AUTONOMOUS VEHICLES

Unlocking aerobatic potential of quadcopters: Autonomous freestyle flight generation and execution

Mingyang Wang[1,2], Qianhao Wang[1,2], Ze Wang[2,3], Yuman Gao[1,2], Jingping Wang[1,2], Can Cui[1,2], Yuan Li[1,2], Ziming Ding[1,2], Kaiwei Wang[3], Chao Xu[1,2], Fei Gao[1,2,4]*

Quadcopter drones are capable of executing complex aerobatic maneuvers when controlled manually by skilled pilots but are limited to simple aerobatic actions when flying autonomously in open spaces. As such, this study introduces a comprehensive system that enables drones to generate and execute sophisticated aerobatic maneuvers in complex environments with dense obstacle distributions. A universal representation is proposed, succinctly capturing flight as a series of discrete aerobatic intentions. These intentions consist of topology and attitude changes, which can be combined in various ways to describe intricate flight maneuvers. A spatial-temporal joint optimization trajectory planner is also introduced to generate dynamically feasible trajectories that are as smooth as possible and devoid of collisions. In addition, we investigate unique yaw sensitivity issues in aerobatic flight and identify the inherent influence of differential flatness singularities on yaw rotations while avoiding associated dynamics issues. A series of ablation studies confirmed the necessity of these spatial-temporal joint optimization and yaw compensation strategies. Additional simulations and physical experiments validated the stability and feasibility of our proposed system for improving uncrewed aerial flight. The proposed system enables drones to autonomously achieve flight performance usually reserved for professional pilots, unlocking boundless potential for aerobatic flight evolution in uncrewed aerial vehicles.

INTRODUCTION

Aerobatic flight involves complex maneuvers that are not typically performed during normal vehicle operation, given that they are inherently dangerous because of the unstable attitudes required. However, these aerial feats are critical for the survival of several flying animals and are often performed instinctively. For example, sparrowhawks (*1*) and falcons (*2*) can rapidly alter speed and direction through vertical or inverted flight to pursue prey or evade obstacles. Bats (*3*) are adept at executing midair flips and hanging upside down from cave ceilings, whereas ravens (*4*) showcase impressive aerobatics to attract their peers. For these species, aerobatic maneuvers offer increased adaptability and agility in complex environments, serving as critical survival skills that are necessary for predation, roosting, and mating. This raises the following question: Could aerial vehicles benefit from aerobatic skills to achieve maneuverability similar to these species, enhancing their flying capabilities and adaptability in complex environments?

Trained human pilots have already demonstrated their ability to perform aerobatic flight, and one can draw insights from their flight performance. Specifically, quadcopters are capable of first-person-view (FPV) freestyle flight and can execute aerobatic maneuvers similar to those observed in nature. FPV human pilots can remotely manipulate quadrotors to perform frequent large-attitude tumbles, featuring high-speed obstacle shuttling, from a first-person perspective using head-mounted display devices (*5*). Demonstrations of human-controlled FPV freestyle flights illustrate the dynamic agility of quadrotors in navigating complex environments, highlighting the immense potential of aerobatic flight.

Aerobatic flight offers opportunities to enhance the applicability and performance of aerial vehicles in practical tasks through advanced maneuverability. For instance, these vehicles can acquire sensor data from critical angles; during missions where sensor perception is limited, aerial aerobatics allow access to positions and orientations that are otherwise unattainable with conventional flight, thereby facilitating the collection of sensor data from optimal viewpoints. Moreover, they can launch supplies upward by performing aerobatic maneuvers to deliver items to otherwise inaccessible locations, such as propelling sensor probes into volcanic craters. In addition, aerial vehicles can engage in search and rescue operations within narrow spaces. By using aerobatic maneuvers, they can swiftly navigate through tight gaps that horizontal flight cannot access, aiding in the timely location of injured individuals. Last, these vehicles assist crewed spacecraft in safely navigating through space debris or a meteoroid belt. In this scenario, the spacecraft must maneuver through hazardous obstacles at various angles while maintaining gentle movements to minimize overload on both the astronauts and the vessel.

The difficulties associated with these tasks can be summarized by two principal components. First, there is a necessity for diverse maneuver types. These missions require aerial vehicles to perform various aerobatic maneuvers tailored to specific scenarios, necessitating user intentions to impose constraints on flight actions. For instance, the vehicle may need to execute a 90° roll while flying through a designated window into a building. As the number of desired intentions increases, defining the problem within a generalized framework becomes more difficult, making the search for solutions increasingly challenging. Second, there is a need for the aircraft to execute actions safely. Commercial aerial vehicles currently struggle with performing agile aerial maneuvers, and the complexities of obstacle avoidance in challenging environments further complicate successful execution. Although previous studies have addressed the limitations of drone control and perception, there has been relatively little focus on the rationale for aerobatic trajectories themselves.

[1]Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou, China. [2]Huzhou Institute of Zhejiang University, Huzhou, China. [3]College of Optical Science and Engineering, Zhejiang University, Hangzhou, China. [4]Differential Robotics Technology Co., Ltd., Hangzhou, China.
*Corresponding author. Email: fgaoaa@zju.edu.cn

These challenges can be illustrated by FPV freestyle flight, in which pilots perform aerobatic maneuvers that produce visually notable effects on the basis of a flight environment. During freestyle sessions, drone pilots envision a future quadcopter flight path and the associated changes in orientation. They then use their skills to translate this vision into breathtaking aerobatics. However, these aerobatic maneuvers unfortunately all rely on the remote control of master-level pilots. Despite substantial advances in artificial aircraft within both industrial (*6*) and academic (*7*, *8*) settings, achieving full autonomy for executing aerobatic maneuvers in complex environments, similar to those traversed by birds or elite pilots, remains an unresolved challenge. As such, we aimed to unlock the boundless aerobatic potential of uncrewed drones (see movie S1).

The potential for uncrewed aerobatic flight has led to extensive research in recent years. Song *et al.* (*9*) and Kaufmann *et al.* (*7*) described the use of reinforcement learning and model predictive control in high-speed drone racing, outperforming champion pilots. During these flights, quadrotors attempt to make drastic attitude changes to pass through certain gates more quickly, possibly even pitching downward. However, this type of racing maneuver only considers the outcome and lacks the capability to respond to required arbitrary attitude changes, which are the primary difficulty of aerobatic flight. In the pursuit of aerobatic maneuvers, Chen *et al.* (*10*) and Lupashin *et al.* (*11*) achieved multiple flips using quadrotors by segmenting the maneuver into different phases. Although this segmented control technique considers attitude changes in aerobatic trajectories, it neglects the three-dimensional (3D) shape of aerobatic maneuvers. In contrast, vertical circles and arcs have been used to fit specific aerobatic maneuvers known as power loops (see Fig. 1A): 360° pitch rotations executed in a shape similar to a circle (*12*, *13*). However, not all aerobatic maneuvers (*14*) can be accurately approximated using predefined arc-shaped geometries, particularly in complex environments. The techniques discussed above focus solely on achieving a specific type of aerobatic maneuver and are unable to generate more complex maneuvers (*10*–*13*). In comparison, a more rational approach is to specify key waypoints and their corresponding velocities or attitudes within a given trajectory, enabling the execution of more intricate aerobatic maneuvers. Building on this concept, Jackson *et al.* (*15*) attempted to use quadcopters to perform 360° flips in a simulated environment, whereas Tal *et al.* (*16*) and Lu *et al.* (*17*) demonstrated the execution of power loop and split-S maneuvers for vertical takeoff and landing (VTOL) aircraft, incorporating a combination of maneuvers. However, the aerobatic trajectories generated in these studies require repeated fine-tuning of key trajectory parameters. As such, it is nearly impossible to generate dynamically feasible flight trajectories without extensive tuning

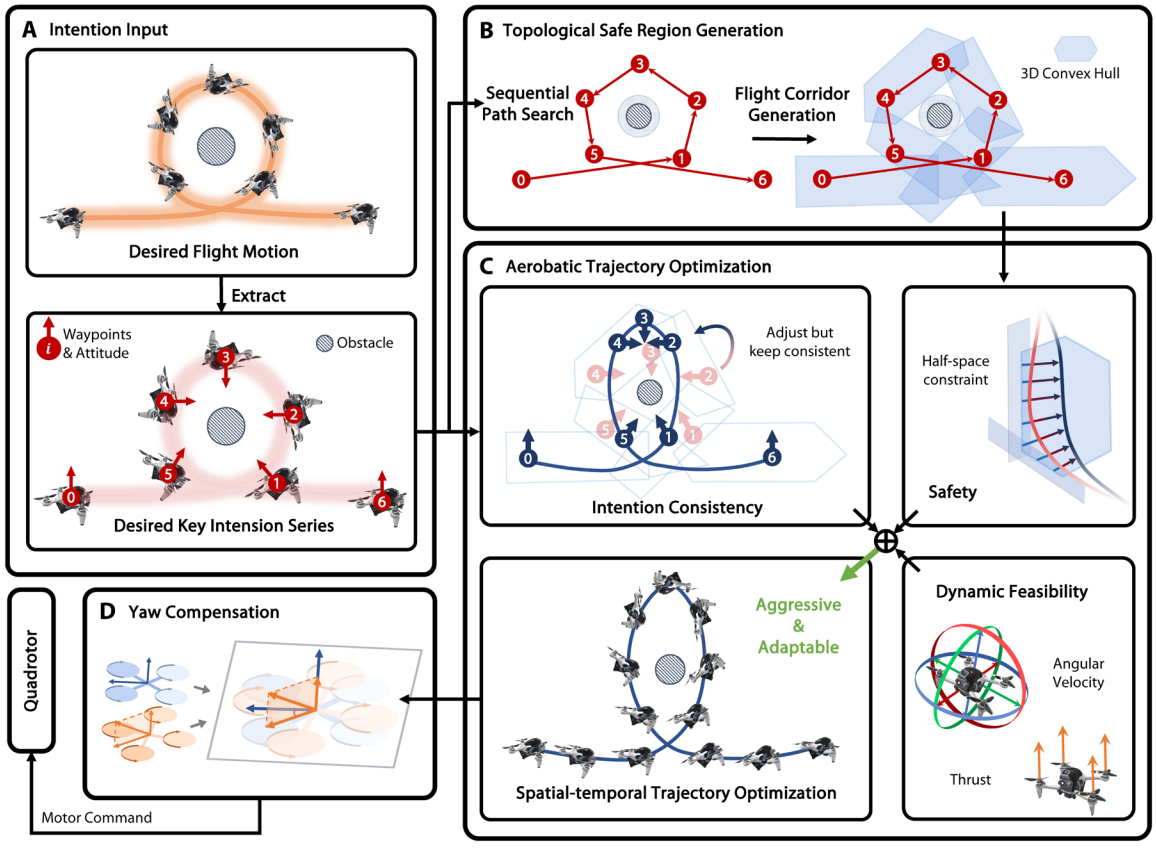


Fig. 1. An illustration of the proposed strategy for aerobatic maneuver representation, generation, and execution. (**A**) Users are asked to envision their desired aerobatic maneuver as a discrete sequence of spatial and attitudinal representations. (**B**) The flight corridor guarantees both trajectory topology and defined collision-free spaces. (**C**) Planner adjusts the position of the attitude to ensure smoother and more rational transitions along the trajectory while satisfying attitudinal intentions. (**D**) Yaw compensation prevents numerical instability issues resulting from differential flatness singularities, ensuring consistency between planned trajectories and control computations. Additional algorithm details can be found in Materials and Methods and Supplementary Materials.

experience. In short, existing research does not adequately capture the primary aim of aerobatic flight, which requires a consideration of human intention to generate various types of maneuvers. Furthermore, these techniques struggle with object avoidance and are thus limited to flying in open, obstacle-free environments. Thus, in more complex scenarios, these conventional strategies cannot ensure the generation of dynamically feasible trajectories.

The autonomous and safe execution of aerobatic maneuvers in complex environments has relied on previous research into highly dynamic quadrotor flight, with substantial progress made in recent years. Now, vision-based (*18*, *19*) and LIDAR (light detection and ranging)–based (*20*, *21*) inertial odometry techniques have achieved high-precision localization, improving on the prior constraints of external localization systems (*22*, *23*). Modern drones are now capable of executing smooth flight, even at high speeds, by using dependable perception (*24*), control (*25*, *26*), and trajectory planning (*27*) modules (*28*, *29*). In addition, spatial and end-state constraints allow drones to maneuver through tilted narrow gaps (*27*, *30*) or land on rapidly moving inclined platforms (*31*–*33*), achieving flight attitudes above 90°. Compared with the manual specification of key parameters for a given trajectory (*15*–*17*), this optimization approach, which involves setting both obstacle avoidance and attitude constraints, enables agile flying while ensuring dynamic feasibility. The resulting trajectory is more viable as a result, and the method itself exhibits excellent scalability. However, extending trajectory optimization methods intended for agile flying directly to aerobatic maneuvers yields two unresolved issues. First, the attitude constraints for agile flight (*27*, *30*–*33*) remain dependent on the position, which does not fundamentally differ from the traditional approach used to specify waypoints and their attitudes. Although trajectory optimization can ensure dynamic feasibility, it cannot arbitrarily combine the required aerobatic maneuvers. Second, the methods discussed above take advantage of differential flatness mapping for multirotor crafts (*26*, *34*) during both planning and control. This mapping has been proven to exhibit singularities in $SO(3)$ space, the group of all rotations in 3D space (*35*), which often occur when the drone is inverted or exhibits pitch or roll angles of 90°. Although normal drone flight (attitude of at most 60°) typically avoids singularities, aerobatic maneuvers that span the entire $SO(3)$ space are bound to pass near them, which poses new challenges for generating dynamically feasible aerobatic trajectories.

The aerobatic potential of quadrotors was investigated in this study by introducing a comprehensive system capable of generating any desired aerobatic maneuver in the presence of obstacles while circumventing the blocks stemming from differential flatness singularities. First, we proposed a universal representation of aerobatic trajectories that supports any combination of aerobatic motions, as shown in Fig. 1A. Specifically, we delineated two essential characteristics of these trajectories: spatial topology and attitude variations. The former can be simplified as a sequence of crucial topological waypoints, whereas the latter can be generalized as a sequence of vital attitudes, collectively referred to in this paper as aerobatic intentions. Users can succinctly express their aerobatic intentions on the basis of envisioned maneuvers, bypassing a complex and difficult tuning process. Second, we meticulously designed a series of differentiable metrics that consider aerobatic intention, safety, and dynamic feasibility (Fig. 1C). We leveraged these metrics and transformed the basic requirements of an aerobatic trajectory into a numerical optimization process, which can be efficiently solved to generate the desired aerobatic maneuvers, even if the initially designated waypoints are infeasible. Third, we extensively analyzed sensitivity issues near the singularity point, caused by differential flatness mapping (see Fig. 1D). To bypass this issue, we developed a mapping strategy called yaw compensation mapping (YCM). This approach avoids possible numerical instabilities and ensures that the aerobatic trajectory satisfies dynamic feasibility requirements.

Our system was deployed in two different classical freestyle FPV drone models with different wheelbases and distinct flight capabilities (Fig. 2B). A ground station provided a visual representation of environments and generated aerobatic trajectories, where the user provided intended aerobatic motions (see Fig. 2C). Our system also used known maps as input (Fig. 2A) to provide users with an intuitive demonstration of flight trajectory generation. This is a natural requirement for FPV freestyle flight because even the most skilled pilots require familiarity with the flight environment before executing aerobatic maneuvers. The drone, equipped with embedded computing devices, hosts an entirely autonomous navigation algorithm responsible for trajectory planning, localization, and control (Fig. 2D). Extensive simulations and physical experiments confirmed that our system achieved fully autonomous aerobatic flight previously unattainable by other models, allowing drones to outperform human pilots.

RESULTS

The viability of the proposed system was assessed using both simulations and two distinct flight challenges conducted in real-world environments. The first experiment was intended to validate the practical applicability of the proposed system in real freestyle scenarios, whereas the second aimed to verify the stability during extreme maneuvers in confined spaces. In addition to these two experiments, we also conducted flight comparisons with one of the top drone pilots in China. The results demonstrated that our proposed system performed aerobatic maneuvers comparable to those of professional aviators while consistently attaining higher success rates across test scenarios. Corresponding simulations illustrated the process of generating trajectories on the basis of specified intentions and quantitatively verified the necessity of key system modules through ablation analysis.

Large-scale aerobatic flight

The proposed system ensures safe autonomous aerobatic flight for drones in vast, unstructured environments. As depicted in Fig. 3, A and B, this experiment involved an unstructured setting that included both artificial objects and natural foliage, spanning a large volume measuring 100 m by 40 m by 20 m. This classic scene is ideal for human pilots to showcase their aerobatic skills, with bridges and abundant trees providing rich visual references that require visually complex maneuvers during flight. By premapping the environment and inputting the aerobatic intentions, we generated a 220-m-long aerobatic trajectory that seamlessly combined three classic maneuvers. Figure 3C illustrates the flight trajectory and associated attitude change for each maneuver. The included yaw compensation strategy allowed the drone to maintain its heading in the direction of flight throughout the process, preventing any superfluous yaw rotations when the attitude was fully reversed. The smooth and continuous angular velocity curve in Fig. 3D further verifies this effect. A detailed analysis of the YCM and the differential flatness singularity

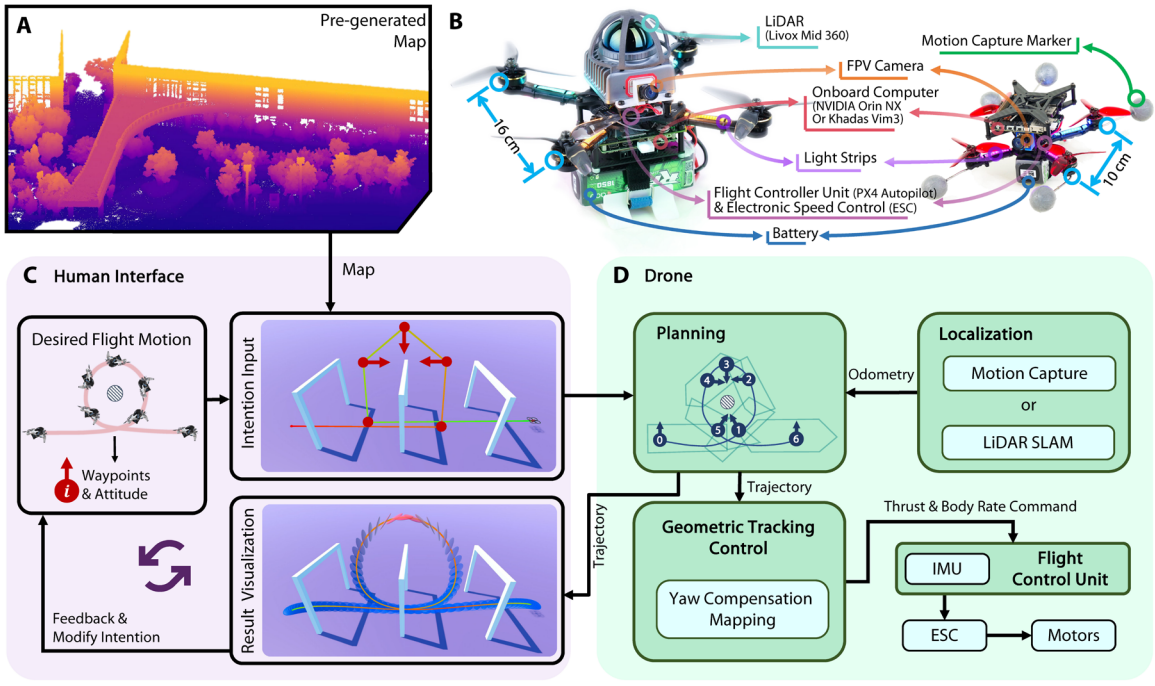


Fig. 2. Hardware and system architecture specifications. (**A**) A prescanned map includes rich environmental data and provides topological references for user intention inputs and trajectory planning. (**B**) Flight platform hardware components. Colored light strips were installed on the arms of the drone to distinguish its orientation during experiments. (**C**) A visual ground station with an interactive interface was provided to users, allowing them to preview aerobatic trajectories and adjust their input intentions, providing better alignment with their expectations. (**D**) Localization, planning, and control modules running on the onboard drone computer.

can be found in the "Yaw compensation mapping" section and Supplementary Methods.

Furthermore, to avoid excessive tracking errors that could lead to crashes, the proposed system aims to generate aerobatic trajectories that are as smooth and gradual as possible, eliminating overloads caused by drastic changes in thrust or angular velocity. In this experiment, the upper limit of the net thrust during trajectory planning was set to only 1.5 times the acceleration due to gravity, whereas the maximum angular velocity was restricted to 4 rad/s. These physical parameters are easily achievable for almost any quadcopter. Benefiting from strict dynamic constraints imposed on the aerobatic trajectory, both the thrust and angular velocity were rigorously confined within their respective intervals throughout the entire planned trajectory. Figure 3D depicts key flight data, including position and attitude tracking errors, net thrust, and three-axis angular velocity. We used additional color bands to highlight the time intervals during which these aerobatic maneuvers were executed. During the time intervals of 2 to 4 s, 8 to 10 s, and 16 to 18 s (while maneuvers were being performed), positional errors did not exceed 15 cm, and the attitude tracking error remained within 12°. These tracking errors did not exhibit a noticeable increase compared to the transitional phases between aerobatic maneuvers (i.e., time intervals of 4 to 8 s and 10 to 16 s). Trajectory optimization strictly constrained the net thrust and angular velocity to remain within the generous upper limits of the dynamic parameters, ensuring that the drone could easily execute aerobatic maneuvers. For more details on trajectory optimization, refer to the "Trajectory representation" and "Problem formulation" sections. Video footage of this experiment is provided in movie S2.

Successive aerobatic maneuvers in confined spaces

The experiments discussed previously highlighted the impressive performance achieved by the proposed system in large-scale aerobatic flight scenarios. In comparison, the experiment shown in Fig. 4 was intended to evaluate system capabilities for generating aerobatic trajectories in confined spaces, which require drones to perform maneuvers near obstacles. As illustrated in Fig. 4C, the experiment restricted aerobatic flight to within the narrow range of a motion capture system (*36*) at a height of only 3.5 m. Tunnels, circular gates, and flags were strategically positioned in the environment and served as obstacles. Aerobatic intentions were incorporated into the system, which generated agile aerobatic trajectories designed to avoid obstacles. This included shuttling through circular gates in reverse and frequent passage through tunnels (see Fig. 4B and movie S3).

Notably, users were only required to specify the general position and attitude of the aerobatic intentions, without the need to meticulously adjust specific position details. This is because the planner assists users in autonomously adjusting the timing and positioning of aerobatic intentions along the entire trajectory. Figure 4B depicts a scenario in which the original intentions input by the user are spaced too closely. As such, the resulting trajectories would likely be infeasible if strictly adhering to the rules established by existing methods (*15*–*17*). In contrast, the proposed system gradually adjusts the aerobatic intentions toward more rational positions to ensure that the generated actions are dynamically feasible. The proposed model can also ensure stable execution of highly dynamic trajectories because of the strict constraints imposed by dynamic variables. In this experiment, the flight speed at most positions along the trajectory exceeded 4 m/s, yet position tracking errors generally remained below

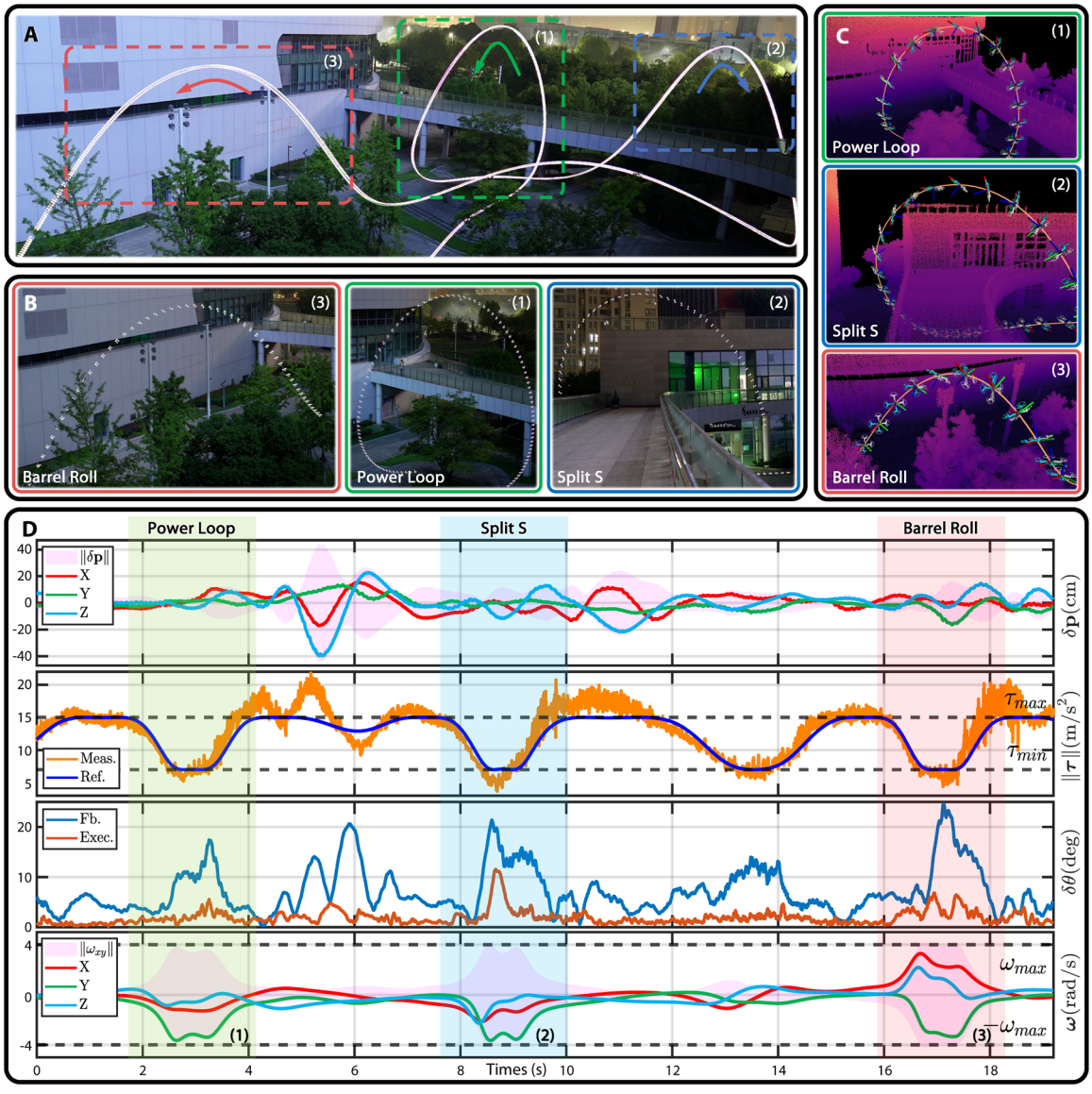


Fig. 3. Large-scale aerobatic flight in an unconstrained environment. (**A**) Sequential snapshots of the flight trajectory, including a barrel roll (*3*), power loop (*1*), and split-S (*2*) from left to right. (**B**) Detailed close-up snapshots of three classic aerobatic maneuvers when the drone was upside down at its highest point. (**C**) Environment point cloud. The quadrotor attitude and trajectory were recorded and replayed in the visualization software, which showed changes in drone attitude during each aerobatic maneuver. The red, green, and blue drone axes correspond to the *x*, *y*, and *z* axes in the body frame. (**D**) Aerobatic flight parameters. The $\delta\mathbf{p}$ term represents the positional tracking error; *X*, *Y*, and *Z* denote the single-axis error; and $\|\delta\mathbf{p}\|$ indicates the total tracking error (the two norm of positional errors along individual axes). The $\boldsymbol{\tau}$ term denotes the net thrust, Ref. signifies expected values obtained from the planned trajectory, and Meas. represents the two norm of the 3D linear acceleration measured by the inertial measurement unit (IMU) of the drone. The $\delta\theta$ term represents the attitude tracking error, which refers to the angular value of the difference between two quaternions converted into axis-angle representations. In our experiments, a cascaded proportional-derivative controller was used to regulate the drone by adjusting the desired attitude on the basis of positional and velocity tracking errors. The attitude corrected by the controller reflects the actual attitude adopted by the drone during execution. Fb. denotes the error between the controller-corrected attitude and the theoretically planned trajectory attitude, and Exec. represents the deviation between the actual drone attitude and the controller-corrected attitude. The $\boldsymbol{\omega}$ term denotes the angular velocity, where $\|\omega_{xy}\| = \sqrt{\omega_x^2 + \omega_y^2}$. Each of these symbols can be found in Materials and Methods and Supplementary Materials.

0.15 m (see Fig. 4A). Furthermore, this aerobatic trajectory was repeated eight times in complex environments. The position and attitude tracking errors were then calculated at aerobatic intention points during each flight (see Fig. 4C). The flight data confirmed that our generated aerobatic trajectories could be reliably and safely executed.

Combinations of aerobatic intentions

In the previous section, we briefly discussed aerobatic intentions and their corresponding generated maneuvers. In this section, we further describe various types of aerobatic intentions and the various maneuvers they can flexibly combine to produce flight trajectories. As

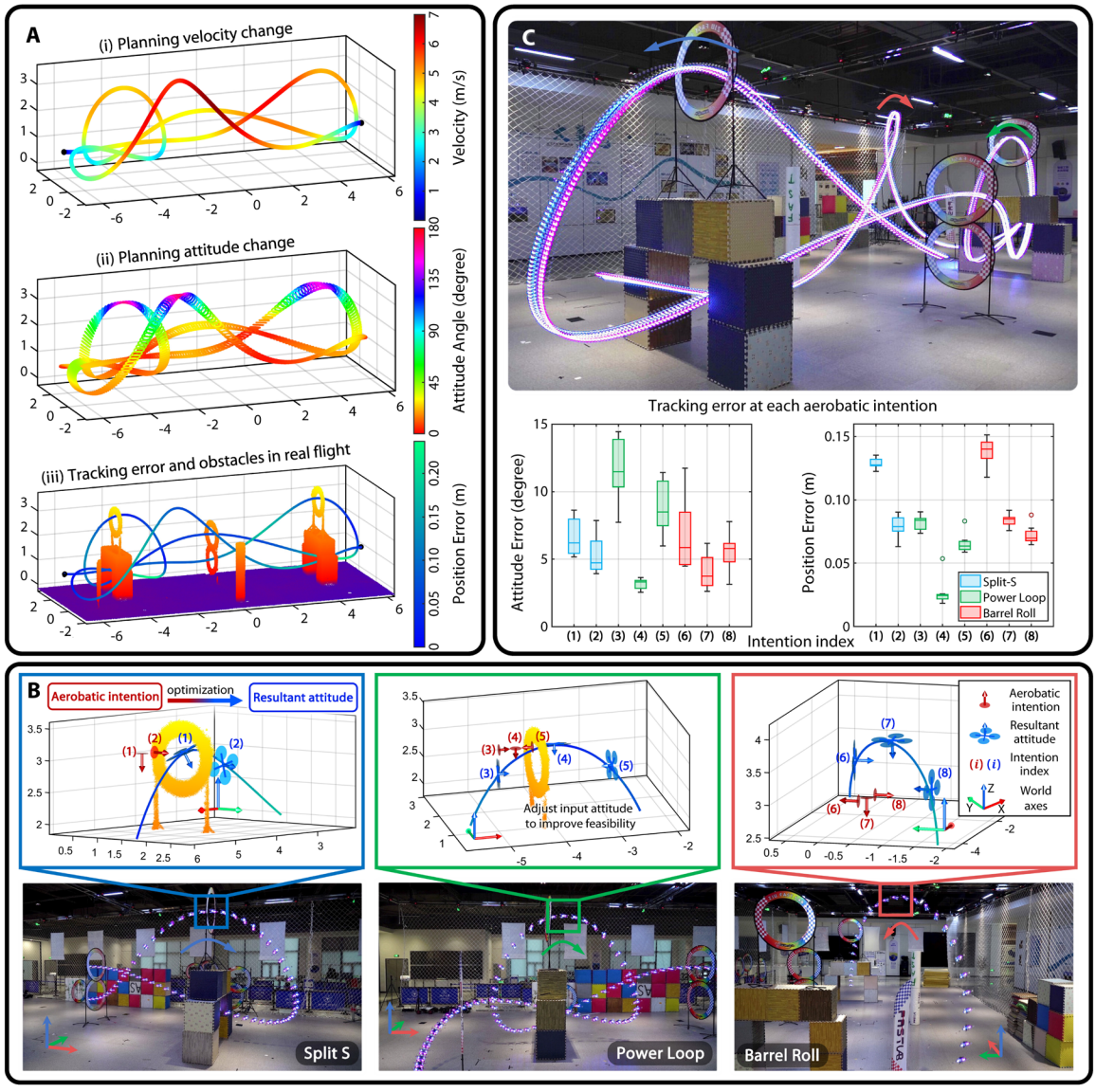


Fig. 4. Performing aerobatics while avoiding obstacles. (**A**) Overview of the aerobatic flight trajectory. (i) The planned trajectory and corresponding speed. (ii) The planned trajectory attitude. The attitude angle was defined as the angle between the direction of the drone's *z* axis and the direction opposite the acceleration of gravity. The attitude of the drone at each sampling point was represented by a circle perpendicular to the *z* axis. (iii) Environmental obstacles and tracking errors for trajectory execution. (**B**) Specific input intentions for each aerobatic maneuver, corresponding position and attitude for the planned trajectory, and snapshots of the actual flight trajectory. The point cloud and trajectory are of the same significance as [(A), iii]. (**C**) Real aerobatic snapshots and statistics. The quadrotor was equipped with light bars on all four arms, with the nose bar colored purple and the tail colored blue. The horizontal coordinates in the box plot represent the indices of the eight aerobatic intentions entered by the user. All data were derived from the eight complete aerobatic flights.

mentioned in the description of Fig. 1A, the key components of aerobatic actions include spatial topology and attitude variations, each of which can be represented by discrete positions and attitudes. Specifically, three types of user intentions were defined, as shown in Fig. 5. The first type of intention included only position information, wherein positions along the whole intention sequence were sequentially connected to represent the associated topology. The second and third intentions also included attitude directions, adding attitude constraints to the trajectory and aiming for a position that reached the corresponding attitude. The difference between these two cases is that the second type of intention does not constrain the attitude position, whereas the third intention requires the trajectory to reach a specific attitude while remaining near the current position.

Although these three types of intentions are simple, their combinations can represent various aerobatic maneuvers (see movie S4). Here, we present the intentions corresponding to some classic aerobatic maneuvers. In Fig. 5B, each maneuver is represented by five intentions, excluding the starting and ending points, with three of these intentions providing attitude information. To emphasize the importance of both spatial topology and attitude, we ensured that the position information of the five intentions was completely identical. By rearranging the order of the five intentions and altering attitude

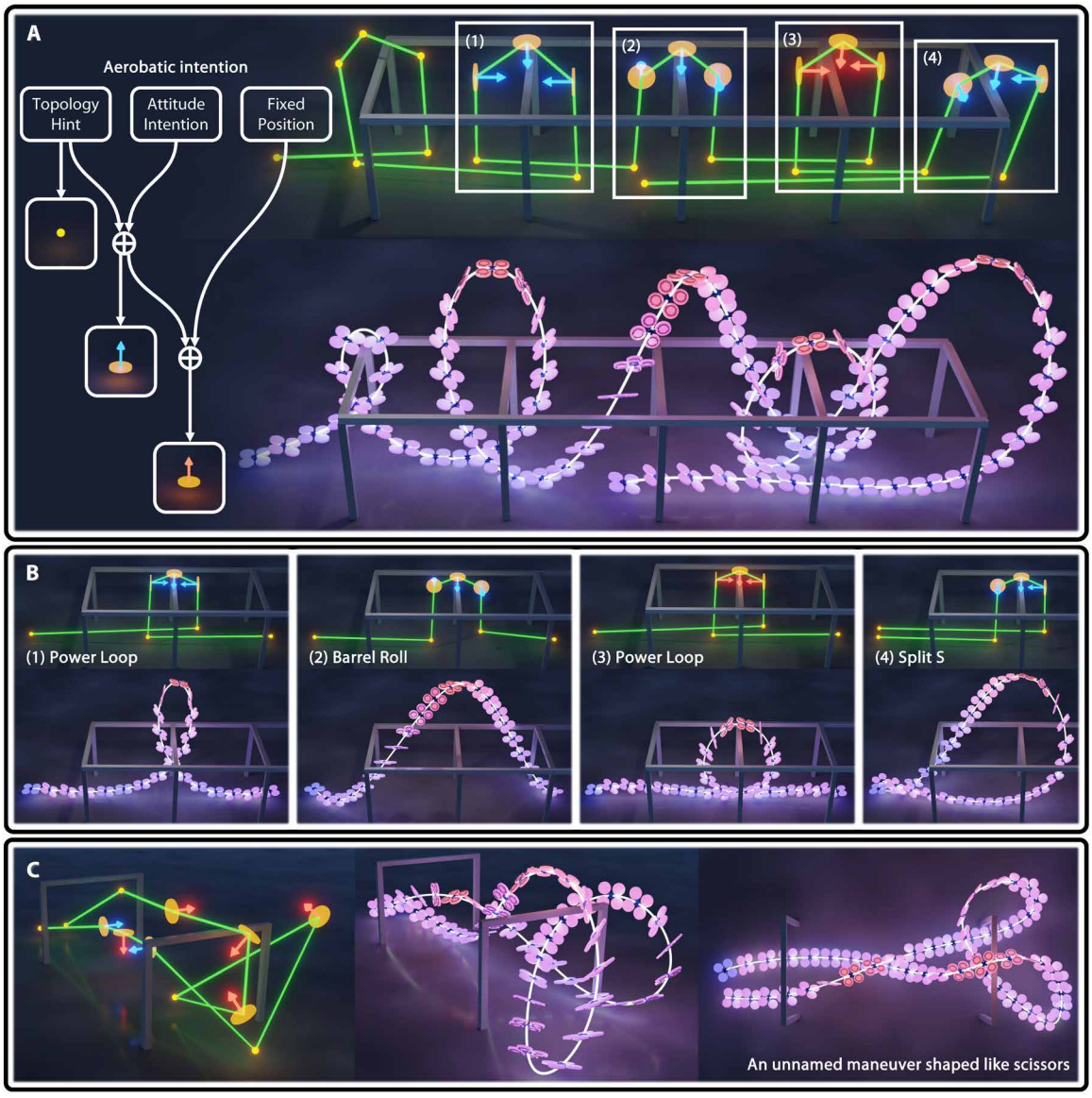


Fig. 5. Various aerobatic intentions and their corresponding trajectories. (**A**) A sophisticated combination of trajectories and associated intentions, divided into three categories and connected in sequential order by green lines. Four consecutive aerobatic maneuvers were performed along this trajectory, each corresponding to the classic maneuvers shown in (B). (**B**) Four classic aerobatic maneuvers and their respective intentions. Each maneuver involved slightly different topological sequences, attitude directions, or intention types, resulting in completely different trajectories. (**C**) Aerobatic maneuver generated by a unique combination of intentions that does not conform to any established classic aerobatic maneuvers.

directions, we effectively generated aerobatic trajectories exhibiting various topologies and attitude variations. Furthermore, the first and third intentions shown in Fig. 5B feature identical positions and attitude directions. However, because the third intention added position constraints, the shape of the trajectory was flattened to achieve an inverted attitude at the position specified by the aerobatic intention.

Discrete intentions not only allowed for a distinct representation of different aerobatic maneuvers but also offered diverse possibilities for combining them. Figure 5A illustrates a complex combination of such trajectories, wherein each submaneuver is shown in Fig. 5B. By connecting discrete aerobatic intentions, we could easily assemble aerobatic maneuvers from single components, creating more intricate and sophisticated results. We could also use a chance combination of the three types of intentions, as shown in Fig. 5C, to create unexpected aerobatic maneuvers. For additional examples of how different combinations of aerobatic intentions lead to various maneuvers, please refer to Supplementary Results.

Matching and surpassing human pilots

The proposed system was further evaluated by comparison with a professional pilot with 5 years of specialized flying experience. S. Zhao has earned second place and several top-eight finishes in Chinese drone racing competitions. Although primarily a racer, he is also skilled in freestyle flight, especially the power loop. In this comparison experiment, he was allowed to operate both his own drones and drones that were identical to our autonomous platform.

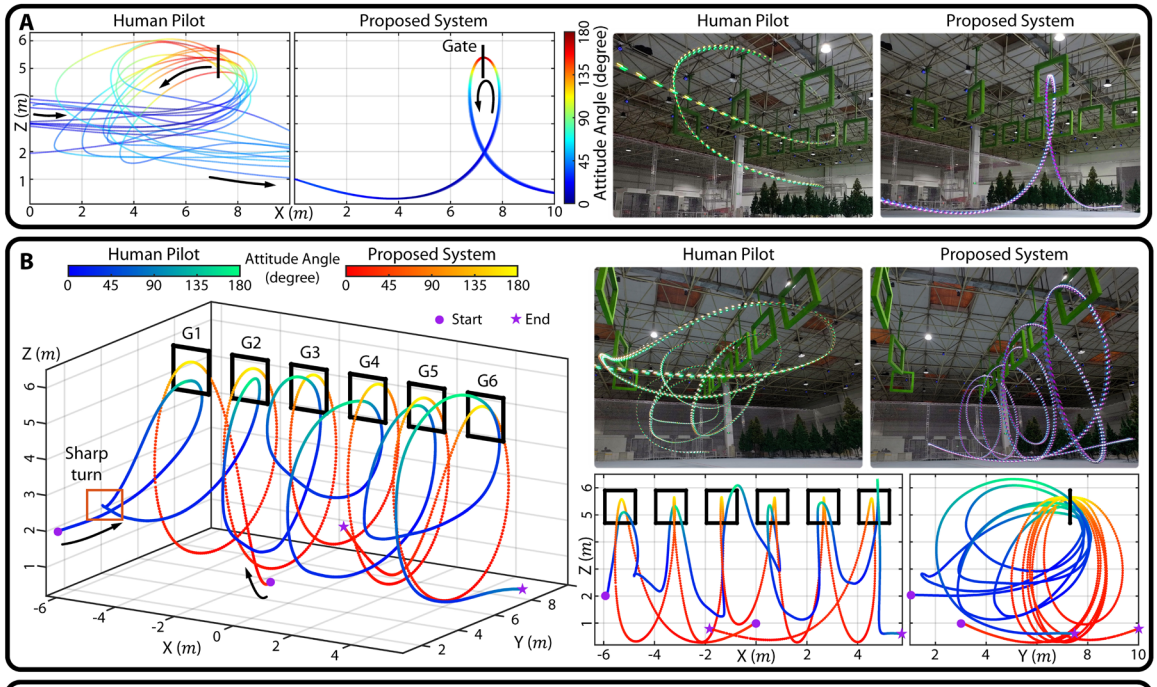
The pilot was given ample opportunity to practice before the competition, allowing him to familiarize himself with the environment and fine-tune the drone to his preferred handling settings. Subsequently, both the human pilot and the proposed system performed identical aerobatic maneuvers, specifically flying backward through a series of 1.2 m–by–1.2 m gates. During flight, we focused solely on the power loop (the pilot's preferred event), without introducing any obstacles other than the gates.

Figure 6A illustrates trajectory shapes and attitude changes during 10 successful executions of the power loop. The colors and shapes indicate that the human pilot maintained an inverted attitude for a longer duration, requiring greater horizontal and vertical distances to recover the drone to a safe attitude. In contrast, the trajectories generated by the proposed system restricted attitude inversion maneuvers to a more compact area. This suggests that when obstacles were present in front of the gate, the human pilot was more susceptible to collisions and crashes because of the difficulty of correcting the aircraft's orientation.

We further devised a challenging maneuver involving six consecutive power loops, in which the drone was required to travel backward through six parallel gates. Figure 6C provides the statistics for these attempts. Although the success rate for a single loop by the human pilot was relatively high (~70%), the overall success rate under continuous flight conditions decreased markedly, with only 3 successes of the 24 attempts. Conversely, the algorithm demonstrated notable stability, achieving a 100% success rate across five flights. Figure 6B provides a comparison between the trajectory of the optimal human pilot flight and that of the algorithm. It is evident that the human pilot encountered difficulties in maintaining continuity throughout the entire flight. The pilot often found it necessary to reduce speed and adjust the drone's attitude and direction to align with the upcoming gate, resulting in trajectories that were characterized by sharp turns, as illustrated in Fig. 6B. Ultimately, Fig. 6 (A and B) indicates that the human pilot had difficulty maintaining a safe distance from the gates, making them more vulnerable to collisions compared with autonomous drones.

Ablation analysis

This section provides a comprehensive overview of core algorithm functionality, encompassing yaw compensation and trajectory optimization. The necessity of these features was validated through a series of ablation experiments.



C

	Human Pilot						Proposed System					
Gate ID	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6
Success Count	13	11	9	5	4	4	5	5	5	5	5	5
Danger Count	3	1	1	0	0	0	0	0	0	0	0	0
Failure Count	3	1	1	2	0	1	0	0	0	0	0	0
Crush Count	5	1	2	4	1	0	0	0	0	0	0	0
Success Rate	67%	86%	77%	45%	80%	80%	100%	100%	100%	100%	100%	100%

Fig. 6. In comparison with human pilot. (**A**) A comparison of 10 successful power loop maneuvers by the human pilot and the proposed system. The attitude angle was defined as the angle between the drone's *z* axis and the direction opposite to gravitational acceleration. (**B**) The best performances from a continuous sequence of six power loops performed by both the human pilot and the proposed system. The color bar represents the attitude angle, and the black arrows indicate flight directions. (**C**) Statistics describing the success rate for continuous power loops (see movie S5). Success indicates safe passage through the gate; danger denotes that a near-crash occurred but the pilot recovered successfully, allowing the flight to continue; failure refers to unsuccessful maneuvers but no crash occurred; and crash signifies an irretrievable fall because of a collision. The success rate was calculated as (successful attempts + dangerous attempts) / (total attempts).

Yaw compensation

Ablation experiments were conducted to visually demonstrate sensitivity issues near the differential flatness singularity and the functionality of YCM, as shown in Fig. 7A. An aerobatic trajectory was generated to represent a classic power loop maneuver, which served as the execution trajectory. We used both classic differential flatness (*26*) and differential flatness with YCM to calculate drone states along the aerobatic trajectory. These two mapping strategies differed only in the calculation of the γ angle. This angle, also known as the yaw angle in classic differential flatness, only affected the drone heading and *z*-axis angular velocity in the final calculations. Additional information is available in the "Yaw compensation mapping" section and the Supplementary Materials.

Figure 7A displays a case without YCM, in which the computed *z*-axis angular velocity fluctuated extensively, reaching almost 200 rad/s when the γ angle was held constant at 0. This caused the heading of the drone to rotate 360° within a brief 80-ms interval, which was both unnecessary and dynamically infeasible. However, in the case with YCM, the γ angle was calculated using the desired velocity direction, thereby preventing these fluctuations. Despite a 360° variation in γ, the computed *z*-axis angular velocity remained stable, and the drone avoided any additional rotations. This suggests that γ lacks any physical importance and is not synonymous with the conventional yaw angle, merely resembling it near-level flight attitudes. In situations where singularities are encountered (e.g., aerobatic flight), incorrect γ values can cause a heading to rotate inexplicably, causing sensitivity issues near the singularities. Therefore, it is imperative to use YCM to prevent additional drone rotations. In this process, the variable γ is adjusted to ensure that physically meaningful *z*-axis angular velocities remain smooth, thereby ensuring dynamic feasibility of the computed trajectory.

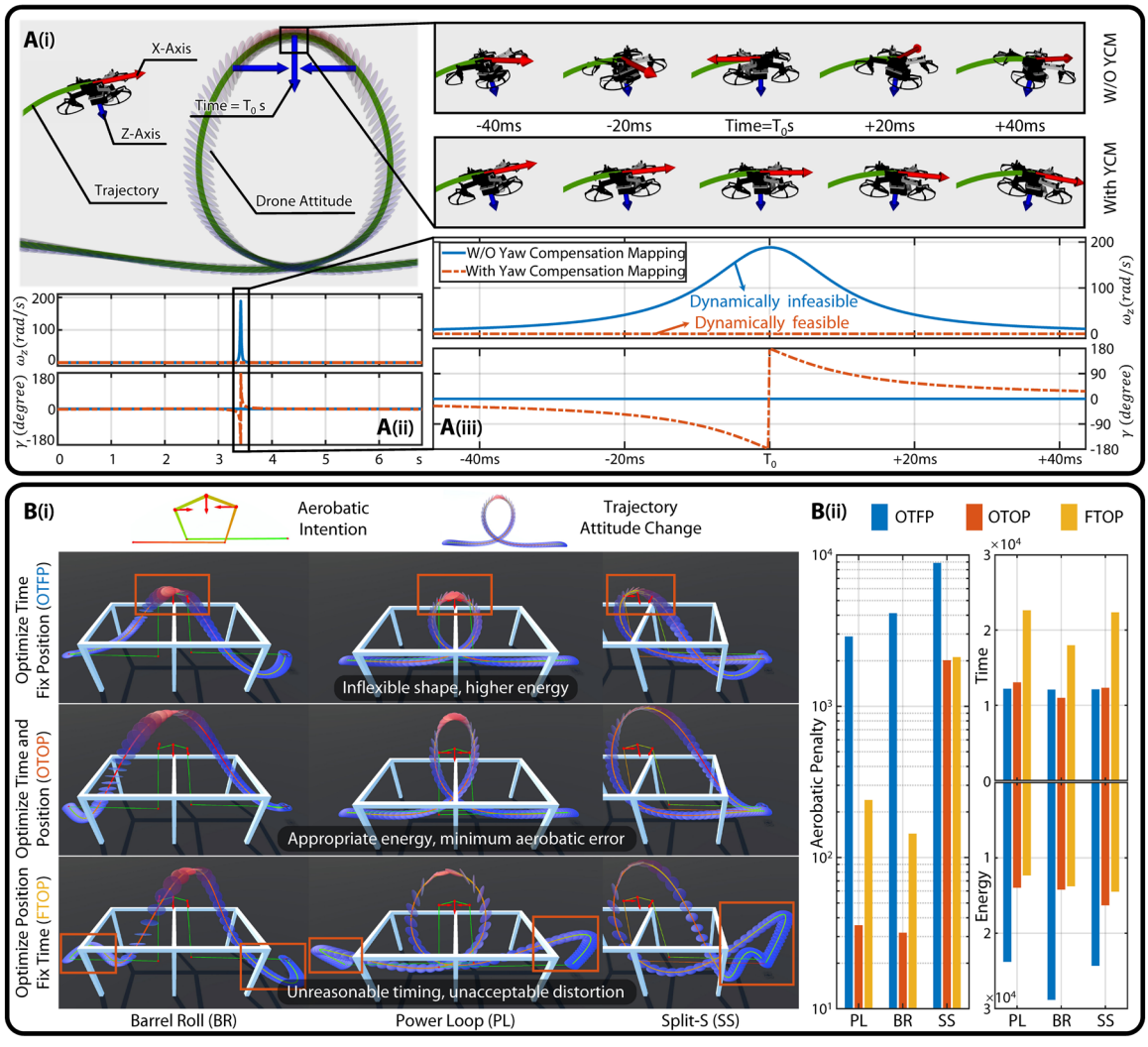


Fig. 7. Ablation analysis. (**A**) A comparison of trajectory state calculations with and without YCM. (i) shows attitude variations for the entire trajectory during aerobatic flight, whereas (ii) depicts drone states computed with and without YCM. (iii) provides a magnified view of drone attitude and calculated state variables within a total time span of 80 ms near T_0, representing the moment in the planned trajectory at which the drone attitude completely reverses. (**B**) Performance comparisons for various optimization conditions. (i) shows typical trajectory diagrams for three classic aerobatic maneuvers, optimized for three different conditions. The bar charts included in (ii) compare several different metrics, with each bar representing a sample size of 10. To be specific, aerobatic penalty measures the extent to which the trajectory aligns with the intended maneuver, with higher values indicating poorer alignment. Energy assesses the overall energy expenditure along the trajectory, and time denotes the allocation of execution time. Formulas for these metrics can be found in Materials and Methods. Plots and graphs can be recreated using the released dataset (*45*).

Trajectory optimization

An additional ablation experiment was conducted to demonstrate the necessity of our proposed spatial-temporal simultaneous optimization method, as shown in Fig. 7B. This approach, termed optimize time and position (OTOP), simultaneously optimizes the temporal and spatial positioning of aerobatic maneuvers along given trajectories, providing a large solution space for trajectory generation. The following techniques were compared in this ablation experiment: Optimize time fix position (OTFP) optimizes only the time while fixing the spatial positions in aerobatic maneuvers, fix time optimize position (FTOP) optimizes only the position while fixing the percentage of aerobatic maneuver time along a trajectory, and OTOP simultaneously optimizes both attributes. By examining these different techniques, we demonstrate the distinct trajectories produced by each condition.

Specifically, we tested three classic aerobatic maneuvers under three different optimization conditions. By simply altering the positions of individual aerobatic intentions, we repetitively generated each trajectory 10 times while ensuring that the resulting actions remained unchanged. As shown in Fig. 7B, OTFP effectively stabilized the trajectory shapes but required more energy for execution. Conversely, although FTOP required the lowest energy consumption, its trajectory shape was distorted by irrational time allocations. In contrast, OTOP maintained a balance between energy savings and time allocation, thereby minimizing the sum of energy and time costs. Furthermore, appropriate time and energy allocations ensured the smooth execution of aerobatic maneuvers. In addition, OTOP ensured minimal aerobatic penalties, regardless of maneuver type. Our proposed system optimized time requirements by default, given that trajectory distortions can render the associated dynamics infeasible. However, we retained the capability to fix the spatial positions of aerobatic intentions, catering to users with specific requirements for the trajectory shape (see Fig. 5A).

DISCUSSION

The primary goal of this study was to achieve the graceful maneuvers performed by birds and professional FPV freestyle pilots, navigating complex environments with natural fluidity. To this end, we proposed a comprehensive system for the generation and execution of aerobatic trajectories. The aerobatic intentions input by users encompassed boundless possibilities for generating diverse maneuvers. A meticulously crafted spatial-temporal trajectory optimization algorithm ensured the safety, stability, and viability of these trajectories. In addition, YCM obviated the safety hazards inherent in executing complex flight trajectories. Simulated ablation studies and real-world experiments substantiated the efficacy of the proposed system for generating more plausible and diverse aerobatic trajectories, which could be safely executed in areas that are densely populated by obstacles. The proposed system enables drones to autonomously achieve flight performance typically reserved for professional pilots, unlocking limitless potential for the evolution of aerobatic flight in uncrewed aerial vehicles.

In the aerobatic demonstrations presented in this study, the drone heading was consistently aligned with the direction of travel (the heading was calculated using a YCM based on the trajectory velocity described in Materials and Methods). However, skilled FPV pilots can achieve visually notable maneuvers (e.g., inverted and sideways flight) by controlling yaw rotations. Specifying a heading that aligns with the direction of travel is merely a means of reducing the dimensionality of variables during trajectory optimization, thereby ensuring that the aerobatic maneuvers remain as concise and clear as possible. Our proposed YCM does not reduce the drone's freedom to perform yaw rotations. Instead, it thoroughly decouples yaw and position within a classic differential flatness framework. To achieve additional yaw rotations, one simply needs to include the desired heading orientation in the aerobatic intention while also incorporating yaw variables as part of trajectory optimization.

Even the most adept FPV freestyle pilots require a familiarity with the flight environment before engaging in aerobatic maneuvers. Our system also relies on known maps, which requires constraining drones to follow preplanned trajectories with known obstacles. In scenarios where detailed maps are unavailable, the trajectories generated by the proposed system can serve as a robust global guidance, allowing for the swift avoidance of unknown obstacles through a real-time, high-frequency replanning module. This strategy, which combines global planning for guidance with local replanning for rapid obstacle avoidance, has been validated for its reliability in numerous studies (*37*–*39*). Furthermore, manually input intentions can directly affect the quality of aerobatic maneuvers because the trajectories generated under extreme intent constraints often prioritize dynamic feasibility. Our experience suggests that a well-defined topology of intentions is more effective than fine-tuned positional adjustments. More examples of aerobatic maneuvers can be found in Supplementary Results. The proposed system uses widely used classical geometric drone controllers (*25*), thereby neglecting model mismatch issues. The consideration of more precise aerodynamic models [i.e., nonlinear model predictive control (*40*) or higher-order residual dynamic models (*7*, *41*)] would likely provide enhanced control outcomes compared with those presented in this article.

Overall, the proposed system is designed to address deficiencies in existing aerobatic flight trajectory planning algorithms, as measured by feasibility and safety in intricate environments. It should also be acknowledged that aerobatic flight poses a systemic challenge, demanding high standards for the perception, localization, planning, and control of aerial vehicles. Designing a comprehensive flight system that fully exploits the potential of these vehicles for achieving autonomous real-time aerobatic flight in unknown environments represents a promising research direction with considerable potential for advancement.

MATERIALS AND METHODS

System architecture

A comprehensive strategy was developed for aerobatic maneuver intention input, trajectory generation, and execution, as shown in Fig. 2. Similar to the way in which experienced aerobatic pilots prefer to familiarize themselves with new environments before performing aerobatic maneuvers, it is necessary to build a 3D map of the environment beforehand (see Fig. 2A). These environmental details not only served as topological references and obstacle constraints for trajectory generation but also provided localization assistance for high-speed drone flights, as detailed in Fig. 1B and the "Corridor generation" section in Supplementary Methods. Aerobatic maneuvers are based on aerobatic intentions, as shown in Fig. 1A and explained in the "Aerobatic parameterization" section. We created a Unity-based visual interactive program for ground station operations. Figure 2C and movie S4 illustrate how the human-machine interface facilitated user input of aerobatic intentions and visualized trajectory outcomes for fine-tuning by the user. Our system's primary feature is an aerobatic planning

module, which generates high-quality trajectories (Fig. 1C). As such, comprehensive explanations are given in the "Trajectory representation" and "Problem formulation" sections. The issue of yaw sensitivity during aerobatic flights was addressed as shown in Fig. 1D, with the "Yaw compensation mapping" section and Supplementary Methods providing essential information. In addition, Supplementary Methods contain further detail concerning system control, localization, and hardware-related system aspects.

Aerobatic parameterization

As discussed in the Introduction, aerobatic maneuvers exhibit distinct and identifiable characteristics known as aerobatic intentions. We categorized these features using trajectory topologies within the environment and attitude changes achieved during the maneuvers, which can be further defined as key waypoints and attitude sequences, respectively. These two features can be expressed as follows

$$\sigma = \{\sigma_1, \ \dots, \sigma_K\} \tag{1}$$

$$\mathbf{P}^{\sigma} = \{\mathbf{p}_1^{\sigma}, \ \dots, \mathbf{p}_K^{\sigma}\} \tag{2}$$

where K is the number of input intentions and $\sigma_k \in S^2$ is a vector in the world frame, indicating the quadrotor thrust direction. The $\mathbf{P}^{\sigma}$ term is a key waypoint, which can be used to generate a flight corridor sequence $\mathcal{F}$ in a known environment $\mathcal{E}$. Taking the power loop as an example, this trajectory typically involves a 360° flip around the obstacle along the pitch axis, as shown in Fig. 1A. According to this description, trajectories around obstacles represent the topology of the trajectory in the environment, which can be further illustrated using a series of specific waypoints. The 360° flip along the pitch axis represents the attitude change in this maneuver, which can also be summarized by a few key attitudes.

Yaw compensation mapping

Yaw compensation involves analyzing specific mathematical formulas and proofs. This section briefly introduces the background that led to this issue ("Differential flatness" section), the driving mechanisms ("Sensitivity around singularity" section), and our primary approach to resolving the issue ("Yaw compensation strategy" section). For more detailed information, please see Supplementary Methods.

Differential flatness

Multicopters exhibit differential flatness properties, which allow the drone's state $\boldsymbol{x}$ and the input $\boldsymbol{u}$ to be expressed as algebraic functions of carefully selected flat outputs $\zeta(t)$ and their derivatives. The differential outputs for quadrotors can typically be expressed as follows

$$\zeta(t) = \{\mathbf{p}(t), \gamma(t)\} = \{p_x(t), p_y(t), p_z(t), \gamma(t)\} \tag{3}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the position of the quadrotor and $\gamma \in SO(2)$ represents the rotation angle that affects the quadrotor's z axis. Differential flatness mapping eliminates differential constraints in the system dynamics equations and reduces the dimensionality of the trajectory representation space, thereby facilitating trajectory generation for drones (*25*, *26*, *34*). For this reason, differential flatness is commonly used in planning and control modules, achieving impressive flight outcomes.

Sensitivity around singularity

One disadvantage of differential flatness is the presence of singularities. Watterson *et al.* (*26*) demonstrated the impossibility of constructing a differential flatness mapping without introducing a singularity. As such, conventional techniques relocate these singularities to regions that are far from the normal flight. For example, a singularity occurs when a robot is pitched at a 90° angle from the hover position (*34*) or when the robot is fully inverted (*26*). In aerobatic maneuvers, states reachable by a drone span the entire $SO(3)$ space, which necessitates addressing these issues. The influence of singularities on differential flatness mapping involves two contributions. First, the quadcopter state and inputs cannot be computed at the singularity. Second, numerical instabilities near the singularity can produce unnecessary yaw rotations in the drone (yaw sensitivity issues), even with a smoothly planned trajectory for γ. Although singularities are mathematically unavoidable, certain engineering practices can avoid undefined states through conditional checks. Our proposed technique primarily addresses numerical instability concerns near these points.

Yaw compensation strategy

Fundamentally, γ represents the rotation angle extending from the quadrotor's z axis, although it cannot be considered as the Euler yaw angle. Previous studies (*30*, *32*, *33*, *42*, *43*) tend to confuse the γ angle in differential flatness with the Euler yaw angle because the two are nearly equivalent during low-speed stable flight. However, the disparities between them become more apparent as the flight angle increases, bringing the aircraft closer to the singularity range. As such, we refrained from simplistically treating γ as the yaw angle for planning purposes during aerobatic maneuvers. Here, we introduce the variable $\mathbf{d}(t) \in \mathbb{R}^3$ to represent the quadrotor heading (instead of the angle γ) as follows

$$\gamma(t) = \mathcal{D}(\mathbf{d}(t), \mathbf{p}(t)) \tag{4}$$

This expression allows us to represent the differential flatness variable $\zeta(t)$ as a function of $\tilde{\zeta}(t) = \{\mathbf{p}(t), \mathbf{d}(t)\}$. By constraining and optimizing the variables with physical significance ($\mathbf{d}$), we can ensure that sensitivity issues around the singularity do not affect the heading during aerobatic flight. It is worth noting that $\mathbf{d}$ and $\mathbf{p}$ are completely decoupled, indicating that after introducing $\mathbf{d}$, rotation around the z axis remains unrestricted. In aerobatic maneuvers, the quadrotor's head direction always aligns with the forward direction. Hence, we directly define the heading direction as the velocity direction (i.e., $\mathbf{d} = \dot{\mathbf{p}}$).

Trajectory representation

A 3D M-piece polynomial spline $\mathbf{p}(t) = \{\mathbf{p}_1(t), \ \dots, \mathbf{p}_M(t)\} \in \mathbb{R}^3$ was used to represent the flat-output trajectory. Here, $\mathbf{T} = \{T_1, \dots, T_M\} \in \mathbb{R}^M_{>0}$ represented the durations of all pieces, and $T_\Sigma = \sum_{i=1}^{M} T_i$ denoted the total time for the whole trajectory. We defined the ith piece spline as a D-degree polynomial

$$\mathbf{p}_i(t) = \mathbf{c}_i^T \beta(t), \ \forall t \in [0, T_i] \tag{5}$$

where $D = 2s - 1$ and s are the order of the relevant integrator chain, $\mathbf{c}_i \in \mathbb{R}^{2s \times 3}$ is the coefficient matrix for each piece, and $\beta(t) = \left[t^0, t^1, \dots t^D\right]^T$ is the natural basis. The spatial and temporal profiles of the trajectory were efficiently deformed during optimization by adopting the minimum control effort (MINCO) (*27*) as the trajectory representation method. The core feature of MINCO is a linear-complexity smoothing map $\mathbf{c} = M(\mathbf{q}, \mathbf{T})$, where $\mathbf{q} = \{\mathbf{q}_1, \dots, \mathbf{q}_{M-1}\}$ are the intermediate waypoints. According to theorem 2 in (*27*), this mapping directly constructs a minimum control trajectory for a 3D s-integrator chain using any specified initial and terminal conditions. In this way, by optimizing the

trajectory parameters (**q** and **T**), we can change the shape of the trajectory in the spatial and temporal dimensions.

Problem formulation

The trajectory optimization problem can be formulated as follows

$$\min_{\mathbf{q},\mathbf{T},\mathcal{T}} H = \sum_{i=1}^{M} \int_{0}^{T_i} \left\| \mathbf{p}_i^{(s)}(t) \right\|^2 dt + \lambda_t T_\Sigma \tag{6A}$$

$$\text{such that } \mathcal{G}(\mathbf{q},\mathbf{T}) \preceq \mathbf{0}, \forall t \in \left[0, T_\Sigma\right] \tag{6B}$$

$$C(\mathbf{q},\mathbf{T}) = \mathbf{0}, \forall t \in \left[0, T_\Sigma\right] \tag{6C}$$

$$\mathcal{A}\left(\mathbf{q},\mathbf{T},t_k^\sigma\right) = \mathbf{0}, \forall t_k^\sigma \in \mathcal{T} \tag{6D}$$

where the function in Eq. 6A represents the control effort and time allocation. We can adjust the weight λ_t to achieve a trade-off between trajectory smoothness and aggressiveness. Equations 6B and 6C represent the inequality and equality constraints related to the trajectory parameters (**q** and **T**), specifically encompassing dynamic constraint, corridor constraint, and others. Equation 6D ensures compliance with aerobatic constraints, where t_k^σ denotes the time for an aerobatic intention along the entire trajectory and $\mathcal{T} = \{t_1^\sigma, \ldots, t_K^\sigma\}$. We removed all constraints in Eqs. 6A to 6D by formulating penalty functions or substituting variables. The resulting trajectory generation problem can thus be reformulated as an unconstrained optimization problem

$$\min_{\mathbf{q},\mathbf{T},\mathcal{T}} H = \sum_{\star} \lambda_\star \mathcal{J}_\star \tag{7}$$

where $\lambda_\star$ is the weight of the penalty function $\mathcal{J}_\star$ and the subscripts $\star = \{e, t, a, c, d\}$ represent the control effort e, time allocation t, aerobatic intention a, corridor topology c, and dynamic feasibility d. This unconstrained problem can be solved using an efficient optimizer called L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno) (*44*). The detailed definitions of all constraints, along with the transformation into an unconstrained optimization function, can be found in Supplementary Methods.

Statistical analysis

We used three statistical analysis methods: bar charts (Fig. 7B), box plots (Fig. 4C and fig. S6C), and violin plots (fig. S7B). The bar chart displays the mean of the data. The box plot summarizes the data distribution by showing the median and interquartile range (IQR), with whiskers extending to 1.5 times the IQR; data points beyond this range are considered outliers and are represented in the figure. Violin plots are a hybrid visualization method that combines the distribution information of kernel density estimation with key summary statistics from box plots. Each "violin" shape has two symmetrical halves representing the probability density of the data at different values, with wider sections indicating higher probability concentrations. The kernel density estimator is the estimated probability density function of a random variable. For any real values of x, the kernel density estimator's formula is given by

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{8}$$

where x_i are random samples from an unknown distribution, n is the sample size, h is the bandwidth parameter, and $K(\cdot)$ is the kernel smoothing function, and we used Gaussian kernel in our analysis. All original data can be found in our publicly available datasets (*45*), with the sample sizes for each statistical analysis indicated in the figure captions.

Supplementary Materials

The PDF file includes:

Methods
Table S1
Results
Figs. S1 to S7
References (*46*–*60*)

Other Supplementary Material for this manuscript includes the following:

Movies S1 to S5

REFERENCES AND NOTES

1. BBC, "How sparrowhawks catch garden birds - life in the air: Episode 2 preview" (2016); www.youtube.com/watch?v=Ra6l6svXQPg&ab channel=BBC&t=84.
2. otk3244, "Sony a1 impressive peregrine hunt 2207 08" (2016); www.youtube.com/watch?v=IuP8UMm2k2Y&abchannel=otk3244&t=136.
3. National Geographic, "How do bats land upside down?" (2015); www.youtube.com/watch?v=342Y_040f1Y&ab_channel=NationalGeographic.
4. Exploring wildlife with Vance Crofoot, "Young ravens displaying skilled aerial maneuvers" (2022); www.youtube.com/watch?v=I8CJAXu1p-s&ab channel=ExploringwildlifewithVanceCrofoot&t=25.
5. D. Tezza, D. Caprio, D. Laesker, M. Andujar, "Let's fly! An analysis of flying fpv drones through an online survey" in *Interdisciplinary Workshop on Human-Drone Interaction 2020* (ACM SIGCHI, 2020).
6. DJI, "DJI enterprise" (2024); https://enterprise.dji.com/.
7. E. Kaufmann, L. Bauersfeld, A. Loquercio, M., V., D. Müller, Koltun, Scaramuzza, Champion-level drone racing using deep reinforcement learning. *Nature* **620**, 982–987 (2023).
8. X. Zhou, Z. Wang, H. Ye, C. Xu, F. Gao, Ego-planner: An ESDF-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **6**, 478–485 (2020).
9. Y. Song, A. Romero, M., V., D. Müller, Koltun, Scaramuzza, Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Sci. Robot.* **8**, eadg1462 (2023).
10. Y. Chen, N. O. Pérez-Arancibia, Controller synthesis and performance optimization for aerobatic quadrotor flight. *IEEE Trans. Control Syst. Technol.* **28**, 2204–2219 (2020).
11. S. Lupashin, A. Schöllig, M. Sherback, R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips" in *2010 IEEE International Conference on Robotics and Automation* (IEEE, 2010), pp. 1642–1648.
12. G. Lu, W. Xu, F. Zhang, On-manifold model predictive control for trajectory tracking on robotic systems. *IEEE Trans. Ind. Electron.* **70**, 9192–9202 (2022).
13. E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, D. Scaramuzza, "Deep drone acrobatics," in *Proceedings of Robotics: Science and Systems XVI* (RSS Foundation, 2020).
14. C. Mollica, *FPV Flight Dynamics: Mastering Acro Mode on High-Performance Drones* (Vespula Ventures LLC, 2020).
15. B. E. Jackson, K. Tracy, Z. Manchester, Planning with attitude. *IEEE Robot. Autom. Lett.* **6**, 5658–5664 (2021).
16. E. Tal, G. Ryou, S. Karaman, Aerobatic trajectory generation for a VTOL fixed-wing aircraft using differential flatness. *IEEE Trans. Robot.* **39**, 4805–4819 (2023).
17. G. Lu, Y. Cai, N. Chen, F. Kong, Y. Ren, F. Zhang, Trajectory generation and tracking control for aggressive tail-sitter flights. *Int. J. Robot. Res.* **43**, 241–280 (2024).
18. T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **34**, 1004–1020 (2018).
19. K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, V. Kumar, Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robot. Autom. Lett.* **3**, 965–972 (2018).

20. T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, Lio-sam: "Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020), pp. 5135–5142.
21. W. Xu, F. Zhang, Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robot. Autom. Lett.* **6**, 3317–3324 (2021).
22. Vicon, "Creativity in flight - Georgia Tech aerospace" (2024); www.vicon.com/resources/case-studies/creativity-in-flight-georgia-tech-aerospace/.
23. "Global Positioning System" (2024); https://de.wikipedia.org/wiki/Global_Positioning_System.
24. H. Moravec, A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation* (IEEE, 1985), vol. 2, pp. 116–121.
25. T. Lee, M. Leok, N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on se (3)" in *49th IEEE Conference on Decision and Control (CDC)* (IEEE, 2010), pp. 5420–5425.
26. M. Watterson, V. Kumar, "Control of quadrotors using the Hopf fibration on so (3)" in *Robotics Research: The 18th International Symposium ISRR* (Springer, 2020), pp. 199–215.
27. Z. Wang, X. Zhou, C. Xu, F. Gao, Geometrically constrained trajectory optimization for multicopters. *IEEE Trans. Robot.* **38**, 3259–3278 (2022).
28. B. Zhou, F. Gao, L. Wang, C. Liu, S. Shen, Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **4**, 3529–3536 (2019).
29. Y. Ren, F. Zhu, W. Liu, Z. Wang, Y. Lin, F. Gao, F. Zhang, "Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022), pp. 6332–6339.
30. D. Falanga, E. Mueggler, M. Faessler, D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 5774–5781.
31. J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, V. Kumar, Aggressive flight with quadrotors for perching on inclined surfaces. *J. Mech. Robot.* **8**, 051007 (2016).
32. J. Ji, T. Yang, C. Xu, F. Gao, "Real-time trajectory planning for aerial perching," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022), pp. 10516–10522.
33. Y. Gao, J. Ji, Q. Wang, R. Jin, Y. Lin, Z. Shang, Y. Cao, S. Shen, C. Xu, F. Gao, Adaptive tracking and perching for quadrotor in dynamic scenarios. *IEEE Trans. Robot.* **40**, 499–519 (2023).
34. D. Mellinger, V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation* (IEEE, 2011), pp. 2520–2525.
35. B. Morrell, M. Rigter, G. Merewether, R. Reid, R. Thakker, T. Tzanetos, V. Rajur, G. Chamitoff, "Differential flatness transformations for aggressive quadrotor flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 5204–5210.
36. NOKOV, "Nokov—Optical motion capture system" (2022); https://en.nokov.com/.
37. F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, S. Shen, Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments. *IEEE Trans. Robot.* **36**, 1526–1545 (2020).
38. Q. Wang, D. Wang, C. Xu, A. Gao, F. Gao, "Polynomial-based online planning for autonomous drone racing in dynamic environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023), pp. 1078–1085.
39. D. Wang, H. Ye, N. Pan, J. Huang, B. Zhang, Y. Mao, G. Huang, C. Xu, F. Gao, "Flexible and topological consistent local replanning for multirotors," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2024), pp. 5348–5355.
40. G. Torrente, E. Kaufmann, P. Föhn, D. Scaramuzza, Data-driven MPC for quadrotors. *IEEE Robot. Autom. Lett.* **6**, 3769–3776 (2021).
41. M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, S.-J. Chung, Neural-fly enables rapid learning for agile flight in strong winds. *Sci. Robot.* **7**, eabm6597 (2022).
42. Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu, F. Gao, Fast-racing: An open-source strong baseline for SE(3) planning in autonomous drone racing. *IEEE Robot. Autom. Lett.* **6**, 8631–8638 (2021).
43. D. Mellinger, N. Michael, V. Kumar, Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* **31**, 664–674 (2012).
44. D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528 (1989).
45. M. Wang, Q. Wang, Z. Wang, Y. Gao, J. Wang, C. Cui, Y. Li, Z. Ding, K. Wang, C. Xu, F. Gao, Dataset - Unlocking aerobatic potential of quadcopters: Autonomous freestyle flight generation and execution (2024); https://zenodo.org/records/11240812.
46. Z. Wang, C. Xu, F. Gao, "Robust trajectory planning for spatial-temporal multi-drone coordination in large scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022), pp. 12182–12188.
47. Y. Wang, J. Ji, Q. Wang, C. Xu, F. Gao, "Autonomous flights in dynamic environments with onboard vision," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021), pp. 1966–1973.
48. Q. Wang, Z. Wang, M. Wang, J. Ji, Z. Han, T. Wu, R. Jin, Y. Gao, C. Xu, F. Gao, Fast iterative region inflation for computing large 2-d/3-d convex regions of obstacle-free space. arXiv:2403.02977 [cs.RO] (2024).
49. TMOTOR, "F1404 FPV racing drone motor kv2900/kv3800/kv4600" (2024); https://www.getfpv.com/t-motor-f1404-2900kv-3800kv-4600kv-motor-gray-red.html.
50. TMOTOR, "F2203.5 FPV racing drone motor 3-6s kv1500/kv2850/kv3550" (2024); https://www.getfpv.com/t-motor-f2203-5-1500kv-2850kv-3550kv-motor.html.
51. Holybro, "Tekko32 f4 4in1 mini 50a esc" (2024); https://holybro.com/collections/fpv-esc/products/tekko32-f4-4in1-mini-50a-esc.
52. Holybro, "Kakute h7 mini" (2024); https://holybro.com/products/kakute-h7-mini.
53. KHADAS, "Vim3, amlogic a311d sbc with 5.0 tops npu" (2024); https://www.khadas.com/vim3.
54. NVIDIA, "Robotics and edge computing, jetson-orin" (2024); https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/.
55. LIVOX, "Mid-360, small but mighty" (2022); https://www.livoxtech.com/mid-360.
56. 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).
57. S. Liu, K. Mohta, N. Atanasov, V. Kumar, Search-based motion planning for aggressive flight in se (3). *IEEE Robot. Autom. Lett.* **3**, 2439–2446 (2018).
58. PX4, "Multicopter pid tuning guide (manual/advanced)—Thrust curve" (2024); https://docs.px4.io/main/en/config_mc/pid_tuning_guide_multicopter.html#thrust-curve.
59. MATLAB, "Curve fitting toolbox — fit curves and surfaces to data using regression, interpolation, and smoothing" (MathWorks, 2024); www.mathworks.com/products/curvefitting.html.
60. M. C. Campi, Exponentially weighted least squares identification of time-varying systems with white disturbances. *IEEE Trans. Signal Process.* **42**, 2906–2914 (1994).

Acknowledgments: We express our gratitude to Y. Wang, X. Zhou, Y. Wang, and T. Wu for the invaluable suggestions on the manuscript; N. Pan, X. Zhou, and C. Li for the contribution in system hardware references; and R. Jin, M. Zhang, and C. Ma for providing photography and videography services. Our heartfelt thanks go out to S. Ji, F. Yang, J. Guo, X. Zhou, and other fellow students from the FASTLAB who assisted during outdoor experiments. We would like to express our sincere appreciation to T. Lu, W. Qin, and H. Xu from the Xiwang Dong Research Group at Beihang University for the support in terms of venue and personnel. **Funding:** This work was supported by the National Natural Science Foundation of China under grant no. 62322314, the National Key R&D Program of China under grant no. 2023YFB4706600, the National Natural Science Foundation of China under grant no. 62088101, and the "Pionee" and "Leading Goose" R&D Program of Zhejiang under grant no. 2024C01170. **Author contributions:** M.W. participated in the design and implementation of both software and hardware, conducted all experiments, and contributed to manuscript writing. Q.W. offered algorithmic support for the rapid generation of flight corridors. Q.W. and Z.W. were involved in the development and debugging of the localization module for outdoor aerobatic flight, provided consultation, and edited the manuscript. Y.G. was engaged in photography, graphic design, and video editing. J.W. and C.C. designed the interactive visualization software using Unity. Y.L. and Z.D. assisted in the early indoor aerobatic flight experiments. K.W. and C.X. provided people and sites for drone testing. F.G. supervised the research, provided financial support, offered crucial advice on software and system debugging, and revised the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** The data and source code supporting the conclusions of this study are available at https://doi.org/10.5281/zenodo.14586487 (*45*) and https://github.com/ZJU-FAST-Lab/Aerobatic-Planner.

Submitted 26 May 2024
Accepted 18 March 2025
Published 16 April 2025
10.1126/scirobotics.adp9905

Science Robotics

Unlocking aerobatic potential of quadcopters: Autonomous freestyle flight generation and execution

Mingyang Wang, Qianhao Wang, Ze Wang, Yuman Gao, Jingping Wang, Can Cui, Yuan Li, Ziming Ding, Kaiwei Wang, Chao Xu, and Fei Gao

Sci. Robot. **10** (101), eadp9905. DOI: 10.1126/scirobotics.adp9905

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.