig. S1 presents the comparison between the ground truth pose of the load and the estimated pose while tracking reference Fast. Despite the large acceleration of the motion, over 45 degrees of inclination, and continuous yawing motion, our method provided sufficiently high estimation accuracy to achieve closed-loop trajectory tracking. The position estimation RMSE was 0.136 m and the attitude estimation RMSE was 7.5 deg even with the highly dynamic motion of the system.

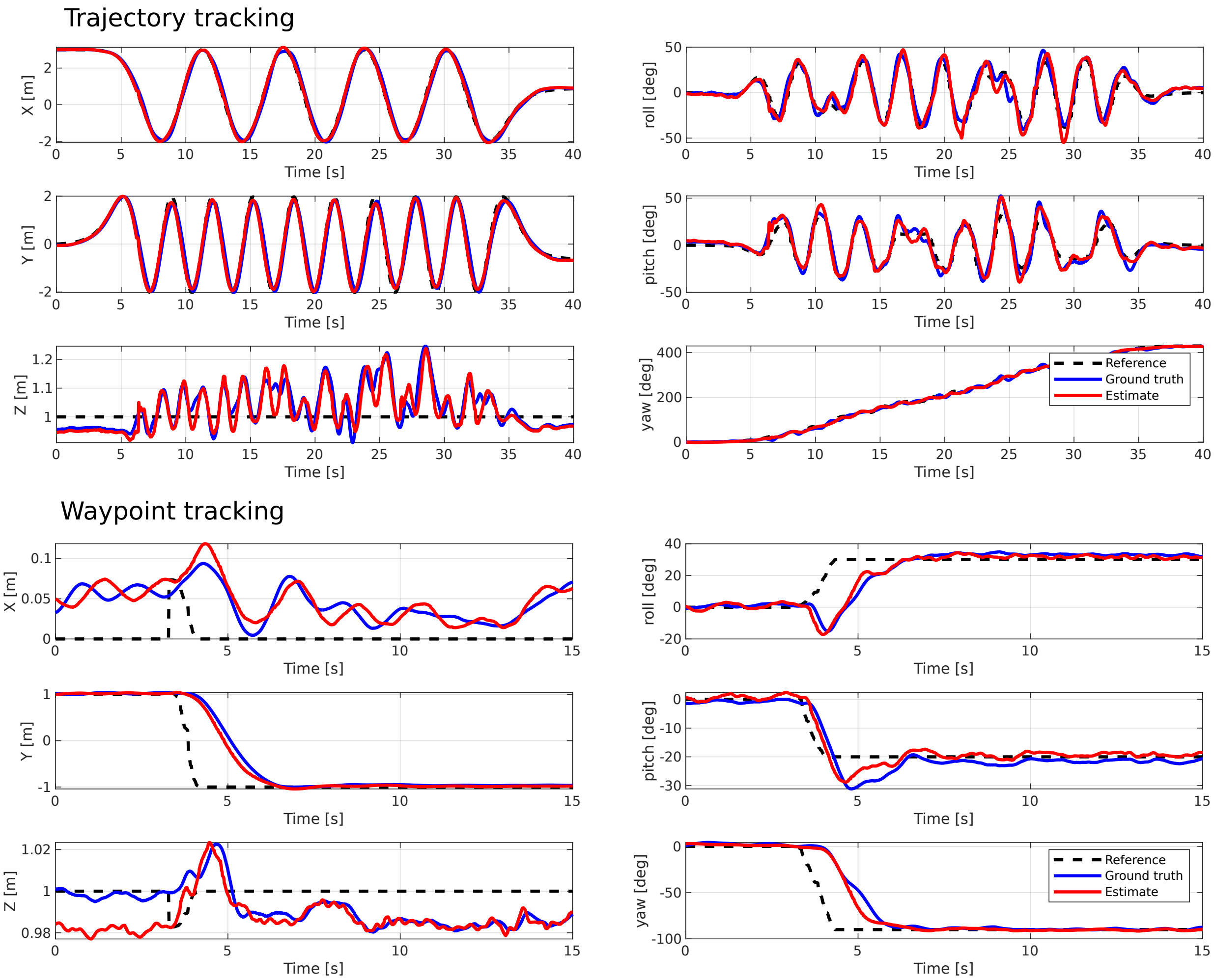

Figure S1: **Load pose estimation and tracking result.** Without adding additional sensors to the load or force sensors to quadrotors, our algorithm could accurately estimate the load's pose for an accurate closed-loop control. **Top**: time history of the position and attitude (Euler angles) estimate in comparison with the ground truth from the motion capture system, as well as the reference load pose while tracking the figure-eight trajectory Fast; **Bottom**: time history of the estimated, the ground truth, and the reference load pose while tracking a setpoint.

Sensitivity to Quadrotor Coordinate Frame Misalignment

We conducted Monte Carlo simulations to examine the effect of misalignment among ground-fixed reference frames (e.g, odometry frames) used by different quadrotors. This issue arises when no global positioning sensors (e.g., motion capture or GPS) are available, such as when using Visual Inertial Odometry (VIO) for state estimation in GPS-denied environments. In this simulation, we assumed that the initial odometry frames of all quadrotors were well-calibrated and aligned with the inertial frame. We then introduced the transformation between the odometry frame and the inertial frame for each quadrotor, in the form of a random-walk process. In this way, we simultaneously simulated the misalignment of reference frames among quadrotors and the pose drift typically observed in visual (-inertial) odometry algorithms.

It is worth noting that extensive research has been conducted on aligning estimated reference frames among multiple robots in the context of multi-robot VIO (e.g., (*43–46*)). A standard approach is to align these coordinate frames in real-time through place recognition to match landmarks seen by different cameras (quadrotors in our case) and by estimating the relative poses between coordinate frames (*45*). The accuracy of relative pose estimation depends on the quality frontend / backend in the VIO algorithms and the quality of the established map. A typical example alignment frequency is 1 Hz (*46*).

Since we were unable to simulate all possible approaches in our study, we only performed a worst-case and best-case scenario analysis. In the worst case, we ran simulations without any real-time alignment. In the best case, we ran simulations and precisely aligned the quadrotors' odometry frames with the inertial frame at 1 Hz. In both cases, we let the system follow the reference trajectory Fast.

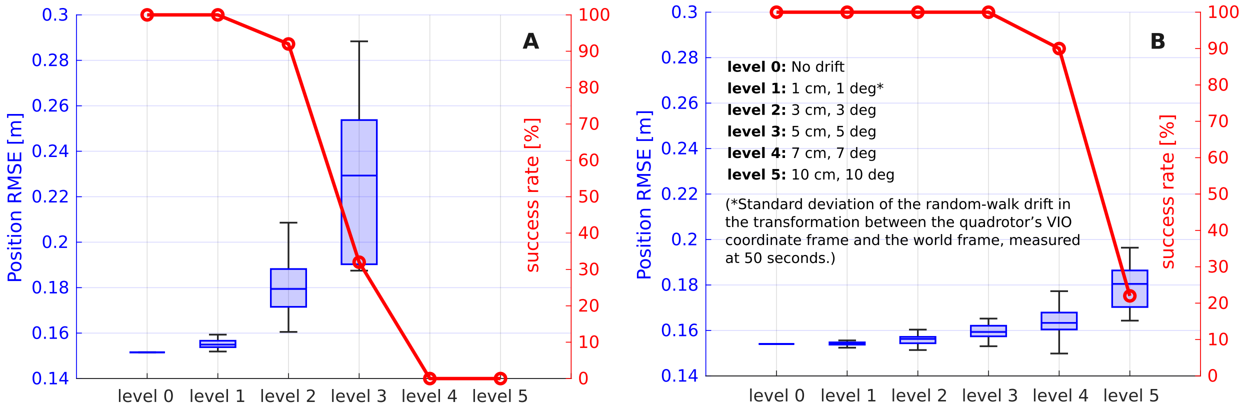

Figure S2: **Simulation result under misalignment between quadrotor reference frames.** The misalignment was induced by random-walk noise at different levels, on the transform between quadrotor VIO frames and the inertial frame. (**A**) The worst case: there was no frame alignment mechanism among quadrotors. (**B**) The best case: the frames were precisely aligned at 1 Hz.

Fig. S2 presents the results of the two cases. It shows that, in the worst case, without any alignment between quadrotors, our method still maintains a success rate of 90% at the 2nd noise level, where the standard deviation of coordinate frame drift is 0.03 m on position and 3 deg on attitude after 50 seconds. We noticed that the attitude misalignment between the coordinate frames is the main cause of failure, as it introduced large position state estimation errors of quadrotors when they follow a trajectory that is far from the origin of the frames. On the other hand, we demonstrated that with an ideal alignment running at 1 Hz, the performance is almost unaffected by the drift of coordinate frames until it reaches noise level 4. These results have demonstrated the strong potential of our method to combine with multi-robot VIO algorithms and deploy in a GPS-denied environment.

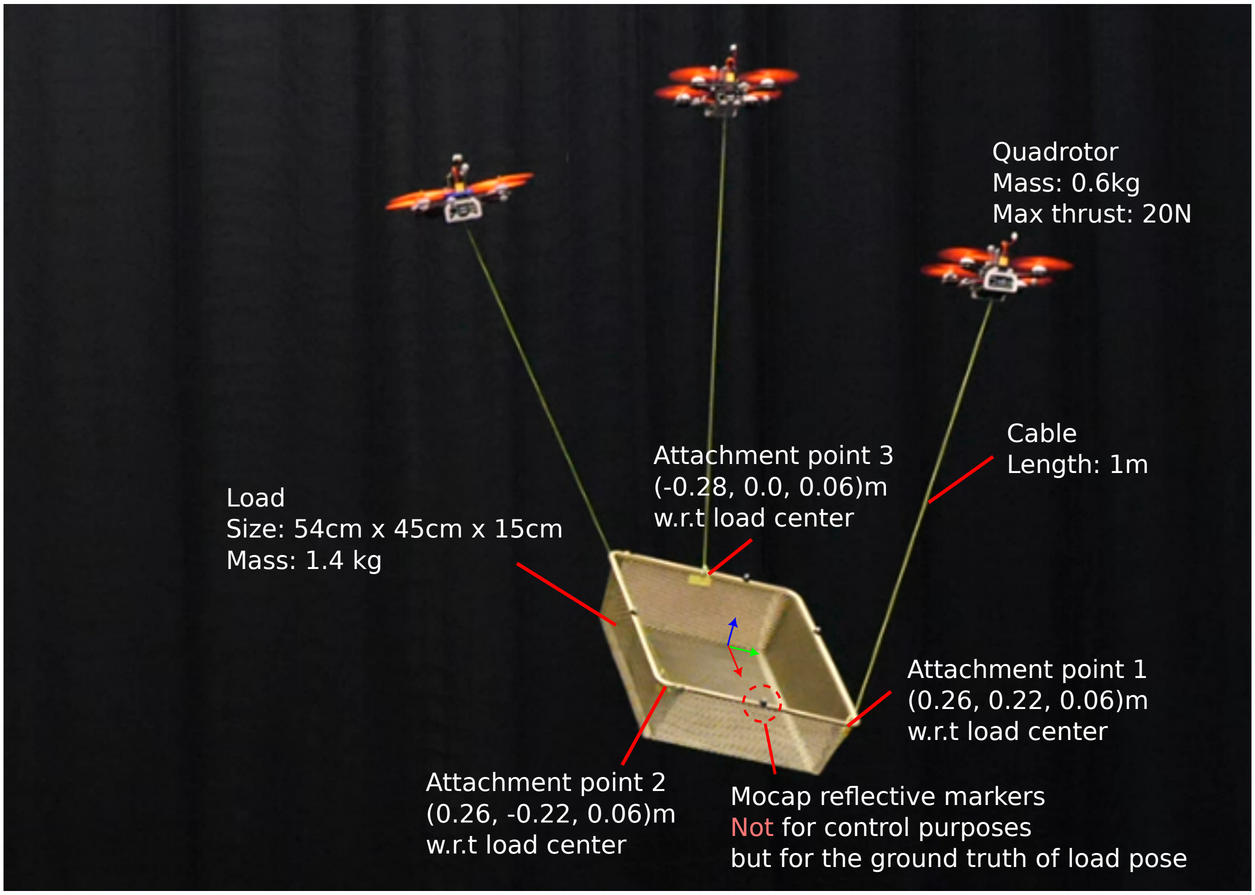

Figure S3: **Experimental setup.** A snapshot of our experiment, together with the parameters of the load, the quadrotors, and the cables.

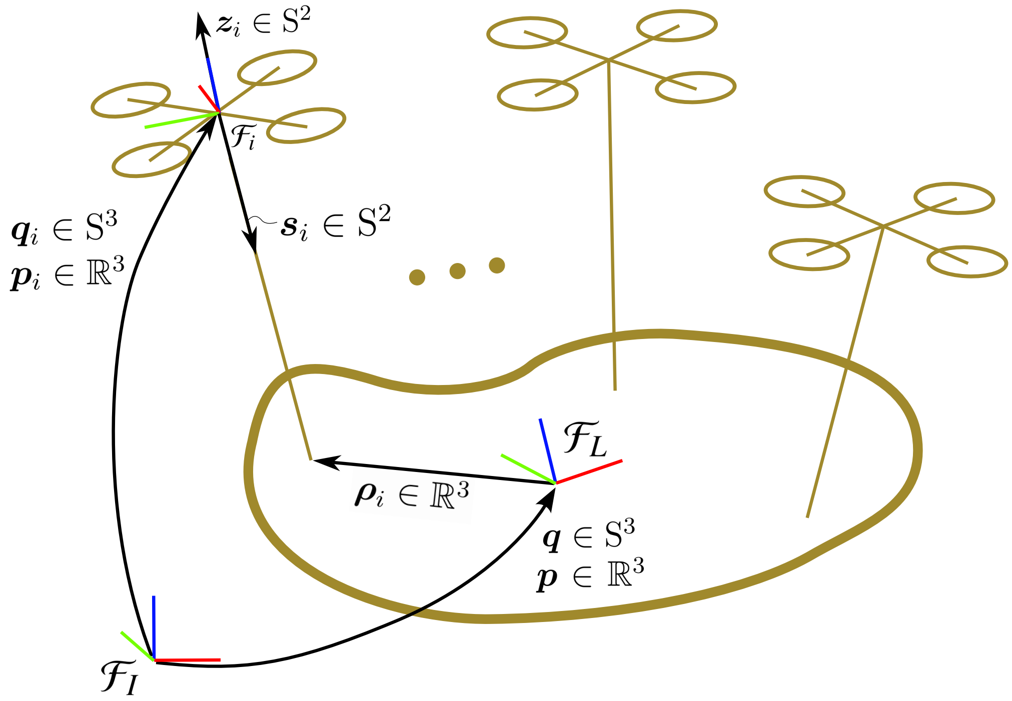
$\boldsymbol{z}_i \in \mathrm{S}^2$
$\mathcal{F}_i$
$\boldsymbol{q}_i \in \mathrm{S}^3$
$\boldsymbol{p}_i \in \mathbb{R}^3$
$\boldsymbol{s}_i \in \mathrm{S}^2$
$\mathcal{F}_L$
$\boldsymbol{\rho}_i \in \mathbb{R}^3$
$\boldsymbol{q} \in \mathrm{S}^3$
$\boldsymbol{p} \in \mathbb{R}^3$
$\mathcal{F}_I$

Figure S4: **Definition of reference frames and symbols.** $\mathcal{F}_I$, $\mathcal{F}_L$, $\mathcal{F}_i$ respectively denote the inertial frame, load-fixed frame, and the *i*-th quadrotor-fixed frame, where the *x*, *y*, *z* are marked in red, green, blue colors respectively.

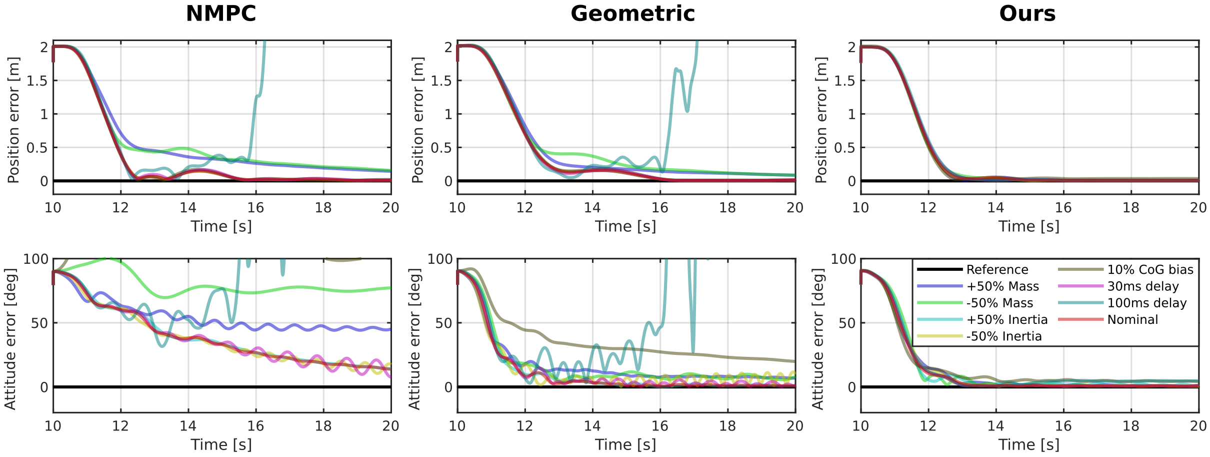

Figure S5: **Step response under load model uncertainties and communication delay**. Simulation result comparing the tracking performance between our method and the two baseline methods (NMPC (*12*) and Geometric (*14*)) under various types of model mismatch on the load, as well as communication delay between the centralized planner and quadrotors. We sent a load reference point at 2 m away along y-axis, and an attitude command of -90 deg, -20 deg, and 30 deg, respectively, on yaw, pitch, and roll. Our method clearly outperforms the two baseline methods in the presence of load model mismatch and communication delays.

Algorithm S1 Iterative Kabsch–Umeyama algorithm to initialize states of the EKF

```
**Input**: $n$, $\boldsymbol{p}_i$, $\boldsymbol{\rho}_i$, $l_i$ for $i \in \{1, 2, \dots, n\}$
Define tolerance and maximum steps for iteration: $tol_{\text{pos}}$, $tol_{\text{att}}$, $iter_{\max}$
Define average cable connection points $\bar{\boldsymbol{\rho}} = \sum_{i=1}^{n} \boldsymbol{\rho}_i / n$, $\boldsymbol{L} = [\boldsymbol{\rho}_1 - \bar{\boldsymbol{\rho}},\ \boldsymbol{\rho}_2 - \bar{\boldsymbol{\rho}}, ..., \boldsymbol{\rho}_n - \bar{\boldsymbol{\rho}}]$
Define initial load pose $\boldsymbol{p} = [0,\ 0,\ 0]^\top$ , $\boldsymbol{R} = \boldsymbol{I}_3$
Initialize the last load pose $\boldsymbol{p}_{\text{last}} = \infty$, $\boldsymbol{R}_{\text{last}} = \boldsymbol{O}_3$
Initial guess of cable directions $\boldsymbol{s}_i = [0,\ 0,\ -1]^\top$ for $i \in \{1, 2, \dots, n\}$
**for** $k = 1, \dots, iter_{\max}$ **do**
  **for** $i = 1, \dots, n$ **do**
    $\boldsymbol{c}_i = \boldsymbol{p}_i + \boldsymbol{s}_i l_i$
  $\bar{\boldsymbol{c}} = \sum_{i=1}^{n} \boldsymbol{c}_i / n$
  $\boldsymbol{C} = [\, \boldsymbol{c}_1 - \bar{\boldsymbol{c}},\ \boldsymbol{c}_2 - \bar{\boldsymbol{c}},\ \dots,\ \boldsymbol{c}_n - \bar{\boldsymbol{c}} \,]$
  $[\boldsymbol{U}, \boldsymbol{V}] = \text{SVD}(\boldsymbol{L}\boldsymbol{C}^\top)$
  $\boldsymbol{R} = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \text{sign}\big(\det(\boldsymbol{V}\boldsymbol{U}^\top)\big) \end{bmatrix} \boldsymbol{U}^\top$        ▷ Estimated load attitude
  **for** $i = 1, 2, \dots, n$ **do**
    $\tilde{\boldsymbol{p}}_i = \boldsymbol{c}_i - \boldsymbol{R}\boldsymbol{\rho}_i$
  $\boldsymbol{p} = \sum_{i=1}^{n} \tilde{\boldsymbol{p}}_i / n$        ▷ Estimated load position
  **for** $i = 1, 2, \dots, n$ **do**
    $\boldsymbol{s}_i = (\boldsymbol{R}\boldsymbol{\rho}_i + \boldsymbol{p} - \boldsymbol{p}_i) / \| \boldsymbol{R}\boldsymbol{\rho}_i + \boldsymbol{p} - \boldsymbol{p}_i \|$        ▷ Estimated cable direction
  **if** $\|\boldsymbol{p} - \boldsymbol{p}_{\text{last}}\| < tol_{\text{pos}}$ **and** $\| \det(\boldsymbol{R}^{-1}\boldsymbol{R}_{\text{last}}) - 1 \| < tol_{\text{att}}$ **then**
    **Break**
  $\boldsymbol{p}_{\text{last}} = \boldsymbol{p}$, $\boldsymbol{R}_{\text{last}} = \boldsymbol{R}$
**Return** $\boldsymbol{p}$, $\boldsymbol{q}(\boldsymbol{R})$, $\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_n$
```

Table S1: **Algebraic expression of load reference trajectories**

Name	$\boldsymbol{p}_x$	$\boldsymbol{p}_y$	$\boldsymbol{p}_z$
Slow	$2.5\cos(0.25t)$	$2\sin(0.5t)$	1.0
Medium	$2.5\cos(0.5t)$	$2\sin(t)$	1.0
Medium Plus	$\cos(t)$	$\sin(2t)$	1.0
Fast	$2.5\cos(t)$	$2\sin(2t)$	1.0

Caption for Movie S1. Video comparing our method with baseline methods in simulation. The video shows the 6-DoF simulation result of our method and the two baseline methods in tracking the two reference trajectories (Fast and Slow). The video is associated with the result given in Table 1.

Caption for Movie S2. Video of flight for obstacle avoidance. The video shows real-world experiments performing two obstacle avoidance tasks: one through a narrow passage between two walls, and the other through a horizontally oriented gap.

Caption for Movie S3. Video of flight with four quadrotors. The video shows the real-world experiment of our method scaling up to a case with four quadrotors following a reference trajectory dynamically.

Caption for Movie S4. Video of flight in windy conditions. The video shows that our method effectively controls the system to follow trajectories under moderate wind disturbances.

Caption for Movie S5. Video showing results with large quadrotor state estimation errors. The video shows real-world experiment and simulation results under different levels of quadrotor state estimation errors.