Green Open Access added to TU Delft Institutional Repository
as part of the Taverne amendment.

More information about this copyright law amendment
can be found at https://www.openaccess.nl.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.

Check for updates

MOTION PLANNING AND CONTROL

Copyright © 2025 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works

Agile and cooperative aerial manipulation of a cable-suspended load

Sihao Sun[1]*, Xuerui Wang[2,3], Dario Sanalitro[4], Antonio Franchi[5,6], Marco Tognon[7], Javier Alonso-Mora[1]

Quadrotors can carry slung loads to hard-to-reach locations at high speed. Given that a single quadrotor has limited payload capacities, using a team of quadrotors to collaboratively manipulate the full pose of a heavy object is a scalable and promising solution. However, existing control algorithms for multilifting systems only enable low-speed and low-acceleration operations because of the complex dynamic coupling between quadrotors and the load, limiting their use in time-critical missions such as search and rescue. In this work, we present a solution to substantially enhance the agility of cable-suspended multilifting systems. Unlike traditional cascaded solutions, we introduce a trajectory-based framework that solves the whole-body kinodynamic motion planning problem online, accounting for the dynamic coupling effects and constraints between the quadrotors and the load. The planned trajectory is provided to the quadrotors as a reference in a receding-horizon fashion and is tracked by an onboard controller that observes and compensates for the cable tension. Real-world experiments demonstrate that our framework can achieve at least eight times greater acceleration than state-of-the-art methods to follow agile trajectories. Our method can even perform complex maneuvers such as flying through narrow passages at high speed. In addition, it exhibits high robustness against load uncertainties and wind disturbances and does not require adding any sensors to the load, demonstrating strong practicality.

INTRODUCTION

Quadrotors stand out for their unparalleled agility, speed, and mobility compared with other robotic systems. These unique capabilities have made them highly suitable for lifting and transporting objects to hard-to-reach locations at high speed (*1*, *2*). However, the payload capacity of a single quadrotor is limited, prompting the exploration of using multiple quadrotors in collaboration to transport (position control) and even manipulate (full pose control) heavy objects, resulting in a multilifting system (*3*–*5*). This strategy has great potential in a wide range of applications requiring heavy object manipulation, such as construction, disaster relief, and agriculture, as well as space exploration missions on Mars and Titan, where aerial vehicles have very limited resources and payload capacities (*6*, *7*). Among the various manipulation mechanisms, the cable-suspended solution stands out for its simplicity and low weight (*8*–*17*). By connecting each quadrotor to a different location on the load through cables, a team of three quadrotors, or more, can change the full pose of the cable-suspended load by adjusting their positions, eliminating the need for additional mechanisms like robotic manipulators.

However, existing cooperative autonomous flight algorithms can only achieve pose control of a cable-suspended object at low speed and low acceleration, greatly limiting their performance and endurance in time-critical missions. The main challenge lies in addressing the complex dynamic coupling and kinematic constraints between the robots, cables, and the load. Early works typically resorted to a quasi-static assumption to neglect the dynamic coupling effects (*18*–*21*) and only considered the kinematic constraints to determine the position and the path of quadrotors to reach the target pose of the load. Despite being simple, failing to account for dynamic coupling leads to undesired swinging motions and cannot guarantee a safe load distribution on each quadrotor.

To account for dynamic coupling effects, recent works used a force-based framework that used the full dynamic model of the cable-suspended multilifting system. Given the pose of the reference load, the methods in this framework calculated a desired wrench (force and torque) that acted on the load through an outer-loop controller, for example, inverse-dynamics control (*22*), nonlinear model-predictive control (NMPC) (*12*, *23*), and geometric control (*14*, *16*, *24*). Then, the commanded wrench was allocated to each cable for their desired tension and directions through the Moore-Penrose inverse of the allocation matrix, which was determined by the connection points on the load (*14*). Some works further exploited the system redundancy in the null space of the allocation matrix, offering capabilities for secondary tasks, such as equal force distribution (*15*) and obstacle avoidance (*12*, *16*), while retaining the collective wrench on the load. Once the required tension and cable directions were determined, a mid-level controller calculated the command thrust and attitude for each quadrotor, which were then executed by an inner-loop attitude controller. Despite considering the dynamic coupling effects, force-based methods are still far from fully exploiting the high agility of the cable-suspended system. Only low-speed (under 1.5 m/s) and low-acceleration (under 0.5 m/s^2) flights have been successfully demonstrated in real-world experiments for load pose control using force-based approaches (*12*, *16*, *24*). Given that a loaded quadrotor with a thrust-to-weight ratio of 1.5 can easily reach an acceleration of more than 4 m/s^2, existing solutions still largely compromise the inherent agility of quadrotors, making the cable-suspended multilifting system far from being able to operate in time-critical missions.

[1]Department of Cognitive Robotics (CoR), Delft University of Technology, Delft, Netherlands. [2]Department of Aerospace Structures & Materials (ASM), Delft University of Technology, Delft, Netherlands. [3]Department of Control & Operations (C&O), Delft University of Technology, Delft, Netherlands. [4]Institut des Systèmes Intelligents et de Robotique (ISIR), Sorbonne University, Paris, France. [5]Robotics and Mechatronics Group (RaM), University of Twente, Enschede, Netherlands. [6]Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome, Rome, Italy. [7]Université de Rennes, CNRS, Inria, IRISA-UMR 6074, Rennes, France.
*Corresponding author. Email: sihao.sun@outlook.com

Downloaded from https://www.science.org at Delft University on November 11, 2025

Here, we identify three major challenges obstructing the existing methods to achieve high agility in reality. First, the aforementioned force-based methods typically use a cascaded control structure, which assumes that the load dynamics are substantially slower than those of the quadrotor. This assumption fails during agile flights, where the load needs to change its pose rapidly. With a cascaded control structure, the outer-loop commands can easily exceed the bandwidth of the inner loops, leading to instability, particularly in the presence of communication delays and actuator dynamics, which are often overlooked in simulation studies. The second challenge is the high reliance on an accurate dynamic model, which is difficult to obtain. The mismatch of the model, especially the mass and inertia of the payload, leads to an error in the thrust command sent to each quadrotor, ultimately causing tracking error and even instability. The third challenge is the reliance on high-frequency load and cable measurements for closed-loop control, requiring additional sensors to be installed onto the load, such as reflective markers for a motion capture system (*15*, *16*, *24*), or installing additional sensors on the quadrotors, such as downward-facing cameras (*11*), cable tension sensors, and cable direction sensors (*25*, *26*). These methods inherently suffer from sensor noise and latency and typically require nontrivial engineering efforts for installation and calibration, making them largely impractical for day-to-day real-world operations.

Trajectory-based framework

In this article, we propose a trajectory-based framework to address the above challenges. Our framework separates the controller into two submodules: an online kinodynamic motion planner and onboard trajectory-tracking controllers. The kinodynamic motion planner considers the whole-body dynamics of the cable-suspended multilifting system, including the force-coupling effects, to generate dynamically feasible trajectories to each quadrotor in a receding-horizon fashion. Then, a trajectory-tracking controller is deployed on board each quadrotor to generate the rotor-speed-level commands to follow the online-generated trajectories while considering the effect of the cable forces.

Specifically, we formulated the kinodynamic motion planner into a finite-time optimal control problem (OCP) that could be effectively solved in tens of milliseconds to generate predicted trajectories with a horizon of 2 s. The OCP formulates safety-related constraints as path constraints, including thrust limitations, cable tautness, interquadrotor collision avoidance, and obstacle avoidance. Because the planner also takes into account the bandwidth and actuation constraints of the inner loop, the assumption of the timescale separation principle required by existing solutions can be circumvented. The generated trajectories include the full state of quadrotors along the horizon; hence, our method allows the planner to run at a considerably lower frequency ($\leq$10 Hz) than the outer-loop controllers of existing works ($\geq$100 Hz). This makes our method substantially more robust against the delay and noise on the state estimate of the load.

We deployed an estimator based on an extended Kalman filter (EKF) leveraging the load-cable dynamic model, quadrotor position and velocity estimates (generally available from an onboard state estimator), and accelerometers on quadrotors to provide satisfactory state estimates of the load and cables for the planner, achieving high-accuracy closed-loop tracking that outperformed state-of-the-art methods. The onboard trajectory-tracking controller uses the incremental nonlinear dynamic inversion (INDI) technique (*27*–*29*), leverages the differential-flatness property of quadrotors (*30*) to follow the reference trajectories, and instantly compensates for the forces from cables using measurements from the inertial measurement unit (IMU). The mismatch in the planned cable tension that stems from the possible mismatch of the load inertia model is thereby effectively compensated for by the trajectory-tracking controller, which eventually ensures high robustness against model uncertainties.

In the remainder of this article, we study the performance of the proposed trajectory-based framework in real-world experiments. The results reveal that the cable-suspended multilifting system controlled by our framework can achieve superior agility in pose control and trajectory following at high speeds (more than 5 m/s) and accelerations (more than 8 m/s^2). It can even rapidly change configurations to avoid obstacles and fly through narrow passages dynamically (Fig. 1). Our method also shows robustness against load model uncertainties, external wind disturbances, and quadrotor state estimation errors. Moreover, the experiments were conducted without adding any sensors to the load to measure its pose, enhancing practicality in day-to-day real-world operations. The results and methods are summarized in Movie 1.

RESULTS

Experimental setup

We tested our algorithm through real-world experiments using three quadrotors to manipulate a payload of 1.4 kg. Each quadrotor, weighing 0.6 kg, experienced a substantial additional force because of the payload. Without loss of generality, we set the cable length to 1 m for all quadrotors. These cables were attached to three distinct points on the rigid-body payload to enable pose control, with the other end connected to each quadrotor 0.03 m below its center of gravity (CoG). The quadrotors were modified from the Agilicious open-source hardware platform (*31*), and each operated its onboard algorithms using a Raspberry Pi 5 mini PC. The centralized planner for our algorithm ran on a laptop at 10 Hz, sending commands to each quadrotor via WiFi.

We used motion capture systems to measure the poses of quadrotors at 100 Hz. These measurements were fused with onboard IMUs through an EKF to obtain state estimates of each quadrotor. On the other hand, the state of the load for closed-loop control was estimated from the quadrotor states, agnostic to the sensors and state estimation algorithms onboard each quadrotor. This also offered high practicality because no sensors were required to be attached to the load. The effect of quadrotor state estimation error, typically seen in field operations without a motion capture system, is analyzed in the "Robustness against quadrotor state estimation error" section. A snapshot of the experimental setup is provided in fig. S3.

Agile pose control

To demonstrate that our method could control the cable-suspended multilifting system to achieve high agility, we tested its performance in tracking figure-eight trajectories with various levels of agility (increasing velocities, accelerations, and jerks), listed in Table 1. The algebraic expressions of the reference trajectories are given in table S1. At the same time, the reference heading also varied over time with a constant yawing rate of 0.25 rad/s. We present the results of our method obtained from real-world experiments. We also present the results of our method compared with two baseline methods

Downloaded from https://www.science.org at Delft University on November 11, 2025

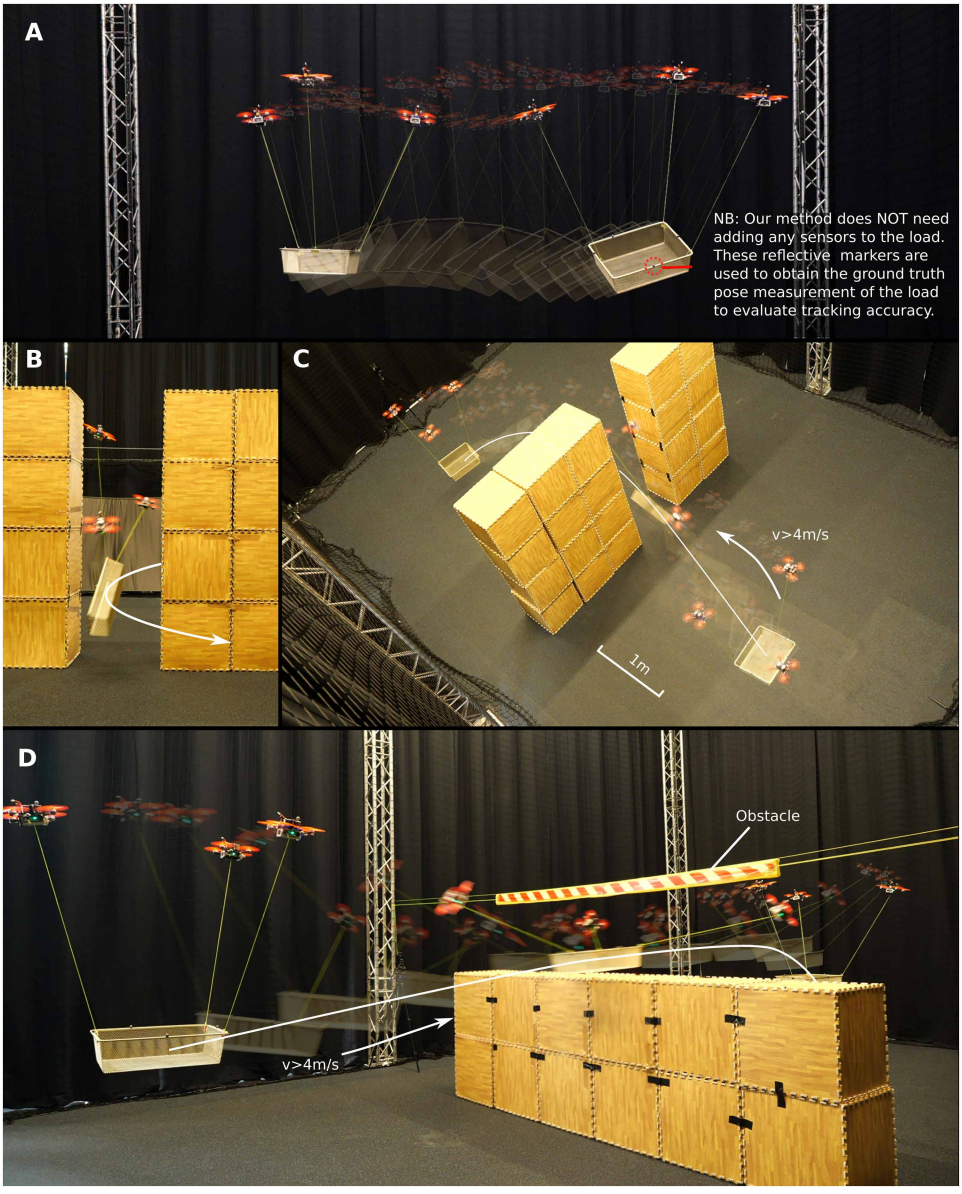


Fig. 1. Snapshot of the real-world experiments. We propose an approach to control a cable-suspended load using multiple quadrotors with high agility. (**A**) Our approach enables agile full-pose control of a cable-suspended load. (**B** to **D**) It enables the quadrotors to dynamically control the load pose and fly through a narrow passage and a horizontally oriented gap. A summary of the experiments is highlighted in Movie 1.

obtained in a simulation environment. The parameters of the quadrotors and loads in these experiments were kept consistent between different approaches to ensure a fair comparison.

We selected two representative state-of-the-art methods, geometric control (*14*, *24*) and NMPC (*12*), that have been successfully demonstrated in real-world experiments as the baseline. These two force-based approaches use a conventional cascaded structure, namely, an outer-loop controller to generate the desired collective load wrench through a geometric controller or NMPC and distribute it to each quadrotor through an inner-loop controller. Table 1 lists the tracking error in these reference trajectories. Both baseline approaches could follow trajectories with relatively low agility [up until v_{max} (maximum velocity) = 2 m/s, a_{max} (maximum acceleration) = 2 m/s^2]. However, they started to fail in following the reference Medium Plus, which involves higher peaks in acceleration and jerk, requiring rapid changes in the cables' directions to produce a fast time-varying wrench on the load. Our method, by contrast, avoids using the cascaded structure used by the baseline methods and can consequently allow fast variation of load pose and cable directions. Therefore, it still successfully followed the reference Medium Plus and even the reference Fast, which has substantially larger accelerations and jerks. A video recording of the comparison in simulation environments is provided in movie S1.

Figure 2A presents the path and pose error while tracking the trajectory Fast that has a v_{max} of 5 m/s and an a_{max} of 8 m/s^2. The reference velocity and acceleration started from zero and gradually reached their maximum values. As the reference velocity increased, both baseline methods failed to track the reference. By contrast, our method succeeded in tracking the reference trajectory, with a position-tracking root mean square error (RMSE) of 0.197 m and an attitude-tracking RMSE of 12.9°, in real-world experiments. The high closed-loop tracking accuracy came from the combined efforts of our controller and estimator. The time history of pose reference, estimate, and ground truth is presented in fig. S1.

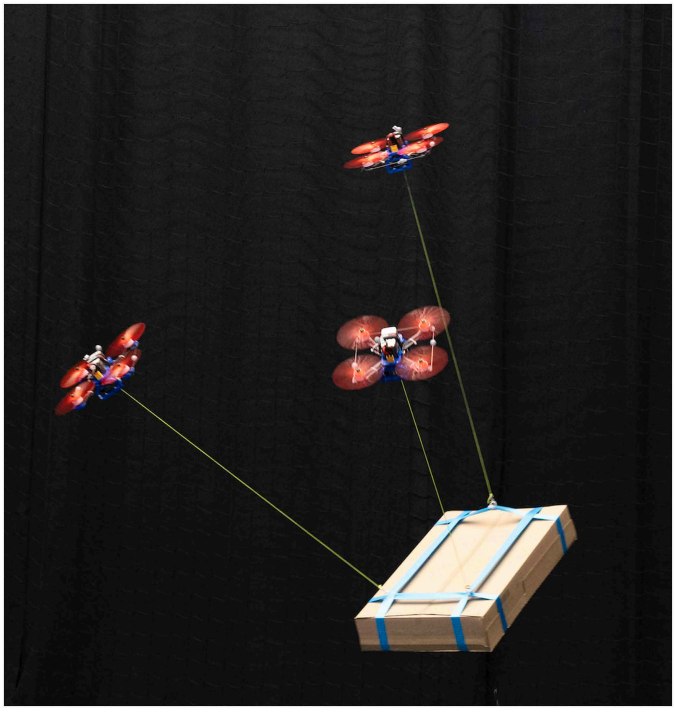
Our algorithm considers dynamic coupling and thrust limits to prevent overloading the quadrotors. In another experiment, we limited the maximum thrust of two quadrotors on the same side of the load from 20 to 11 N and that of the third quadrotor on the opposite side to 15 N, because it needed to carry more lift. Then, we let the system track the reference Fast. Consequently, the reference trajectory became dynamically infeasible for the thrust-limited multilifting system to follow precisely, leading to larger tracking error (0.363 m on position, 18.9° on attitude). The tracking result is presented in Fig. 2B. Despite these thrust limits, our method still enabled the multilifting system to follow the reference trajectory and avoid instability. Our controller modulated trajectory curvature around turns to lower the required acceleration. Throughout this process, our method ensured that the commanded thrust of each quadrotor was maintained within the reduced thrust limits. In addition, the variation in the collective thrust of each quadrotor was notably reduced with a tightened thrust limit, demonstrating that our method automatically adjusted the level of agility to match the capabilities of the quadrotors.

Downloaded from https://www.science.org at Delft University on November 11, 2025

Obstacle avoidance

Our algorithm enabled high-speed obstacle avoidance without designing obstacle-free trajectories in advance, which required hundreds of seconds to generate for a multilifting system composed of more than three quadrotors (*16*). Instead, we surrounded the obstacles with predefined no-fly zones and formulated them as second-order inequality constraints in the OCP. This way, the online-generated reference trajectories for the quadrotors ensured that the quadrotors and the load avoided the no-fly zones, thereby preventing collisions with the obstacles.

We demonstrated the obstacle avoidance capability of our algorithm in two previously unexplored challenging tasks. In both tasks, the multilifting system had to navigate through gaps smaller than its original configuration size by leveraging its kinematic redundancy to reconfigure and squeeze through the narrow passage. In the second task, traversal was performed dynamically, exploiting the momentum gained at high speed, because the system could not counteract gravity with its configuration at the moment of traversal if it attempted to fly statically. A video of the obstacle avoidance experiments is provided in movie S2.



Movie 1. A video of experiments and a brief introduction of the method.

Flying through a narrow passage

In the first scenario, the multilifting system was commanded to fly through a narrow passage with a width of 0.8 m. The width of the entire system in hovering condition was ~1.4 m, which was greater than the size of the gap. We first commanded the quadrotors to carry the load to hover at an initial position. Then, we set a target at 6 m away from the initial position along the *y* axis of the inertial frame. A minimum-snap reference trajectory (*30*) of the load was generated starting from the initial position to the target. However, this reference trajectory intersected with the obstacle. Without an obstacle-avoidance mechanism, the system would have flown directly toward the wall and crashed. We used a constant orientation reference, with the load frame aligned with the world frame.

To guide the system through the opening, we defined two vertical cylinders as no-fly zones, each with a radius of 1.5 m, encompassing the real obstacles. These two no-fly zones created a gap of 0.2 m for the system to pass through, ensuring a 0.3-m clearance from the real obstacles. We selected several reference points on the load and on each quadrotor. The algorithm then ensured that none of these reference points entered the no-fly zones. Specifically, the reference points in this experiment were the center of each quadrotor and the four edges of the payload. At the same time, the error between the actual and the reference pose was minimized in the cost function, encouraging the system to continue moving toward the final target pose.

Figure 3A presents the experimental data, illustrating the maneuvering process. Our proposed algorithm generated predicted trajectories for both the load and the quadrotors at 10 Hz, allowing them to fly through the gap while adhering to the system kinodynamic model. The planner automatically exploited the system's kinematic redundancy to change the cable directions. Because the width of the load (0.54 m) was greater than the gap between the two no-fly zones (0.2 m), the quadrotors managed to steer the load at a steep inclination of ~70° to squeeze through the gap. The distances between quadrotors were also included as constraints in the optimization problem. Hence, their distances were kept greater than a safe margin (0.8 m) throughout the traversal.

Despite successfully avoiding the obstacles, the speed of the maneuver was not compromised. The system reached a top speed of 4 m/s during the fly-through maneuver, with a peak acceleration of more than 5 m/s^2. The load passed through the gap within 1.2 s from the start of the maneuver and eventually stabilized at the target pose after successfully completing the traversal.

Downloaded from https://www.science.org at Delft University on November 11, 2025

Table 1. Position tracking result. Position RMSE in tracking references with different levels of agility. All reference trajectories had a figure-eight shape. Our method substantially outperformed the two baseline methods [geometric (*14*) and NMPC (*12*)], especially in tracking agile trajectories. The baseline methods were tested in a simulation environment, whereas our method was tested in both simulation and real-world experiments.

Name of ref. trajectories	vel_{max} (m/s)	acc_{max} (m/s^2)	$jerk_{max}$ (m/s^3)	Geometric (*14*) (m)	NMPC (*12*) (m)	Ours (m)	Ours (real world) (m)
Slow	1	0.5	0.25	0.032	0.036	0.031	0.102
Medium	2	2	2	0.135	0.159	0.067	0.093
Medium Plus	2	4	8	Crash	Crash	0.062	0.117
Fast	5	8	16	Crash	Crash	0.152	0.197

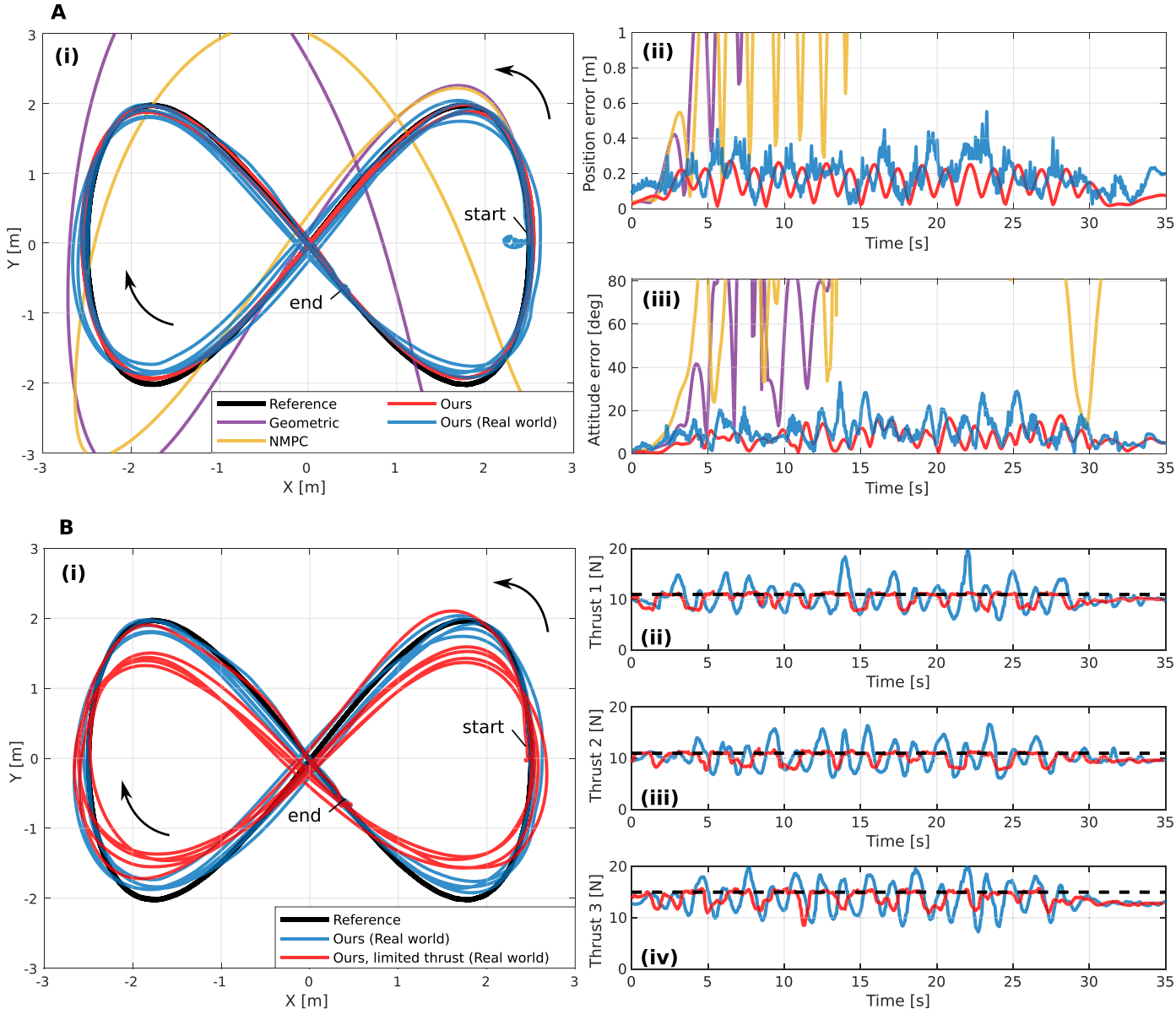


Fig. 2. Performance in tracking the reference Fast. (**A**) Experiment comparing our method against two baseline methods to follow the reference trajectory Fast, a figure-eight trajectory with a maximum speed of 5 m/s and an a_{max} of 8 m/s^2. The detailed expression of the reference is given in table S1. (i) Top view of the flight path of the CoG of the load. (ii and iii) Time history of the RMSE of the load position and attitude tracking error of the load. We used axis-angle representation for the attitude error. (**B**) Experiment comparing our methods with and without tightened thrust limits while tracking the reference Fast. (i) Top view of the flight path of the CoG of the load. Once the maximum thrust was limited, the reference trajectory became dynamically infeasible for the system to follow accurately (red). (ii to iv) The commanded collective thrust of the three quadrotors with the reduced thrust limits (black dashed lines).

Flying through a horizontally oriented narrow gap

In the second task, the quadrotors were commanded to carry the load through a horizontally oriented gap with a height of 0.6 m while the height of the multilifting system in hover was around 1.2 m. The experimental data of this fly-through maneuver are presented in Fig. 3B. In this case, we defined two horizontal cylinders as no-fly zones, each with a radius of 4 m, ensuring that the obstacles were encompassed within the no-fly zones with a minimum safety margin of 0.2 m. The load was initially controlled to hover. Next, a target pose behind the gap was sent to the algorithm, which generated a minimum-snap reference trajectory. This trajectory, however, intersected with one of the obstacles.

Because the vertical size of the gap between the two no-fly zones was only 0.2 m, which was much smaller than the system's hovering height of ~1.2 m when all cables were nearly vertical, our algorithm controlled the quadrotors to spread out and stretch the cables, reducing the overall height of the system to enable it to pass through the gap. During this process, the cable directions changed rapidly from nearly vertical to almost horizontal within 1.2 s. At the moment of traversal, when the cables were nearly horizontal, the vertical components of the cable

Downloaded from https://www.science.org at Delft University on November 11, 2025

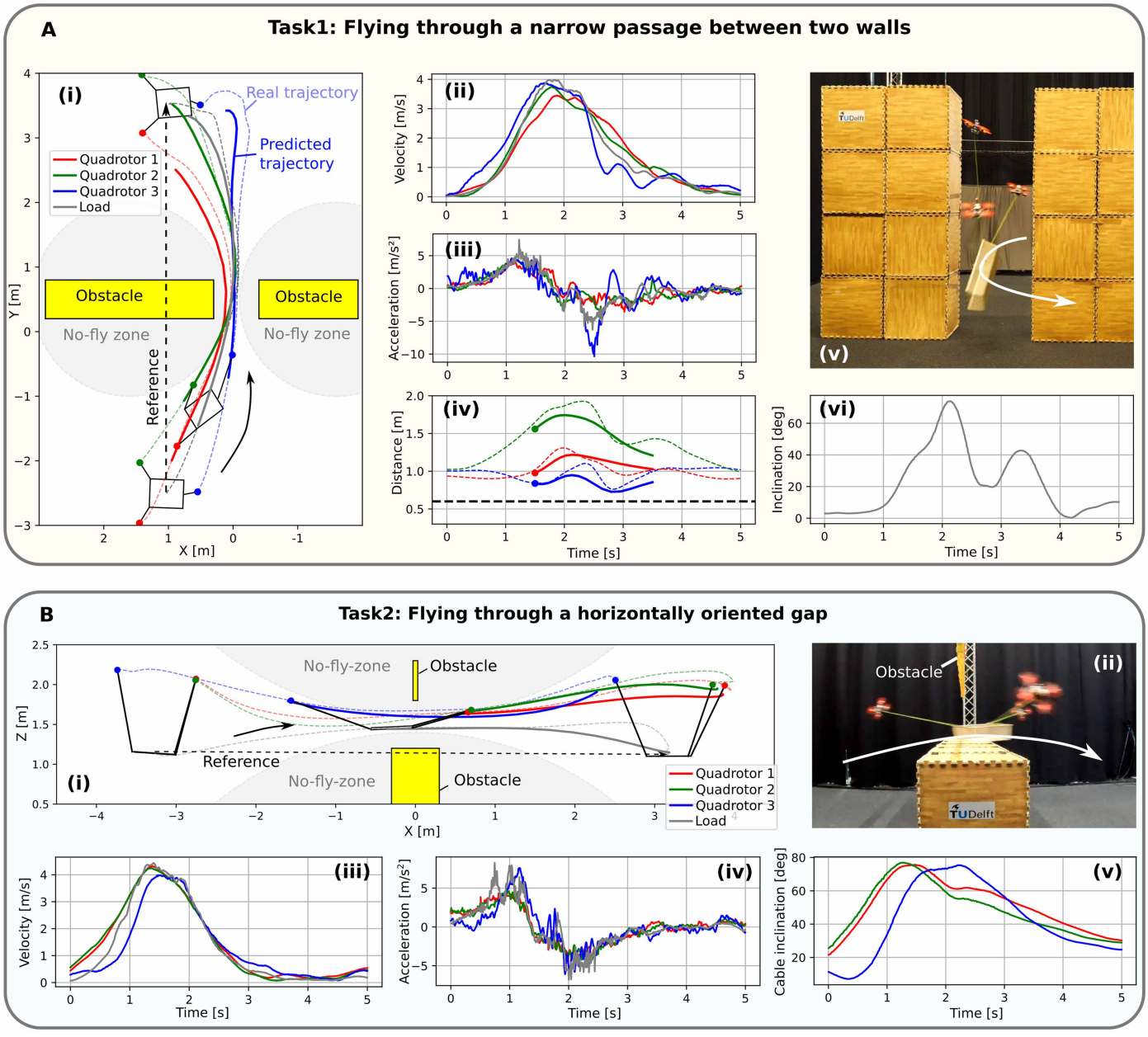


Fig. 3. Obstacle avoidance through dynamic motion. Both tasks were provided with a line segment reference that originally intersected the obstacles. (**A**) Task 1: Flight through a narrow passage between two walls. (i) Top view of the load center and three quadrotors with predicted trajectories at $t = 1.5$ s. (ii and iii) Velocity and acceleration profiles. (iv) Distances between quadrotors. (v) Snapshot of the experiment when the multilifting system flew through the narrow passage. (vi) Load inclination during traversal, defined as the angle between the load-fixed z axis and the world-frame z axis. (**B**) Task 2: Flight through a horizontally oriented narrow gap. (i) Side view of the trajectory and predicted trajectories at $t = 1.5$ s. (ii) Snapshot of the experiment when the multilifting system flew through the horizontally oriented gap. (iii and iv) Velocity and acceleration profiles. (v) Cable inclinations during traversal, defined as the angle between cable directions and the gravity.

tensions could not compensate for the gravity of the load. Therefore, the algorithm induced dynamic motions and took advantage of the momentum of the system to complete the fly-through. In this maneuver, the load reached a v_{max} of 4 m/s and a peak acceleration of more than 7 m/s^2, generating the momentum necessary for a successful traversal.

Robustness

In this section, we investigate our method's robustness against load model uncertainties, external wind disturbances, and quadrotor state estimation errors. These uncertainties are commonly seen in field operations.

Robustness against load model uncertainties

Being robust to uncertainties in the load model is a desirable feature, because it is often impractical to obtain an accurate load model during real-world operations. In Fig. 4A, we compared our method with the two baseline controllers in the presence of various types of model uncertainties on the load in a well-controlled simulation environment that quantified the model mismatch. The results

Downloaded from https://www.science.org at Delft University on November 11, 2025

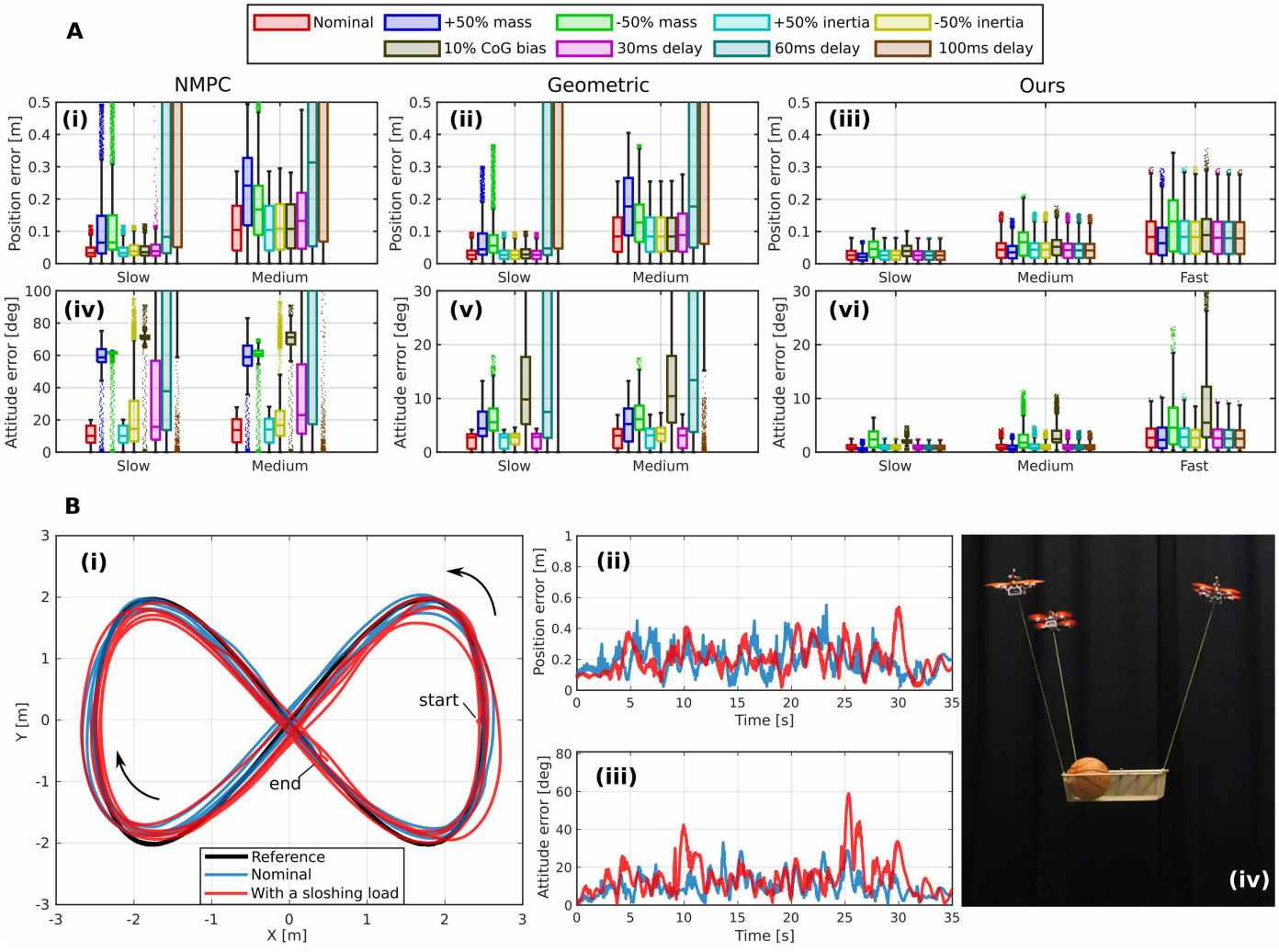


Fig. 4. Test under load model uncertainties and communication delays. (**A**) Tracking performance of our method versus two baseline methods under various load model mismatches and communication delays while tracking references Slow, Medium, and Fast as defined in table S1. The baseline methods failed to follow the reference Fast even without mismatches, whereas our method remained robust. Each box corresponds to one run and summarizes the error at 4500 reference points: median (center line), 25th to 75th percentiles (box), and whiskers extending to the minimum and maximum nonoutlier values; outliers are defined as points lying beyond 1.5 interquartile range (IQR) from the box edges. (i to iii) Position error is in meters. (iv to vi) Attitude error is in degrees, calculated through axis-angle representations. (**B**) Real-world experiment where a 0.6-kg basketball was placed onto the 1.4-kg basket-shaped load and introduced a considerable inertia model mismatch of the load. Our method ran without knowing the presence of the basketball. (i) Top view of the path of the load CoG with and without a sloshing load. (ii and iii) Time history of the position and attitude tracking error. (iv) A snapshot of the experiment.

showed that the baseline controllers were more sensitive to the model mismatch, especially in attitude control. By contrast, our approach sustained more than 50% mass and inertial mismatch in all of the tests. These types of load model mismatch, commonly seen in practice, did not degrade the tracking performance, including the fastest reference trajectory given in Table 1. Only the case with a 10% bias of CoG led to an increase in attitude tracking error of about 5°. We also conducted simulations under step inputs of pose commands, which led to the same conclusion. The results are shown in fig. S5.

We further conducted a challenging real-world experiment, where we placed a 0.6-kg basketball into the original basket-shaped load to introduce sloshing motion (Fig. 4B). This led to a mass mismatch of 43%, given that the mass of the original load was 1.4 kg. The motion of the basketball during flight also caused a pronounced time-varying CoG and inertia of the load if the basketball and the original load were considered as a single unit. We did not modify any parameters in the algorithm; the presence of the basketball was entirely unknown to our method. Despite that, we commanded the multilifting system to follow the trajectory Fast. Because the sloshing inevitably introduced additional swaying motions, particularly during dynamic maneuvers, the tracking error with the sloshing load was slightly larger (0.225 m versus 0.197 m for position RMSE and 18.9° versus 12.9° for attitude). Nevertheless, our algorithm managed to control the multilifting system to follow the reference Fast with an unknown sloshing load, which the baseline methods could not achieve even with a perfect model (see Table 1).

Downloaded from https://www.science.org at Delft University on November 11, 2025

Robustness against wind disturbance

We evaluated the performance of our method in both simulation and real-world experiments under windy conditions. In the simulation, we compared our method against the two baseline methods in the presence of various levels of wind. The system was commanded to hover at a target position. A horizontal wind was then ramped from zero to the designated speed over 5 s and persisted until the end of the simulation. We recorded the position error of the load once the system became stable. We used the quadrotor drag model introduced in (*29*) with parameters identified in real-world experiments. For the load, we used a second-order drag model with a reference surface area of 0.05 m^2 and a drag coefficient of 1.05 [value for a cube (*32*)]. Figure 5A presents the simulation results with wind up to 15 m/s (greater than the maximum wind resistance of most commercial quadrotors), showing that our method notably outperformed the baselines. Note that the wind effect on the cables was neglected.

We also conducted real-world experiments under a wind field of around 5 m/s generated by a 1.5-m-diameter fan. We first commanded the multilifting system (consisting of three or four quadrotors) to follow a straight line across the wind field at only 0.3 m/s, exposing the system directly to the wind for a duration of 5 s. We then evaluated higher-speed flight at 2 m/s by tracking a curved trajectory, which allowed us to assess performance under more dynamic conditions. In addition, we introduced the system into the wind field from an initially windless environment, creating a wind-gust–like scenario that further tested its ability to handle sudden changes in airflow.

Figure 5 (D to F) compares the top view of the trajectories under windy and windless conditions. Our framework enabled the multilifting system to operate at a moderate wind speed of 5 m/s. The disturbances acting on the quadrotors were effectively compensated for by the onboard flight controller. Because the aerodynamic model was not considered by the planner, the disturbance acting on the load led to greater tracking error compared with the case without wind disturbance (0.048 m versus 0.055 m for position RMSE with three quadrotors; 0.048 m versus 0.070 m with four quadrotors). Such tracking errors could be further reduced in future work by identifying and integrating a wind-effect model into the centralized planner. Videos of the above experiments are provided in movie S4.

Robustness against quadrotor state estimation error

Despite having no sensors on the load, our experiments still required state estimates from the quadrotors. Outside the lab environment

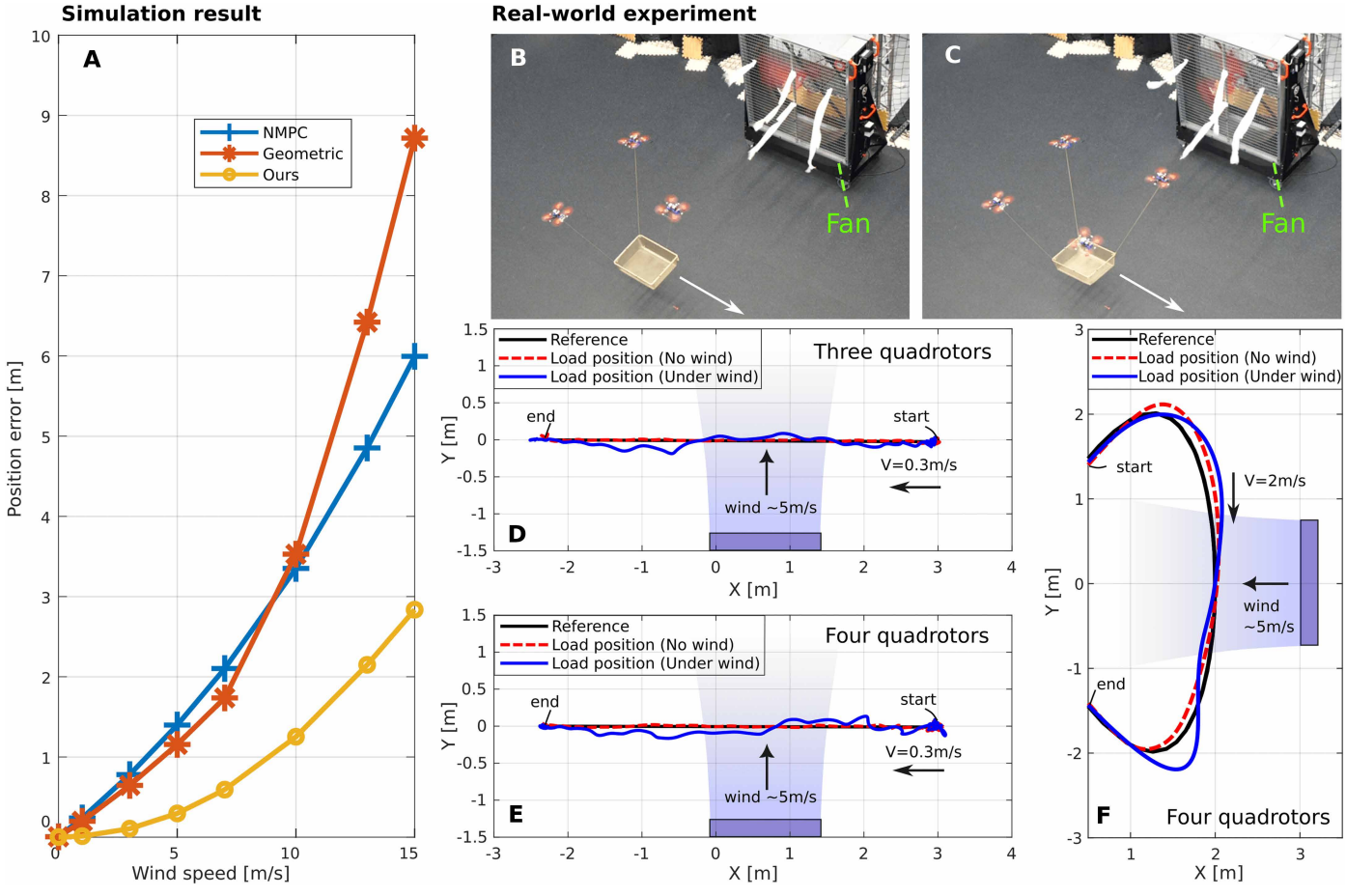


Fig. 5. Test under wind disturbances. (**A**) Comparison of position error between our approach and the baseline methods at different wind speeds in simulation environments. (**B** and **C**) Snapshots of experiments with three and four quadrotors, respectively, under windy conditions generated by a 1.5-m-diameter fan. (**D** and **E**) Real-world experimental data from three or four quadrotors carrying a load to follow a straight line at a speed of 0.3 m/s in a 5-m/s wind field. (**F**) Real-world experimental data with four quadrotors carrying the payload and flying over the wind field while following a curved trajectory at a speed of 2 m/s. The videos of the experiments are provided in movie S4.

Downloaded from https://www.science.org at Delft University on November 11, 2025

without motion capture systems, greater estimation errors could occur because of imperfections in the quadrotor estimation algorithms and sensor noise from the Global Positioning System (GPS), barometer, IMU, or onboard cameras. The quadrotor state estimation errors could deteriorate the closed-loop control performance of our framework. To test the sensitivity to quadrotor state estimation error, we deliberately added noise to the quadrotor positions, attitudes (Euler angle representation), and velocities. We selected random-walk noise on the quadrotor states because it captured both the drift and stochastic fluctuations in the states. The level of random-walk noise was quantified by its SD after 50 s starting from zero. We defined eight noise levels, each including 50 simulations where the system was commanded to follow the reference Fast.

Figure 6A presents the Monte Carlo simulation results at each noise level, including the load position tracking RMSE and the success rate (i.e., no crash occurred). The closed-loop control performance gradually degraded as the noise level increased, which also reduced the success rate. Despite that, the success rate remained more than 95% even under noise level 4, where the SDs of position, attitude, and velocity were, respectively, 0.07 m, 7°, and 0.07 m/s after 50 s of flight.

A real-world experiment was also conducted by adding random-walk pose and velocity noise to the original state estimator output of each quadrotor. Fig. 6 (B to E) presents the estimation errors of all of the quadrotors and the tracking performance compared with the case without added errors. The position errors of the quadrotors exceeded 0.1 m over 90 s of flight, the attitude errors exceeded 10°, and the velocity errors exceeded 0.1 m/s, which was considerably less accurate than a commercial quadrotor operating in the field. Consequently, the position tracking error of the load increased from 0.197 to 0.278 m, and the attitude tracking error of the load increased from 12.9° to 19.4°. Nevertheless, our method successfully controlled the multilifting system to follow the reference Fast without crashing. A video of the flights in simulation and in the real world under quadrotor state estimation error is provided in movie S5.

Computational load and scalability

Our method used a centralized structure, aiming to reach optimality in coordinating all of the agents to manipulate the payload. The centralized structure, particularly the planner, posed a challenge to running in real time when multiple quadrotors were involved. The

Downloaded from https://www.science.org at Delft University on November 11, 2025

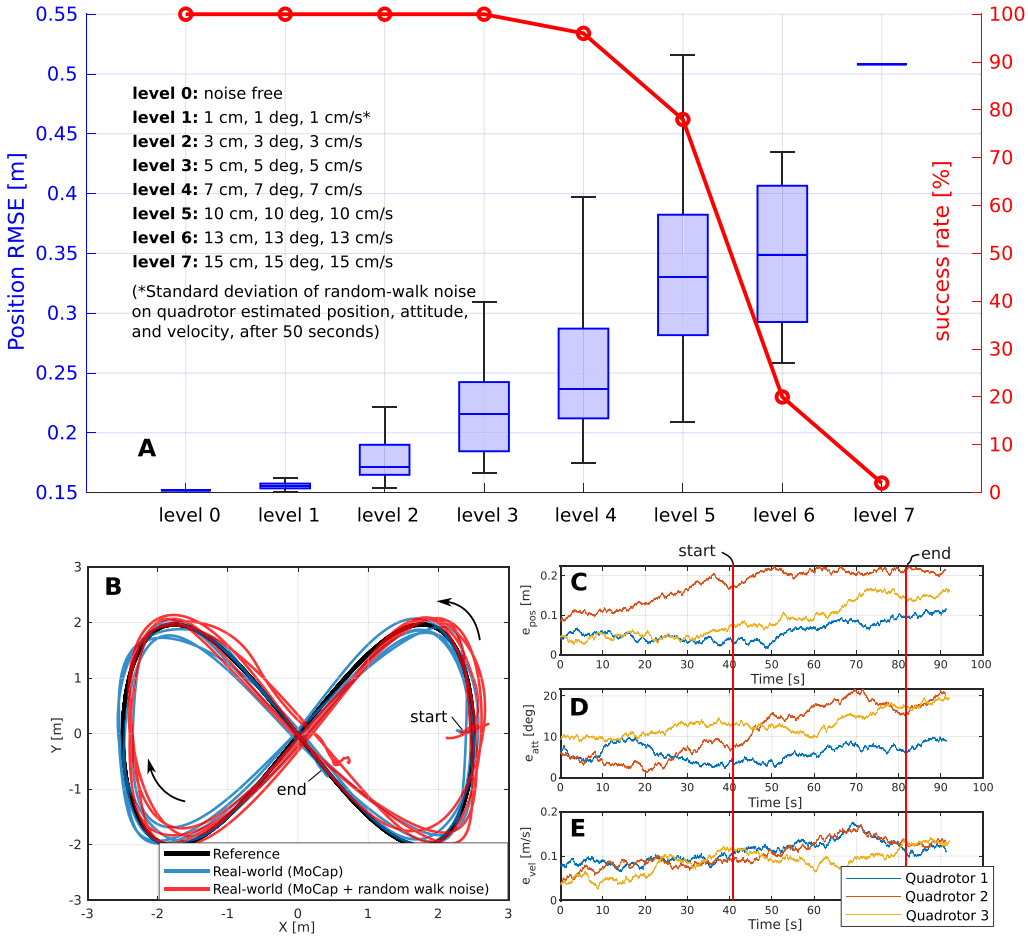


Fig. 6. Test under quadrotor state estimation errors. (**A**) Monte Carlo evaluation under varying levels of state estimation noise (50 runs per level), performed while tracking the Fast reference using noisy position, attitude, and velocity measurements. The red curve shows the success rate. The box plots summarize load position RMSE from the successful runs only: median (center line), 25th to 75th percentiles (box), and whiskers extending to the minimum and maximum values. (**B**) Real-world flight test result under quadrotor state estimation error (level 3). We introduced random-walk noise on the original quadrotor state estimator that used a motion capture system. (**C** to **E**) Time history of the quadrotor position, attitude, and velocity noise introduced in this flight test.

real-world experiments demonstrated that our method successfully coordinated three units in real time. Figure 7A presents the central processing unit (CPU) time of the laptop running the planner (Intel Core i7-13700H) while tracking the trajectory Fast. The average CPU time consumed by the planner was 15.3 ms. Because the planner ran at 10 Hz (i.e., every 100 ms), our algorithm only took 15.3% of the computational budget.

We further explored the potential of scaling up our algorithm to include more units. First, we successfully scaled up our method to four units in real-world experiments, as shown in movie S3. We also conducted simulations of set-point tracking tasks with larger numbers of quadrotors. Without loss of generality, we assumed the load to be centrosymmetric, with cables connected to points evenly distributed around the load center at equal angular intervals depending on the number of quadrotors involved. The mass of the load was scaled proportionally with the number of quadrotors. We set the minimum distance constraint between quadrotors to be 1.6 times the distance between the corresponding contact points on the load, ensuring that constraint violations were possible if it was not imposed.

As a result, Fig. 7B shows that CPU time grew exponentially with the number of quadrotors included in the system. With our hardware setup, the planner supported nine units at 10 Hz without exceeding the compute budget. Figure 7C presents the simulation results with nine quadrotors, including pose tracking, interquadrotor distances, and a three-dimensional (3D) illustration. We believe that with a tailored optimizer for this particular problem, along with additional software and hardware optimizations, our method could be scaled up to substantially more units.

DISCUSSION

Our experiments have shown that the proposed trajectory-based framework can substantially enhance the agility, robustness, and practicality of cable-suspended multilifting systems compared with the state of the art. To consider the dynamic coupling effects, the state-of-the-art framework requires a cascaded structure to coordinate and control multiple quadrotors to collaborate. This conventional structure is built on the principle of timescale separation, assuming that quadrotors can instantly generate a resultant wrench on the load requested by an outer-loop controller, which limits the outer-loop gains to prevent instability and makes the tuning process tedious and task specific (*16*). Consequently, agility (high gain) and safety (low gain) are considered contradictory in the traditional framework.

In contrast, our method does not require the cascaded structure and addresses this issue by solving an online kinodynamic planning problem that considers the whole-body dynamics of the multilifting system. The solution generates the states of the system over a future horizon, offering a predictive capability that allows for the inclusion of safety constraints during agile motion rather than simply limiting gains in the traditional framework. This key component of our solution enables precise pose control, trajectory tracking, and obstacle avoidance at high speeds (more than 5 m/s) and accelerations (more than 8 m/s^2). Although the kinodynamic motion planning accounts for the whole-body dynamics, we have demonstrated that this problem is solvable on a midrange CPU in just a few milliseconds, enabling fast online generation to adjust to disturbances and avoid obstacles and even with the potential to scale up to nine units with the current hardware and software setup.

We sent the predicted trajectories to each quadrotor instead of a single reference point, offering two major advantages. First, deploying a reference sampler together with a robust trajectory-tracking controller on the quadrotors makes our framework substantially more robust to load model uncertainties, communication delays, and external wind disturbances. This enables our algorithm to safely control the cable-suspended multilifting system, even when handling

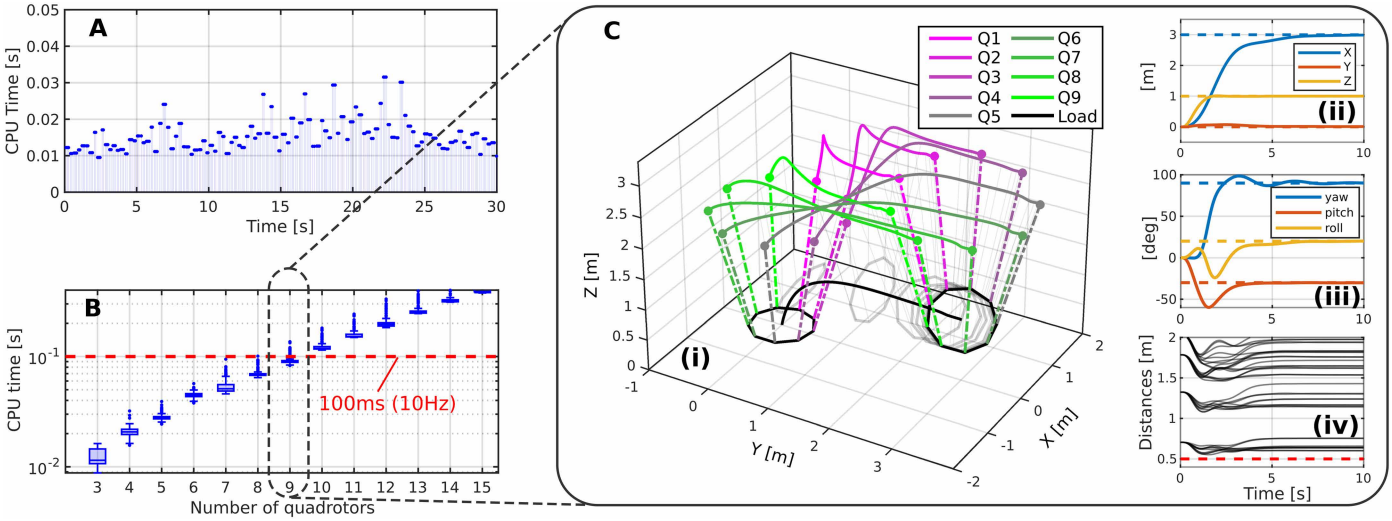


Fig. 7. Computational load and scalability. (**A**) The CPU time to solve each OCP to follow the reference trajectory Fast with three quadrotors in real-world experiments. (**B**) Box plots of the CPU time of planning a trajectory with different numbers of quadrotors in a set-point control task. Each box corresponds to data from 200 simulation steps: median (center line), 25th to 75th percentiles (box), and whiskers extending to the minimum and maximum nonoutlier values; outliers are defined as points lying beyond 1.5 IQR from the box edges. (**C**) Simulation result in a set-point tracking task involving nine quadrotors. (i) 3D plot of the load and the CoG of the nine quadrotors. (ii and iii) Time history of the load pose (solid lines) compared with the reference pose (dashed lines). (iv) Time history of the distances between quadrotors (black solid lines) and the minimum distance allowed by our algorithm (red dashed line).

Downloaded from https://www.science.org at Delft University on November 11, 2025

an unknown-mass sloshing load that brings more than 40% mismatch on the load mass model or under moderate wind breeze at 5 m/s. Second, sending a trajectory allows us to run the planner at a more than 10 times lower frequency than traditional controllers, avoiding reliance on high-frequency measurements from sensors installed on the load.

We see several opportunities for future work. Beyond the quadrotor multilifting problem, the trajectory-based framework has the potential to be applied to a wider range of robotic collaboration challenges, particularly those involving dynamic coupling, agility, and safety constraints. Although our method guarantees high accuracy in the presence of load mismatch, underestimating the load mass and inertia can lead to violations of the maximum thrust constraints. This problem can be alleviated by estimating the inertial properties online (*33*) or using the constrained tightening technique used by a robust nonlinear optimal control framework, such as robust MPC (*34*, *35*). This requires a preestimation of the uncertainties and provides a more conservative, yet safer reference for quadrotors to avoid any violation of their dynamical constraints. Another opportunity is combining our method with onboard perception algorithms in a GPS-denied environment. This requires aligning coordinate frames across quadrotors, which is still a challenge for multiagent perception algorithms, especially in agile flights. To support future works in this direction, we provide a preliminary analysis in Supplementary Discussion to demonstrate the performance of our algorithm in the presence of misalignment between quadrotor coordinate frames. Overall, our work paves the way for future aerial manipulation systems with substantially higher resilience, versatility, and agility to perform complex collaborative tasks in day-to-day operations, from search and rescue to precision delivery in difficult terrains.

MATERIALS AND METHODS

An overview of the method is shown in Fig. 8. The proposed framework incorporates an optimization-based kinodynamic motion planner that generates real-time reference trajectories for the quadrotors. It also includes a time-based sampler and an INDI-based trajectory-tracking controller on board each quadrotor. In addition, the framework uses a centralized EKF to estimate the load pose and cable directions from the quadrotors' position, velocity, and IMU measurements. All modules are model-based and rely on the dynamic model of the cable-suspended multilifting system. The following sections provide a detailed description of each module.

Modeling of cable-suspended multilifting systems

Load-cable dynamic model

The load-cable dynamic model describes the six–degrees-of-freedom motion of the load and the motions of all of the cables attached to the load. Specifically, we used the following definition of the state of the load-cable dynamic model

$$\mathbf{x} = \left[\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{s}_1, \mathbf{r}_1, \dot{\mathbf{r}}_1, \ddot{\mathbf{r}}_1, t_1, \dot{t}_1, \ \ldots, \mathbf{s}_n, \mathbf{r}_n, \dot{\mathbf{r}}_n, \ddot{\mathbf{r}}_n \ t_n, \dot{t}_n\right]^{\top} \quad (1)$$

where n is the number of quadrotors; $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are positions and velocities of the load, respectively; $\mathbf{q} \in \mathbb{S}^3$ is the unit quaternion describing the load attitude; and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the load angular velocity expressed in the load-fixed coordinate frame $\mathcal{F}_L$. The subscript i indicates variables of the cable connected to the ith quadrotor, where $\mathbf{s}_i \in \mathbb{S}^2$ is the cable direction pointing from the quadrotor to the load, $\mathbf{r}_i \in \mathbb{R}^3$ is the cable angular velocity, and $t_i \in \mathbb{R}_{\geq 0}$ is the cable tension. An illustration of the reference frames and some symbols defined above can be found in fig. S4.

We also adopted the following dynamic equations for the load

$$\begin{aligned} &\dot{\mathbf{p}} = \mathbf{v}, \dot{\mathbf{v}} = -\frac{1}{m}\sum\nolimits_{i=1}^{n} t_i \mathbf{s}_i + \mathbf{g}, \\ &\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Lambda}(\mathbf{q})\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \\ &\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \sum\nolimits_{i=1}^{n} t_i \left(\mathbf{R}(\mathbf{q})^{\top}\mathbf{s}_i \times \boldsymbol{\rho}_i\right) \end{aligned} \quad (2)$$

where $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the load inertia, m is the load mass, $\boldsymbol{\rho}_i \in \mathbb{R}^3$ is the displacement of the ith attachment point, expressed in the load frame, and $\mathbf{g}$ is the constant gravity vector. $\boldsymbol{\Lambda}(\mathbf{q})$ represents the quaternion multiplication, and $\mathbf{R}(\mathbf{q}) \in \mathrm{SO}(3)$ is the rotation matrix of the unit quaternion $\mathbf{q}$. To ensure smooth quadrotor reference trajectories up to the jerk level, we used the cable kinematic model, with the third-order derivative of cable angular velocity and the second-order derivative of cable thrust treated as bounded inputs, yielding

$$\dot{\mathbf{s}}_i = \mathbf{r}_i \times \mathbf{s}_i, \ddot{\mathbf{r}}_i = \boldsymbol{\gamma}_i, \ddot{t}_i = \lambda_i, \text{for } i = \{1, \ \ldots, n\} \quad (3)$$

where $\boldsymbol{\gamma}_i \in \mathbb{R}^3$ is the angular snap of cable directions, and $\lambda_i \in \mathbb{R}$ is the second-order derivative of cable tensions. In Supplementary Methods, we proved that the generated quadrotor trajectories are smooth up to the jerk level and also lead to a smooth angular velocity reference with states defined in the load-cable dynamic model in Eq. 1, as long as $\boldsymbol{\gamma}_i$ and λ_i are bounded.

Quadrotor dynamic model

For the ith quadrotor, we described the state space as $\mathbf{x}_i = \left[\mathbf{p}_i, \mathbf{v}_i, \mathbf{q}_i, \boldsymbol{\omega}_i\right]^{\top}$, which corresponds to the CoG position $\mathbf{p}_i \in \mathbb{R}^3$, velocity $\mathbf{v}_i \in \mathbb{R}^3$, unit quaternion rotation $\mathbf{q}_i \in \mathbb{S}^3$, and angular velocity expressed in the quadrotor-fixed coordinate frame $\boldsymbol{\omega}_i \in \mathbb{R}^3$. We used the following rigid-body dynamics to derive the quadrotor equations of motion (*36*)

$$\begin{aligned} &\dot{\mathbf{p}}_i = \mathbf{v}_i, \dot{\mathbf{v}}_i = \frac{1}{m_i}\left(T_i\mathbf{z}_i + t_i\mathbf{s}_i + \mathbf{f}_{a,i}\right) + \mathbf{g}, \\ &\dot{\mathbf{q}}_i = \frac{1}{2}\boldsymbol{\Lambda}\left(\mathbf{q}_i\right)\begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix}, \\ &\mathbf{J}_i\dot{\boldsymbol{\omega}}_{\mathbf{i}} = -\boldsymbol{\omega}_i \times \mathbf{J}_i\boldsymbol{\omega} + \boldsymbol{\tau}_i + \boldsymbol{\tau}_{a,i} \end{aligned} \quad (4)$$

where m_i and $\mathbf{J}_i \in \mathbb{R}^{3\times3}$ are, respectively, the mass and the inertia matrix of the ith quadrotor; $T_i \in \mathbb{R}_{\geq 0}$ is the collective thrust; $\mathbf{z}_i \in \mathbb{S}^2$ is the thrust direction aligning with the z axis of the quadrotor body-fixed frame $\mathcal{F}_i$; $\mathbf{f}_{a,i} \in \mathbb{R}^3$ and $\boldsymbol{\tau}_{a,i} \in \mathbb{R}^3$ are the aerodynamic drag force and torque; and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the control torque generated by the rotors.

Kinematic constraints

We assumed that the cables' tautness would be maintained by our algorithm throughout the operation, even during agile motions. On the ith quadrotor, the position of the cable contact point in the inertial frame is denoted as $\mathbf{p}_i$. Then, the following kinematic constraint between the ith quadrotor and load-cable dynamics holds

$$\mathbf{p}_i = \mathbf{p} + \mathbf{R}(\mathbf{q})\boldsymbol{\rho}_i - l_i\mathbf{s}_i \quad (5)$$

Downloaded from https://www.science.org at Delft University on November 11, 2025

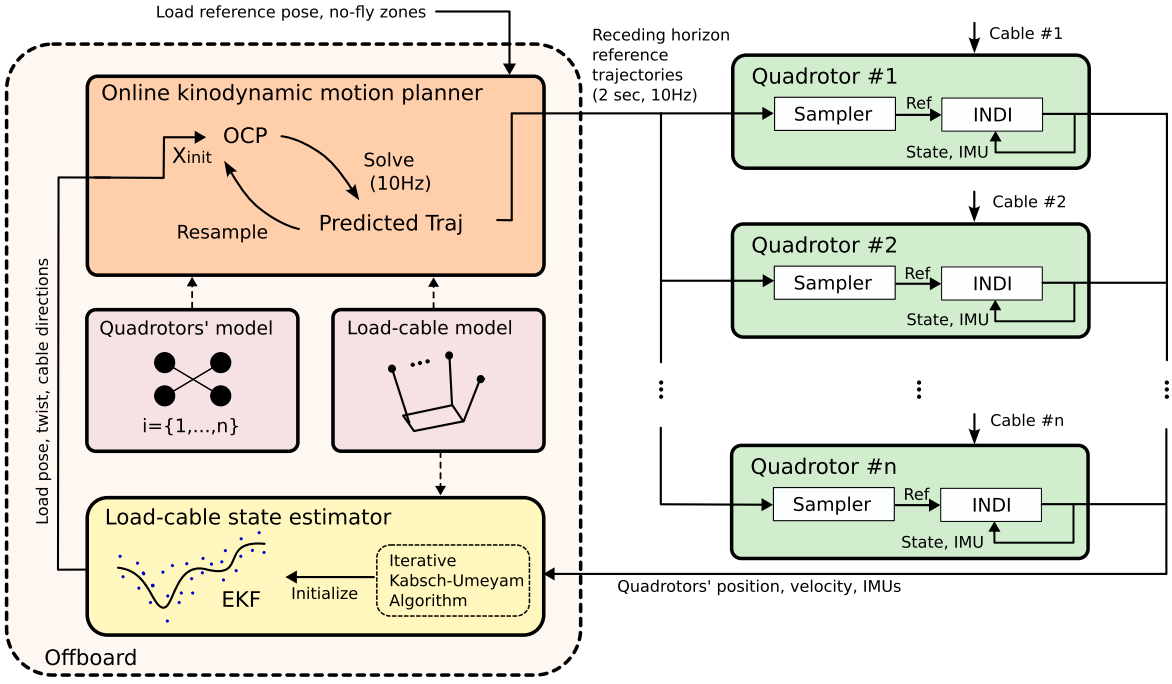


Fig. 8. Method overview. Our framework includes a kinodynamic motion planner solving an OCP online at 10 Hz to generate receding-horizon reference trajectories of quadrotors given load reference pose and predefined no-fly zones. The OCP used the whole-body dynamics of the system, including the quadrotor model and the load-cable model. The load's pose, twist, and cable directions were obtained from an EKF-based estimator. The remaining elements in the initial state of the OCP, namely, the derivatives of the cable directions and tensions, were obtained by resampling the previously generated predicted trajectory to avoid oscillatory motion of the quadrotor when a new reference arrives. The load state estimator fused the load-cable model and the quadrotors' position, velocity, and IMU measurements to obtain estimates of load pose, twist, and cable directions. It was initialized through an iterative Kabsch-Umeyama algorithm given the initial quadrotor states. Onboard each quadrotor, a time-based sampler sampled the received receding-horizon reference trajectory using the current time stamp to generate a single reference point, which was tracked by a trajectory-tracking controller based on the INDI technique that regards the cable tensions as external disturbances and compensates for them using the IMU measurements.

where l_i is the length of the cable.

Online kinodynamic motion planner

Finite-time OCP

Our framework includes a centralized kinodynamic motion planner that generates smooth reference trajectories of all quadrotors in a receding-horizon fashion while considering the dynamic coupling between load and quadrotors. Specifically, the planner is a discretized finite-time OCP, solved by a multiple shooting method (*37*)

$$\begin{aligned} &\min\ J=\sum_{k=0}^{N-1}\left(\|\mathbf{x}_k-\mathbf{x}_{k,\mathrm{ref}}\|_{\mathbf{Q}}^2+\|\mathbf{u}_k-\mathbf{u}_{k,\mathrm{ref}}\|_{\mathbf{R}}^2\right)+\|\mathbf{x}_N-\mathbf{x}_{N,\mathrm{ref}}\|_{\mathbf{P}}^2 \\ &\text{subject to } \mathbf{x}_0=\mathbf{x}_{\mathrm{init}}, \mathbf{x}_{k+1}=\mathbf{f}\left(\mathbf{x}_k,\mathbf{u}_k\right), \\ &\qquad\qquad \mathbf{h}\left(\mathbf{x}_{k+1},\mathbf{u}_k\right)\le 0, k\in\{0,\ \ldots,N\} \end{aligned} \tag{6}$$

where the state equation uses the load-cable dynamic model (Eqs. 2 and 3), and the input is $\mathbf{u}=\left[\gamma_1,\lambda_1,\ \ldots,\gamma_n,\lambda_n\right]$. The quadrotor dynamics, albeit not explicitly included in the state equation for the reason of numerical efficiency, are included in the path constraints, particularly to avoid overloading quadrotors and to perform obstacle avoidance.

The cost function is in a standard quadratic form to minimize the load pose tracking error and control effort for smoothness. The reference states used in the cost function of the OCP are precomputed on the basis of a polynomial load pose reference, with the remaining states derived using the flatness property of the cable-suspended multilifting system (*10*). It is worth noting that in our experiments, the load reference did not consider avoiding obstacles. Instead, we left the planner to decide and generate commands for the quadrotors to carry the load to avoid the obstacles. In other words, the planner had the flexibility to deviate from the reference position or adjust the configurations to satisfy the obstacle avoidance constraints. Naturally, our method could also follow load references generated by a higher-level offline planner [e.g., (*17*)], other than the simple polynomial reference.

We discretized the horizon into $N=20$ nonequidistant segments, with intervals linearly increasing along the horizon. Hence, it ensured higher fidelity of the predicted trajectory in the near future while extending the horizon length without increasing the number of discretization nodes. This OCP was subsequently solved through the sequential quadratic programming (SQP) algorithm in a real-time iteration (RTI) scheme (*38*), implemented using the ACADOS toolkit (*39*). The solution of the OCP was the optimal input $\mathbf{u}_k^*$ and load-cable state $\mathbf{x}_k^*$ along the horizon

$$\begin{aligned} \mathbf{U}^*&=\left[\mathbf{u}_0^*,\mathbf{u}_2^*,\ \ldots,\mathbf{u}_{N-1}^*\right], \\ \mathbf{X}^*&=\left[\mathbf{x}_1^*,\mathbf{x}_2^*,\ \ldots,\mathbf{x}_N^*\right]=\pi\left(\mathbf{U}^*,\mathbf{x}_{\mathrm{init}}\right) \end{aligned} \tag{7}$$

Once the optimal state sequence $\mathbf{X}^*$ was obtained, we converted it to the position, velocity, acceleration, and jerk of the quadrotor through kinematic constraints (Eq. 5) and its derivatives. It is worth noting that the headings of quadrotors, defined as the rotation angle

Downloaded from https://www.science.org at Delft University on November 11, 2025

around $\mathbf{z}_i$, did not affect the thrust directions or the motion of the load. Therefore, we avoided explicitly setting the heading reference for the quadrotors but let the quadrotors maintain a zero yaw rate instead.

The OCP was then solved at a fixed frequency to generate new trajectories online. The initial state $\mathbf{x}_{\text{init}}$ in OCP was provided partially by the load-cable state estimator described in the "Load-cable state estimator" section and partially by resampling the trajectory from the latest OCP solution. Specifically, we used the estimated load pose, twist, and cable directions to renew $\mathbf{x}_{\text{init}}$ to make sure that the trajectories used the up-to-date state of the load for closed-loop control. The other states (cable rate, cable tensions, and their higher-order derivatives) were directly estimated by resampling on the previously generated trajectory. We observed that this treatment had two benefits. First, it ensured smooth transitions between consecutive reference trajectories, avoiding any abrupt and jerky maneuvers by the quadrotors. It also avoided numerical differentiation of state estimator values, which is usually impractical because of the high requirements for accuracy and smoothness in the estimator output.

Path constraints

We included several path constraints $\mathbf{h}(\mathbf{x}) \leq 0$ in the OCP to ensure safety. To avoid overloading each quadrotor, we included the thrust constraint

$$0 \leq T_{i,\min} \leq T_i(\mathbf{x}) \leq T_{i,\max} \tag{8}$$

where $T_i(\mathbf{x})$ is the thrust of each quadrotor as a function of the state of the load-cable dynamic model. Specifically, $T_i(\mathbf{x})$ was obtained through quadrotor dynamics (Eq. 4)

$$T_i(\mathbf{x}) = \| \left(\dot{\mathbf{v}}_i(\mathbf{x}) - \mathbf{g}\right) m_i - t_i \mathbf{s}_i - \mathbf{f}_{a,i} \| \tag{9}$$

where $\dot{\mathbf{v}}_i(\mathbf{x})$ was calculated from the second-order derivative of the kinematic constraints (Eq. 5).

We assumed that the cables would remain taut throughout the operation. Therefore, a cable tension constraint was included

$$0 < t_{\min} \leq t_i \leq t_{\max} \tag{10}$$

To avoid interquadrotor collisions, the minimum distance constraints were also provided for every pair of quadrotors indexed by i and j

$$0 < d_{\min} \leq \| \mathbf{p}_i(\mathbf{x}) - \mathbf{p}_j(\mathbf{x}) \| \tag{11}$$

where $d_{\min}$ is the predefined minimum distance and $\mathbf{p}_i(\mathbf{x})$ and $\mathbf{p}_j(\mathbf{x})$ are the positions of the ith and the jth quadrotor.

We also established several control points on the system to ensure it avoided obstacles. Without loss of generality, we used the CoG of each quadrotor together with the attaching points on the load. For each obstacle and each control point denoted by $\mathbf{p}_c(\mathbf{x})$, the following constraint ensured that none of the control points entered the no-fly zone encompassing the obstacle

$$d_{o,\min}^2 \leq \left(\mathbf{p}_c(\mathbf{x}) - \mathbf{p}_o\right)^\top \mathbf{C} \left(\mathbf{p}_c(\mathbf{x}) - \mathbf{p}_o\right) \tag{12}$$

where $\mathbf{C} \in \mathbb{R}^{3\times3}$ is a diagonal matrix controlling the shape of the no-fly zone, $\mathbf{p}_o$ is the center of the no-fly zone, and $d_{o,\min}$ is the safe distance from the control points to the center. The position and shape of the obstacle can be determined either offline or detected online. In the case when the obstacle is detected online, we can set up the problem to support a large number of obstacles and set the inactive obstacle constraints with a zero radius or place them far from the current positions. Once a new obstacle is detected, we activate one of the reserved inequality constraints by adjusting its parameters to match the size and location of the detected obstacle.

Last, to ensure the bounded input to the OCP, the control input constraint $\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$ was imposed in the experiments. It is worth noting that all path constraints were inequality constraints and handled using slack variables to ensure problem feasibility.

Load-cable state estimator

To update $\mathbf{x}_{\text{init}}$ in the OCP of the planner, the load pose, twist, and cable directions must be estimated in real time. Instead of relying on additional sensors such as downward-facing cameras (*11*) or adding motion capture markers on the load (*24*) in the state-of-the-art approaches, our proposed estimator uses only the quadrotors' states and the load-cable dynamics. This eliminates the need for any hardware modifications.

We chose EKF to solve this state estimation problem because of its simplicity and computational efficiency. We omit the detailed steps in EKF and only describe the selections of states, measurements, models, and initialization.

The state vector of EKF comprised pose and twist of the load, as well as the positions and velocities of all quadrotors, namely, $\hat{\mathbf{x}} = \left[\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{p}_1, \mathbf{v}_1, \ldots, \mathbf{p}_n, \mathbf{v}_n\right]^\top$. State prediction was performed using the load dynamics and quadrotor dynamics (Eqs. 2 and 4). The cable directions to solve the load dynamic were obtained through the kinematic constraint (Eq. 5). The cable forces in these equations were estimated through a spring-damper model, i.e., $t_i = k_{\text{stiff}} d_i + k_{\text{damp}} \dot{d}_i$, where d_i is the distance between the position of the ith quadrotor and its connection point on the load, namely, $d_i = \| \mathbf{p}_i - \mathbf{R}(\mathbf{q})\boldsymbol{\rho}_i - \mathbf{p} \|$; k_{stiff} and k_{damp} are positive coefficients.

The EKF took the cables' directions, together with the quadrotors' positions and velocities as measurements, namely, $\tilde{\mathbf{y}} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{p}}_1, \tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{s}}_n, \tilde{\mathbf{p}}_n, \tilde{\mathbf{v}}_n]^\top$. The quadrotor positions and velocities, and their covariances, were obtained directly from their onboard state estimators. The cable directions were obtained indirectly from the accelerometer sensor that is commonly available on a quadrotor drone. Given that the accelerometer directly measures the specific force (the mass-normalized force excluding gravity), it captures the combined forces, including the cable tension, aerodynamic drag, wind force, and rotor thrusts. For each quadrotor, we identified a collective thrust model, $\overline{T}_i$, and a drag model, $\bar{\mathbf{f}}_{a,i}$, with the following expressions

$$\overline{T}_i = \sum_{j=1}^{4} c_t \omega_{j,i}^2, \quad \bar{\mathbf{f}}_{a,i} = \mathbf{R}\left(\mathbf{q}_i\right) \mathbf{D}_a \mathbf{R}\left(\mathbf{q}_i\right)^\top \mathbf{v}_i \tag{13}$$

where c_t is the thrust coefficient of the rotors, $\omega_{j,i}$ is the rotor speed, and $\mathbf{D}_a \in \mathbb{R}^{3\times3}$ is the aerodynamic coefficient matrix (*40*). According to the quadrotor dynamics (Eq. 4), subtracting them from the accelerometer readings provides the force vector from cables. Then, the cable directions were approximated by

$$\tilde{\mathbf{s}}_i = \left(m_i \mathbf{a}_i - \overline{T}_i \mathbf{z}_i - \bar{\mathbf{f}}_{a,i}\right) / \| m_i \mathbf{a}_i - \overline{T}_i \mathbf{z}_i - \bar{\mathbf{f}}_{a,i} \| \tag{14}$$

where $\mathbf{a}_i = \dot{\mathbf{v}} - \mathbf{g}$ is the unbiased accelerometer measurement.

We observed that the covariance matrix of the cable direction was challenging to determine directly from the accelerometer properties,

Downloaded from https://www.science.org at Delft University on November 11, 2025

because it depends on the accuracy of $\overline{T}_i$ and $\bar{\mathbf{f}}_{a,i}$ and is also affected by wind disturbance. Therefore, this covariance was tuned experimentally. Underestimating the covariance resulted in a noisy load-pose estimate, whereas overestimating it produced an excessively free motion in the estimated load state.

The EKF was initialized with a first-order guess of load pose and twist and cable directions. We assumed a static initial load state, namely, a zero twist. As for the load pose and cable directions, we propose an algorithm to provide a guess iteratively through the Kabsch-Umeyama algorithm (*41*). The details of the algorithm are provided in algorithm S1.

Trajectory-tracking controller on quadrotors

In our setup, every 100 ms, the most recently generated reference trajectories by the planner were sent to each quadrotor and then followed by a differential flatness–based trajectory-tracking controller deployed on board and modified from (*29*). Because the trajectory-tracking controller operated at a higher frequency (300 Hz) than the interval between nodes in the reference trajectory, a time-based sampler was implemented to generate high-frequency reference states by linearly interpolating between the discretized nodes of the reference trajectory. The sampler continued to sample along the reference trajectory until a new reference was received.

The onboard trajectory-tracking controller then computed the thrust command, including magnitude $T_{i,\mathrm{des}}$ and direction $\mathbf{z}_{i,\mathrm{des}}$, through the following PD controller

$$T_{i,\mathrm{des}}\mathbf{z}_{i,\mathrm{des}}/m_i = \mathbf{K}_p\left(\mathbf{p}_{i,\mathrm{ref}}-\mathbf{p}_i\right)+\mathbf{K}_v\left(\mathbf{v}_{i,\mathrm{ref}}-\mathbf{v}_i\right)+\dot{\mathbf{v}}_{i,\mathrm{ref}}+\mathbf{f}_{\mathrm{ext}}/m_i \quad (15)$$

where $\mathbf{K}_p \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_v \in \mathbb{R}^{3\times3}$ are positive definite gain matrices; $\mathbf{f}_{\mathrm{ext}}$ represents external forces on the quadrotor, excluding thrust and gravity, namely, cable tension, aerodynamic drag, and wind. We estimated external forces using the accelerometer on the quadrotor through the relationship $\mathbf{f}_{\mathrm{ext}} = m_i\mathbf{a}_{i,\mathrm{filtered}} - \mathbf{f}_{i,\mathrm{filtered}}$, where $\mathbf{a}_{i,\mathrm{filtered}}$ is the unbiased and low-pass–filtered accelerometer measurement and $\mathbf{f}_{i,\mathrm{filtered}}$ is the current collective thrust vector denoised with the same filter. The collective thrust vector $\mathbf{f}_i$ is calculated using rotor speed measurements and the collective thrust model (Eq. 13). Then, we used a tilt-prioritized attitude controller (*42*) to generate the angular acceleration command $\boldsymbol{\alpha}_{i,\mathrm{des}}$ from the desired attitude $\mathbf{z}_{i,\mathrm{des}}$, the reference jerk, and the zero yaw rate reference.

The angular acceleration and force commands were subsequently allocated to rotor speed commands through an INDI inner-loop controller, which is a sensor-based adaptive controller robust against external torque disturbances such as aerodynamic torque, motor differences, and quadrotor CoG bias. Last, the rotor speed commands generated by INDI were sent to the electronic speed controllers through the DShot protocol. We refer interested readers to (*31*) for further details about the hardware implementations. We also provide key equations for INDI in Supplementary Methods.

Supplementary Materials

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

REFERENCES AND NOTES

1. D. Hanover, P. Foehn, S. Sun, E. Kaufmann, D. Scaramuzza, Performance, precision, and payloads: Adaptive nonlinear MPC for quadrotors. *IEEE Robot. Autom. Lett.* **7**, 690–697 (2021).
2. A. Saviolo, G. Loianno, Learning quadrotor dynamics for precise, safe, and agile flight control. *Annu. Rev. Control* **55**, 45–60 (2023).
3. G. Loianno, V. Kumar, Cooperative transportation using small quadrotors using monocular vision and inertial sensing. *IEEE Robot. Autom. Lett.* **3**, 680–687 (2017).
4. A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, J. Nieto, "Collaborative transportation using MAVs via passive force control" in *IEEE International Conference on Robotics and Automation* (IEEE, 2017), pp. 5766–5773.
5. A. Tagliabue, M. Kamel, R. Siegwart, J. Nieto, Robust collaborative object transportation using multiple MAVs. *Int. J. Robot. Res.* **38**, 1020–1044 (2019).
6. J. Balaram, M. Aung, M. P. Golombek, The ingenuity helicopter on the perseverance rover. *Space Sci. Rev.* **217**, 56 (2021).
7. R. D. Lorenz, E. P. Turtle, J. W. Barnes, M. G. Trainer, D. S. Adams, K. E. Hibbard, C. Z. Sheldon, K. Zacny, P. N. Peplowski, D. J. Lawrence, M. A. Ravine, T. G. McGee, K. S. Sotzen, S. M. MacKenzie, J. W. Langelaan, S. Schmitz, L. S. Wolfarth, P. D. Bedini, Dragonfly: A rotorcraft lander concept for scientific exploration at Titan. *J. Hopkins APL Tech. Dig.* **34**, 14 (2018).
8. P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload" in *Robotics: Science and Systems* (RSS Foundation, 2017).
9. F. Panetsos, G. C. Karras, K. J. Kyriakopoulos, GP-based NMPC for aerial transportation of suspended loads. *IEEE Robot. Autom. Lett.* **10**, 524–531 (2025).
10. K. Sreenath, V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots" in *Robotics: Science and Systems* (RSS Foundation, 2013).
11. G. Li, R. Ge, G. Loianno, Cooperative transportation of cable suspended payloads with mavs using monocular vision and inertial sensing. *IEEE Robot. Autom. Lett.* **6**, 5316–5323 (2021).
12. G. Li, G. Loianno, "Nonlinear model predictive control for cooperative transportation and manipulation of cable suspended payloads with multiple quadrotors" in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2023), pp. 5034–5041.
13. T. Lee, K. Sreenath, V. Kumar, "Geometric control of cooperating multiple quadrotor UAVs with a suspended payload" in *IEEE Conference on Decision and Control* (IEEE, 2013), pp. 5510–5515.
14. T. Lee, Geometric control of quadrotor UAVs transporting a cable-suspended rigid body. *IEEE Trans. Control Syst. Technol.* **26**, 255–264 (2017).
15. J. Geng, P. Singla, J. W. Langelaan, Load-distribution-based trajectory planning and control for a multilift system. *J. Aerosp. Inf. Syst.* **19**, 366–381 (2022).
16. K. Wahba, W. Hönig, Efficient optimization-based cable force allocation for geometric control of a multirotor team transporting a payload. *IEEE Robot. Autom. Lett.* **9**, 3688–3695 (2024).
17. K. Wahba, J. Ortiz-Haro, M. Toussaint, W. Hönig, Kinodynamic motion planning for a team of multirotors transporting a cable-suspended payload in cluttered environments. arXiv:2310.03394 [cs.RO] (2023).
18. N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots. *Auton. Robots* **30**, 73–86 (2011).
19. M. Manubens, D. Devaurs, L. Ros, J. Corte's, "Motion planning for 6-D manipulation with aerial towed-cable systems" in *Robotics: Science and Systems* (RSS Foundation, 2013).
20. E. Fresk, G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor" in *European Control Conference* (IEEE, 2013), pp. 3864–3869.
21. D. Sanalitro, H. J. Savino, M. Tognon, J. Cortés, A. Franchi, Full-pose manipulation control of a cable-suspended load with multiple UAVs under uncertainties. *IEEE Robot. Autom. Lett.* **5**, 2185–2191 (2020).
22. C. Masone, P. Stegagno, Shared control of an aerial cooperative transportation system with a cable-suspended payload. *J. Intell. Robot. Syst.* **103**, 40 (2021).
23. S. Sun, A. Franchi, "Nonlinear MPC for full-pose manipulation of a cable-suspended load using multiple UAVs" in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2023), pp. 969–975.
24. G. Li, X. Liu, G. Loianno, RotorTM: A flexible simulator for aerial transportation and manip-ulation. *IEEE Trans. Robot.* **40**, 831–850 (2024).
25. M. Bernard, K. Kondak, G. Hommel, Load transportation system based on autonomous small size helicopters. *Aeronaut. J* **114**, 191–198 (2010).
26. M. Bernard, K. Kondak, I. Maza, A. Ollero, Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robot.* **28**, 914–931 (2011).
27. E. J. Smeur, Q. Chu, G. C. De Croon, Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *J. Guid. Control Dynam.* **39**, 450–461 (2016).

Downloaded from https://www.science.org at Delft University on November 11, 2025

28. E. Tal, S. Karaman, Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Trans. Control Syst. Technol.* **29**, 1203–1218 (2020).
29. S. Sun, A. Romero, P. Foehn, E. Kaufmann, D. Scaramuzza, A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Trans. Robot.* **38**, 3357–3373 (2022).
30. D. Mellinger, V. Kumar, "Minimum snap trajectory generation and control for quadrotors" in *IEEE International Conference on Robotics and Automation* (IEEE, 2011), pp. 2520–2525.
31. P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, D. Scaramuzza, Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Sci. Robot.* **7**, eabl6259 (2022).
32. S. F. Hoerner, *Fluid-Dynamic Drag: Practical Information on Aerodynamic Drag and Hydrodynamic Resistance* (Hoerner Fluid Dynamics, ed. 2, 1965), pp. 3–17.
33. A. Petitti, D. Sanalitro, M. Tognon, A. Milella, J. Cortes, A. Franchi, "Inertial estimation and energy-efficient control of a cable-suspended load with a team of UAVs" in *IEEE International Conference on Unmanned Aircraft Systems* (IEEE, 2020), pp. 158–165.
34. A. Richards, J. How, "Robust stable model predictive control with constraint tightening" in *2006 American Control Conference* (IEEE, 2006); 10.1109/ACC.2006.1656440.
35. J. Köhler, M. A. Müller, F. Allgöwer, "A novel constraint tightening approach for nonlinear robust model predictive control" in *IEEE American Control Conference* (IEEE, 2018), pp. 728–734.
36. P. Foehn, A. Romero, D. Scaramuzza, Time-optimal planning for quadrotor waypoint flight. *Sci. Robot.* **6**, eabh1221 (2021).
37. H. G. Bock, K.-J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proc. Vol.* **17**, 1603–1608 (1984).
38. M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, F. Allgöwer, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Proc. Control* **12**, 577–585 (2002).
39. R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, M. Diehl, Acados—A modular open-source framework for fast embedded optimal control. *Math. Program. Comput.* **14**, 147–183 (2022).
40. M. Faessler, A. Franchi, D. Scaramuzza, Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robot. Autom. Lett.* **3**, 620–626 (2017).
41. S. Umeyama, Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 376–380 (1991).
42. D. Brescianini, R. D'Andrea, Tilt-prioritized quadrocopter attitude control. *IEEE Trans. Control Syst. Technol.* **28**, 376–387 (2018).
43. D. Zou, P. Tan, W. Yu, Collaborative visual SLAM for multiple agents: A brief survey. *Virtual Real. Intell. Hardw.* **1**, 461–482 (2019).
44. Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, L. Carlone, Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Trans. Robot.* **38**, 2022–2038 (2022).
45. T. Cieslewski, "Decentralized multi-agent visual SLAM," thesis, University of Zurich, Zurich, Switzerland (2021).
46. M. Peterson, P. Lusk, A. Avila, J. How, TCAFF: Temporal consistency for robot frame alignment. arXiv:2405.05210 [cs.RO] (2025).

Acknowledgments: We would like to thank K. Khomenko, M. Pfaff, G. Corsini, R. Belletti, D. Benders, and S. Agarwal for their support during the experiments. We also extend our gratitude to Y. Song, G. Li, G. de Croon, and D. Scaramuzza for their valuable discussions and feedback. **Funding:** This work was supported by the Dutch Research Council (NWO) through the Veni Talent Programme (grant no. 20256, Accurate Aerial Manipulation), by the French National Research Agency (ANR) under grant ANR-24-CE33–5799 (MATES), and by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 871479 (AERIAL-CORE).
Author contributions: S.S. acquired funding, formulated the main ideas, developed the algorithm, conducted experiments, and wrote the manuscript. X.W. developed the algorithm, conducted experiments, and revised the manuscript. D.S. helped with the algorithm implementation and revised the manuscript. A.F., M.T., and J.A.-M. formulated ideas, acquired funding, and revised the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. The data for this study have been deposited in the Dryad database (https://doi.org/10.5061/dryad.n2z34tn6w).

Submitted 4 December 2024
Accepted 1 October 2025
Published 29 October 2025
10.1126/scirobotics.adu8015

Downloaded from https://www.science.org at Delft University on November 11, 2025