SCIENCE ROBOTICS | RESEARCH ARTICLE

NAVIGATION

A self-rotating, single-actuated UAV with extended sensor field of view for autonomous navigation

Nan Chen, Fanze Kong, Wei Xu, Yixi Cai, Haotian Li, Dongjiao He, Youming Qin, Fu Zhang*

Uncrewed aerial vehicles (UAVs) rely heavily on visual sensors to perceive obstacles and explore environments. Current UAVs are limited in both perception capability and task efficiency because of a small sensor field of view (FoV). One solution could be to leverage self-rotation in UAVs to extend the sensor FoV without consuming extra power. This natural mechanism, induced by the counter-torque of the UAV motor, has rarely been exploited by existing autonomous UAVs because of the difficulties in design and control due to highly coupled and nonlinear dynamics and the challenges in navigation brought by the high-rate self-rotation. Here, we present powered-flying ultra-underactuated LiDAR (light detection and ranging) sensing aerial robot (PULSAR), an agile and self-rotating UAV whose three-dimensional position is fully controlled by actuating only one motor to obtain the required thrust and moment. The use of a single actuator effectively reduces the energy loss in powered flights. Consequently, PULSAR consumes 26.7% less power than the benchmarked quadrotor with the same total propeller disk area and avionic payloads while retaining a good level of agility. Augmented by an onboard LiDAR sensor, PULSAR can perform autonomous navigation in unknown environments and detect both static and dynamic obstacles in panoramic views without any external instruments. We report the experiments of PULSAR in environment exploration and multidirectional dynamic obstacle avoidance with the extended FoV via self-rotation, which could lead to increased perception capability, task efficiency, and flight safety.

INTRODUCTION

Uncrewed aerial vehicles (UAVs) have played an increasingly important role in a variety of real-world applications such as search and rescue, cave survey, building mapping, and archeological exploration. To fulfill the task requirements in these applications, the abilities of self-localization, environment mapping, and obstacle avoidance are key. These abilities are usually based on the environmental observation provided by visual sensors on board the UAV, passive (e.g., RGB camera and thermal camera) or active [e.g., light detection and ranging (LiDAR) and infrared depth camera]. An existing issue for UAVs is that the small field of view (FoV) of these sensors severely limits the UAV's perception capability and task efficiency. Although many efforts have been made to deal with the constraints in applications induced by narrow or limited FoVs (*1*–*5*), a larger FoV is still a better solution that not only reduces task time by observing the environment more efficiently (*6*) but also enhances the UAV safety in the wild by perceiving dynamic obstacles (e.g., birds) approaching from an unknown direction (*7*).

Methods for extending sensor FoVs on UAVs

Because of the advantages mentioned above, many methods have been proposed to extend sensor FoVs on UAVs. One method is to use a sensor with a large FoV, such as a fisheye camera (*8*), catadioptric camera (*9*), and 360° LiDAR (*10*). However, for the fisheye camera and the catadioptric camera, obvious distortions often occur and must be properly compensated. There is also a limitation to the installation direction due to the hemispherical FoV (*11*). A 360° LiDAR has all 360° FoVs in the horizontal direction, but the available vertical FoV is still narrow and has rather low resolution (*12*). Using multiple sensors, such as cameras (*13*, *14*), stereo cameras (*15*, *16*), fisheye cameras (*6*, *7*), or LiDARs (*17*, *18*), is another method for obtaining larger FoVs. However, these multisensor systems lead to additional sensor cost and processing time (*19*). Moreover, the component weights due to the sensors and their processing units lead to more power consumption during a UAV flight. These problems also exist for a UAV gimbal system extending sensor FoV.

Instead of merely increasing the sensor quantity (e.g., multisensor system) or using additional actuation (e.g., gimbal system), a more natural and power-saving method for extending the sensor FoV is to use the inherent motions that the UAV is already capable of. One such motion is self-rotation, which effectively extends the sensor FoV to the full 360° without sacrificing the ability to move in three-dimensional (3D) space (*20*). Nevertheless, self-rotation brings two notable challenges: One is the UAV design and control that should, on the one hand, maximally use the inherent UAV motion without adding extra powered actuators and, on the other hand, cope with highly coupled and nonlinear dynamics. The other challenge resides in navigation: The high-rate rotation causes severe motion blurs and rapid FoV change, which markedly degrades the performance and reliability of common visual navigation systems.

Self-rotating UAVs

To date, several self-rotating UAV designs with a varying number of actuators have been explored, such as two-actuator designs (*20*–*27*) and one-actuator designs (*28*–*33*). The designs in (*21*, *22*) used two motors to provide the lift and moment, where the lift controlled the altitude and the moment controlled the attitude (hence horizontal position), hence achieving full 3D position control of the UAV. The designs in (*20*, *23*–*27*) used a motor to provide the UAV lift and a servo to control the attitude. Compared with these two-actuator

---

Department of Mechanical Engineering, University of Hong Kong, Pokfulam, Hong Kong, China.
*Corresponding author. Email: fuzhang@hku.hk

designs, a one-actuator design enjoys simpler mechanical structure, reduced energy loss, and lower component weight, which all help to conserve power consumption during a flight. The single-servo design in (*28*) used a servo to steer the UAV horizontal position and left the UAV altitude uncontrolled because of the lack of lift (unpowered flight). Likewise, the one-motor design in (*29*) used a motor to provide the UAV lift but left the UAV attitude (and hence horizontal position) uncontrolled. One-motor designs achieving full 3D position control are found in (*30*–*33*), where the lift was provided by the motor thrust (*30*, *31*) or wing aerodynamic lift (*32*, *33*) and the attitude was controlled by cyclical adjustment of motor thrust (*30*, *31*) or wing lift (*32*, *33*), respectively, during each period of self-rotation. These designs required a particular geometry and sophisticated mass distribution to trim the system, making it difficult for them to carry extra payload. Moreover, the thrust or wing lift was adjusted only once at every body self-rotation, fundamentally limiting the control rate and degrading the UAV agility and control accuracy.

Self-navigation under self-rotation

The ability of self-navigation is crucial for UAVs to perform autonomous flights in unknown and Global Navigation Satellite System (GNSS)–denied environments, such as caves, urban canyons, and dense forests. However, severe motion blurs and rapid FoV changes due to high-rate rotation in self-rotating UAVs cause great challenges for common visual-based navigation. For this reason, all self-rotating UAVs (*20*–*33*) reviewed above lacked the ability of autonomous navigation in unknown environments. The work in (*21*, *29*) mainly focused on mechantronics and control without considering any navigation. In (*20*), a dedicated onboard camera with a frame rate of more than 500 Hz was used to estimate the UAV attitude by optical flow techniques, but the other UAV states, such as position and velocity, were not considered. Similarly, in (*23*, *30*), an onboard infrared phototransistor or photodiode array was installed to estimate the UAV heading angle by sensing infrared sources or optical sources in the environment, respectively. The other UAV states other than heading angle still remained unsolved. Full UAV state estimations were made in (*22*, *24*, *31*–*33*), but they all relied on external motion capture systems to provide the position and attitude measurements, so they could only be used in indoor instrumented environments. Likewise, Fregene *et al.* (*25*–*27*) and Win *et al.* (*28*) estimated the full UAV states by leveraging external position measurements from GPS or radio frequency instruments, limiting their use in GNSS-denied environments. Besides the

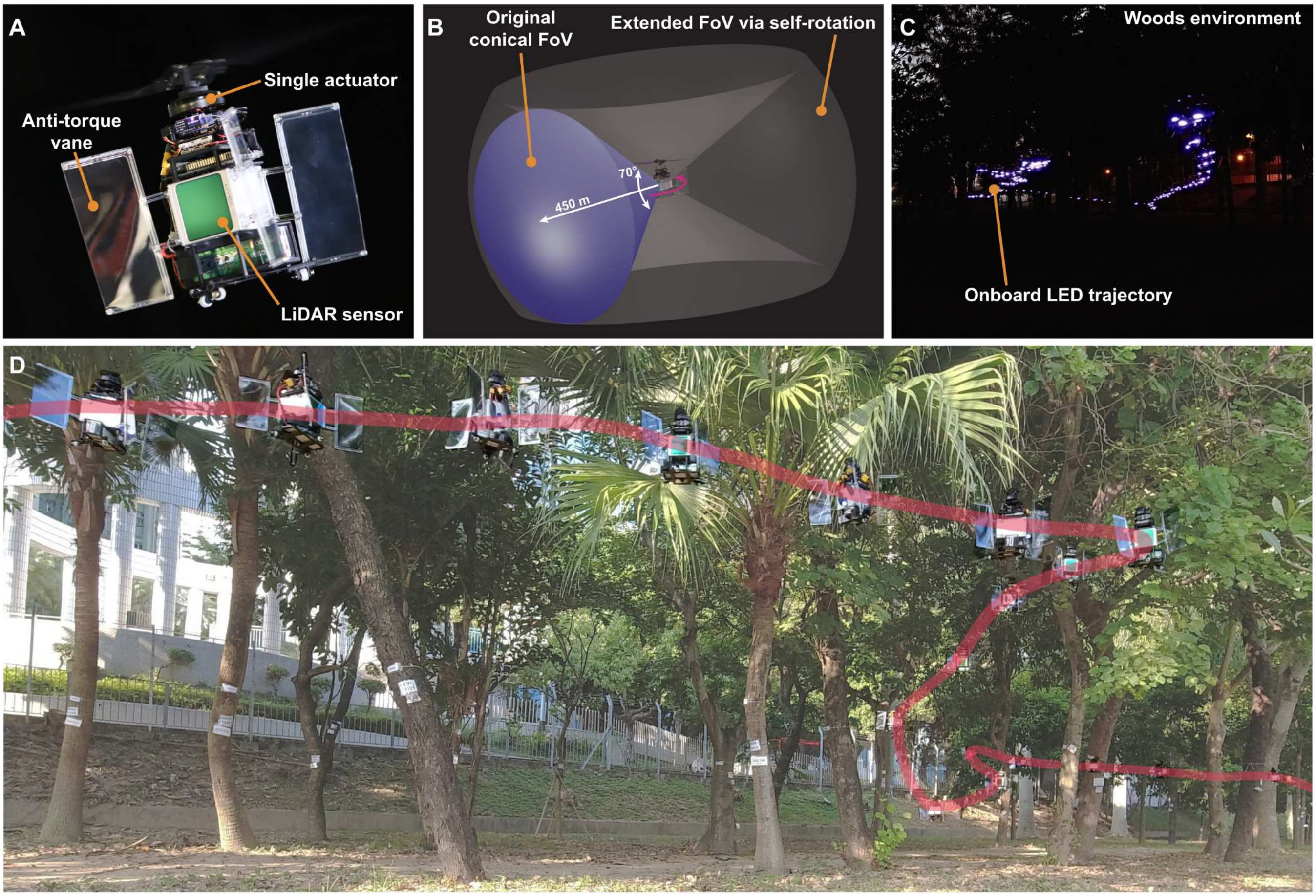


Fig. 1. Overview of powered-flying ultra-underactuated LiDAR sensing aerial robot (PULSAR). (**A**) PULSAR uses one actuator (i.e., a motor) for full 3D position control and an onboard LiDAR sensor for autonomous navigation. (**B**) The uncompensated motor counter-torque naturally causes a self-rotation that extends the sensor horizontal FoV to 360°. (**C**) Autonomous flights of PULSAR in an unknown wooded environment at night; the flight trajectory is indicated by the onboard blue light-emitting diode (LED). (**D**) Autonomous flights in the woods in the daytime; the flight trajectory is shown as the red path.

intended self-rotation in (*20*–*33*), unexpected self-rotation could also occur in common quadrotors in case of partial rotor failures. Sun *et al.* (*34*) estimated the full state of a self-rotating quadrotor using down-facing cameras (either standard frame or event cameras). Down-facing cameras ease the state estimation (by reserving large FoV overlaps) but completely relinquish the FoV extension brought by the self-rotation. Besides state estimation, existing self-rotating UAVs (*20*–*34*) lacked the ability of 3D environment mapping and hence could not navigate in unknown environments, not to mention take advantage of the extended FoV.

Proposed solution

To solve the problems in design, control, and navigation mentioned above and to take the advantages of single-actuator design at the same time, we propose an autonomous, single-actuated, and self-rotating UAV with extended sensor FoVs. The UAV (Fig. 1A and Movie 1) uses a single actuator (motor) to control its full 3D position. The motor average rotation speed determines the propeller thrust controlling the UAV altitude, and the motor acceleration profile within each revolution determines the blade pitch angle, which induces a moment controlling the UAV attitude and hence horizontal position. Unlike the cyclical thrust adjustment in (*30*–*33*), the blade pitch angle in our UAV is adjusted once every propeller rotation (as opposed to body rotation) by changing the motor speed, leading to a high control rate that increases the overall UAV agility and control accuracy. Meanwhile, the motor counter-torque naturally drives the UAV body to rotate, which uses the inherent motion to extend sensor FoV without adding extra powered actuators (Fig. 1B). Using the minimum number of actuators, the proposed UAV has a simple mechanical structure and effectively reduces the energy loss and component weight caused by actuators, which all help to decrease the overall power consumption. Moreover, the symmetric UAV structure allows easy mounting of payloads, such as a 3D LiDAR sensor and an onboard computer, without changing the system controllability. By taking advantage of the active and high-rate LiDAR measurements, the proposed self-rotating UAV is able to navigate fully autonomously in unknown, GNSS-denied environments during both day and night (Fig. 1, C and D). Because the sweeping of the conical LiDAR FoV induced by self-rotation is intuitively similar to the beams of radiation emitted by an astronomical pulsar, we named this UAV "PULSAR," which is also an acronym for powered-flying ultra-underactuated LiDAR sensing aerial robot.

We demonstrated the effectiveness of PULSAR in terms of power efficiency, agility, and self-navigation ability and in both indoor and outdoor environments. In all experiments, the UAV relied fully on its onboard sensor and computer to estimate its states and to perceive the environment without using any external instruments. Experiments showed that PULSAR consumed 26.7% less power than a benchmarked quadrotor with the same total propeller disk area and avionic payloads while maintaining comparable agility in terms of trajectory tracking errors. PULSAR was responsive to external commands and showed high robustness to external disturbances, such as wind gusts, making it suitable for real-world operations. With a small onboard computer running an entire navigation framework in real time, PULSAR successfully demonstrated autonomous flights in a cluttered, GNSS-denied environment. Moreover, benefiting from its extended FoV, PULSAR was capable of detecting and avoiding dynamic obstacles in various directions beyond the original sensor FoV.

Movie 1. Summary of PULSAR. This video shows the LiDAR scan during self-rotation, the process of control moment generation, and the real-world flights in both indoor and outdoor environments of PULSAR.

RESULTS

UAV system overview

Shown in Fig. 2A, PULSAR has an overall weight of 1.23 kg, of which 50% (616 g) is contributed by the payload (i.e., 3D LiDAR, onboard computer, and wires). At static, the motor drives the propeller to rotate up to 6800 rpm, providing a maximum thrust of 25.45 N or a thrust-to-weight ratio of 2.1. PULSAR has a diameter of 37.6 cm, which is also the propeller diameter, and a height of 23.7 cm. By actuating the single motor and using the onboard sensor and computer, PULSAR achieves full 3D position control and completely autonomous flights in unknown environments.

Mechanical design

The mechanical design of PULSAR is shown in Fig. 2A. It mainly consists of three modules: a flight control module at the top (i.e., propeller, motor, and flight controller), a 3D LiDAR sensor with an onboard computer in the middle, and a battery chassis and landing gears at the bottom. This modular design allows the UAV to be easily maintained and reconfigured (see fig. S9). The holding structure for each module was 3D-printed with nylon materials, leading to a compact and rigid structure. The counter-torque of the motor drives the entire UAV body attached with the motor stator to rotate in the opposite direction of the propeller rotation, which then counteracts the propeller speed in an inertial frame and hence reduces the propeller thrust. To mitigate this effect, we used four anti-torque vanes to provide air drag that compensated for the motor counter-torque and hence restricted the body rotation rate to around 2.7 Hz. The landing gear consists of three wheels formed into a circle such that the UAV can start rotating on the ground before takeoff.

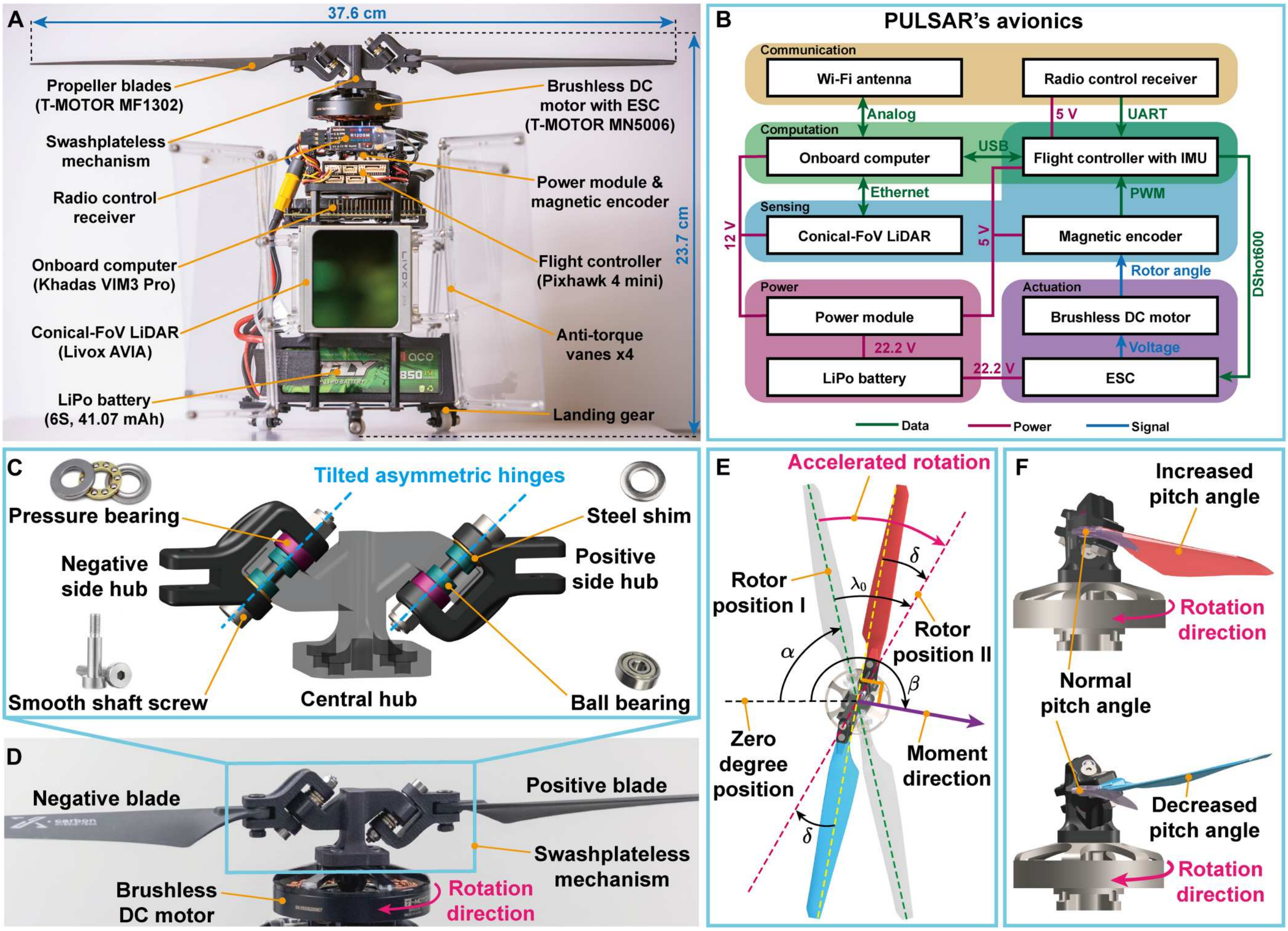


Fig. 2. Mechanical structure, avionics, and swashplateless mechanism. (**A**) Components description. (**B**) Interconnection among all electronic components. (**C**) Detailed mechanical structure of the swashplateless mechanism. (**D**) Assembly of the swashplateless mechanism, propeller blades, and motor. (**E**) Process of rotor acceleration causing the blades to lag from the rotor. With the acceleration voltage being applied at rotor position I (i.e., angle α), the rotor accelerates and reaches rotor position II (i.e., angle $\alpha + \lambda_0$). Because the hinges between the blades and the rotor provide an extra rotation degree of freedom, the blade inertia will cause it to rotate along the hinge and produce a blade lag angle (i.e., δ) behind the rotor. The actual blade position after the acceleration is at position angle $\alpha + \lambda_0 - \delta$, as indicated by the yellow dashed line. (**F**) Because of the tilted asymmetric hinge design, a lag from the rotor will cause the red blade to increase its pitch angle and the blue blade to decrease its pitch angle from the nominal pitch angle indicated by the purple blade. The differential pitch angle changes lead to a net moment $\mathbf{M}_C$ within the propeller disk plane with a direction perpendicular to the actual blade position [i.e., angle $\alpha + \lambda_0 - \delta + \pi/2$, as indicated by the purple arrow in (E)].

Avionics

The avionics of PULSAR are all off-the-shelf available. As shown in Fig. 2B, they are divided into five modules. The entire system is powered by a six-cell 41.07-Wh (1850-mAh) battery. The onboard computer is a Khadas VIM3 Pro with an ARM processor, which runs the core navigation modules in real time. The navigation module plans a trajectory that is sent to a flight controller (Pixhawk 4 Mini) for tracking. The sensing module contains a Livox AVIA LiDAR, which measures 3D point clouds at a frequency of 240,000 points per second in a conical FoV of about 70° (Fig. 1B). Besides the LiDAR, a magnetic encoder (AS5600) was installed on the bottom of the motor stator (see the Supplementary Materials), which measures the rotor's angular position (i.e., rotor angle) at 910 Hz. This provides about 11 measurements per motor revolution when the stator is at 5000 rpm (i.e., 83.3 r/s). The actuation module contains a motor (T-MOTOR MN5006 KV450) driven by a common electronic speed controller (ESC) of the model CYCLONE 45A. The motor command is computed by the flight controller once a rotor angle measurement (by the magnetic encoder) arrives at the pulse-width modulation (PWM) capture port. The computed motor command is lastly sent to the ESC via DShot600 protocol, which enables up to 33.3-kHz command signal transmission.

Swashplateless mechanism

The key component that enables the modulation of a moment on the propeller is a swashplateless mechanism invented in (*35*). As shown in Fig. 2 (C and D), the mechanism consists of two passive hinges connecting two side hubs to the central hub attached to the motor stator. The two hinges are parallel; both are tilted from the motor axis by 45°. Such a specially designed mechanism is able to adjust the cyclic pitch angle of the two blades clamped on the side hubs, a function that has been traditionally realized by a complicated and high-cost swashplate mechanism in helicopters. The change of cyclic pitch angle causes a moment in the propeller disk plane that can then control the UAV's attitude in pitch and roll (movie S1). The working principle of this mechanism and its driving method are described in more detail in Materials and Methods. In our implementation, the swashplateless mechanism was 3D-printed with tough polylactic acid materials, and the propeller blades are standard T-MOTOR 13-inch MF1302 propeller blades. For the swashplateless mechanism to induce moment more easily, the motor T-MOTOR MN5006 KV450 was chosen according to a quantitative analysis detailed in the Supplementary Materials.

The original swashplateless design in (*35*, *36*) uses a hinge directly contacting the moving parts, which brings obvious friction on the contact surface. The friction then prevents the blade pitch angle from responding to small motor acceleration commands, resulting in a "deadband phenomenon" (i.e., no moment output when the inputed command is lower than a threshold) in this mechanism. The deadband phenomenon brings nonlinearity to the system and considerably degrades the overall control performance and system agility. Moreover, the friction also causes energy loss and material wear, which lowers the power efficiency and reliability. To overcome this issue, we used four ball bearings between the central hub and the smooth shaft screws and two pressure bearings and two steel shims between the central hub and the side hubs. They can considerably reduce the friction in contact surfaces, even under high-stress conditions caused by high-rate rotation of the propeller and cyclic twisting of the asymmetric hinges. Consequently, the propeller moment shows no visible deadband phenomenon (fig. S2).

Experimental validation

We performed various real-world experiments to verify the performance of PULSAR. In all experiments, PULSAR used the same LiDAR-inertial odometry and trajectory-tracking controller to estimate its full state and track the trajectory commands, respectively (Fig. 3A). The trajectory commands were generated by the onboard trajectory planner, a dynamic obstacle detector and planner, or the preset trajectory library, depending on the experiment purposes. The software framework overview is described in Materials and Methods.

Flight efficiency

To verify the flight efficiency of PULSAR, we compared its power consumption with two benchmarked quadrotors both carrying the same avionic payloads (i.e., companion computer and LiDAR sensor): One quadrotor has 7.5-inch propellers, leading to the same total disk area as PULSAR, and the other has 8-inch propellers, leading to 16.8% more disk area than PULSAR (fig. S5). For each benchmarked quadrotor, we selected the propulsion systems with the best efficiency among six motors and nine propellers (see the "The benchmarked quadrotors" section in the Supplementary Materials).

We validated the efficiency improvement of PULSAR against the benchmarked quadrotors in actual hover flights. In the experiment, the trajectory command was set to a stationary hovering position, as shown in movie S2 (trajectory commands switched to T3 in Fig. 3A). Results in Fig. 4 and table S1 show that the total power consumption of PULSAR is 26.7% less than that of the 7.5-inch propeller quadrotor (which has a similar propeller disk area) and 17.3% less than that of the 8-inch propeller quadrotor (which has a much larger propeller disk area). The overall efficiency of PULSAR is 17.5 and 4.1% more than that of the 7.5- and 8-inch propeller quadrotor, respectively.

Moreover, we compared PULSAR with three commercial quadrotor UAVs from DJI, including Mavic Air 2, Mavic 3, and Phantom 4 Pro V2.0. Because the standard PULSAR has a different weight and propeller disk area from the commercial quadrotors, to make a fair comparison, we developed two variants of PULSAR: The first one, denoted as "PULSAR (no LiDAR)," removes the LiDAR payload, and the second one, denoted as "PULSAR (no LiDAR, 16.4″ propeller)," additionally scales up the propeller to 16.4 inches (see fig. S13).

The comparison results are shown in Fig. 4 and table S1. As can be seen, PULSAR (no LiDAR, 16.4″ propeller) achieved the longest hover time among all UAVs under comparison and outperformed its counterpart DJI Mavic 3 in terms of flight time (40 min 27 s versus 40 min) and efficiency (8.21 g/W versus 7.75 g/W), despite having disadvantages in every aspect, including a battery with less energy (73.26 Wh versus 77 Wh), less total disk area (1362.8 cm^2 versus 1790.9 cm^2), and more total weight (929 g versus 895 g). Next, comparing PULSAR (no LiDAR) with DJI Mavic Air 2, although the former has a considerably larger weight (750 g versus 570 g), it still achieved higher power efficiency (8.20 g/W versus 7.76 g/W). More comparison analyses are supplied in the Supplementary Materials.

Trajectory tracking in indoor environment

To evaluate power efficiency in dynamic flight conditions as well as maneuverability, we conducted trajectory tracking experiments on both PULSAR and the built quadrotor in an indoor environment. In the experiment, the trajectory command was set to a figure "8" path that was planned offboard (trajectory commands switched to T3 in Fig. 3A). To fully test the UAV dynamics, as shown in Fig. 5, we expanded the figure "8" path in all three directions (3.6 m in x, 1.2 m in y, and 1.0 m in z). The trajectory period T was adjusted to change the flight speed (i.e., T = 8, 6, or 5 s): A smaller period leads to a higher average speed. Each trajectory was executed for five cycles to characterize the average power consumption and tracking accuracy. To make a fair comparison, both PULSAR and the quadrotor used the same LiDAR-inertial odometry method to provide the state feedback and the same cascaded proportional-integral-derivative (PID) controller structure with velocity and acceleration feedforward to track the trajectory (see details in Materials and Methods). The controller parameters of the two UAVs were tuned to our best efforts.

Figure 5A shows the tracking performances of PULSAR. Overall, PULSAR tracked these 3D trajectories tightly, with errors below a small bound. At a period T = 8 s (maximal speed, 1.93 m/s), the tracked trajectory matched with the reference trajectory closely, indicating a good tracking performance at low speeds. The tracking performance degraded with the increase in flight speed. At a period T = 5 s (maximal speed, 3.01 m/s), the tracked trajectories exhibited obvious deviation from the reference. This performance degradation with flight speed is a typical phenomenon for multicopter UAVs due to the increased disturbances, such as blade flapping (*37*). Tracking high-speed trajectories also requires fast actuator responses that are limited by the motor delay. Figure 5 (B and C) shows the quantitative comparison of PULSAR and the quadrotor.

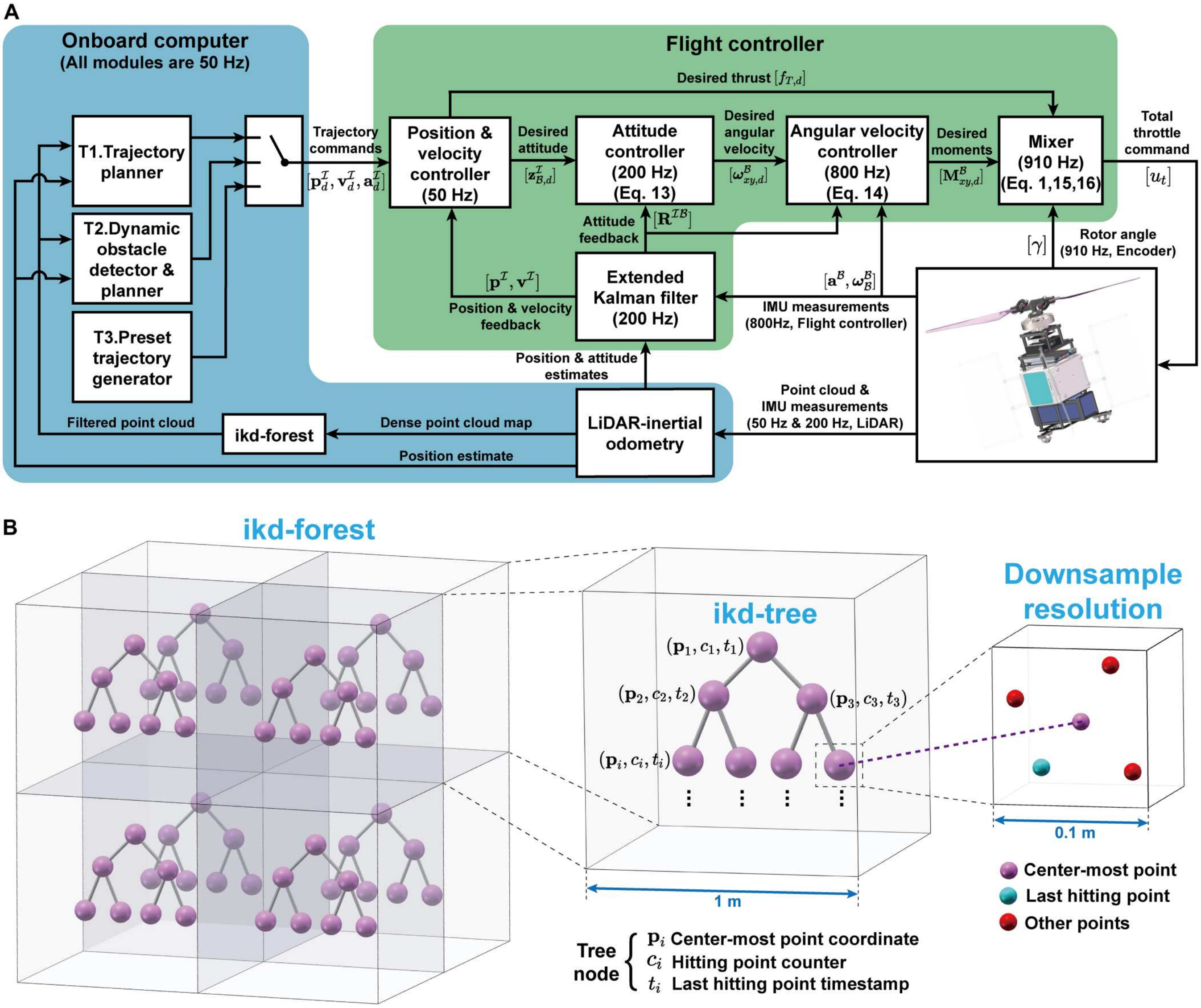


Fig. 3. Software framework. (**A**) Overall software framework. The trajectory commands were switched among different sources according to the application. Details of the position and velocity controller are shown in fig. S12. (**B**) Map used by trajectory planners to represent static and dynamic obstacles in environments. The map is named "ikd-forest," which cuts the space into 1-m voxels, each containing an incremental k-d tree data structure, ikd-tree (*57*). The ikd-tree downsamples all points contained in the voxel with 0.1-m spatial resolution by retaining the centermost point on the tree. Besides the point coordinate $\mathbf{p}_i$, each node of the *k*-dimensional tree saves the total number of the hitting point counter c_i and the last hitting point timestamp t_i.

The average power consumption of PULSAR was around 186 W, whereas those of the two benchmarked quadrotors are about 255 and 226 W for 7.5- and 8-inch propellers, respectively, suggesting better efficiency of PULSAR. The tracking accuracy was assessed by the absolute position error, which is the norm of the position error between the reference trajectory and the tracked one. The results suggest that PULSAR achieved slightly larger, yet acceptable, tracking errors than the quadrotor. Figure 5E shows the 3D maps built by the two UAVs during the flights. Benefiting from the extended FoV via self-rotation, PULSAR had a more complete mapping of the environment, whereas the quadrotor only mapped a small portion because of the constant yaw angle during the flight. The process of PULSAR tracking a trajectory with a period of 5 s is shown in movie S3.

Response to position commands

To verify the agility of PULSAR, we tested it to respond a step position command (movie S4). In the experiment, the trajectory command was set to a step signal in the *x* direction (trajectory commands switched to T3 in Fig. 3A). Figure 6A shows the step response. The response in the *x* direction suggests a rise time of 0.91 s and a settling time of 2.21 s without any noticeable steady-state error. In responding to the command in *x*, the position in *y* was also perturbed because of the self-rotation–induced gyroscopic effect, leading to a maximum position error of 0.173 m. Similarly, the tilting of thrust to respond to the command in *x* also caused a temporary altitude drop, leading to a 0.101-m maximum position error in the *z* direction. Figure 6 (B and C) shows the attitude and angular velocity responses, which demonstrate tight tracking of

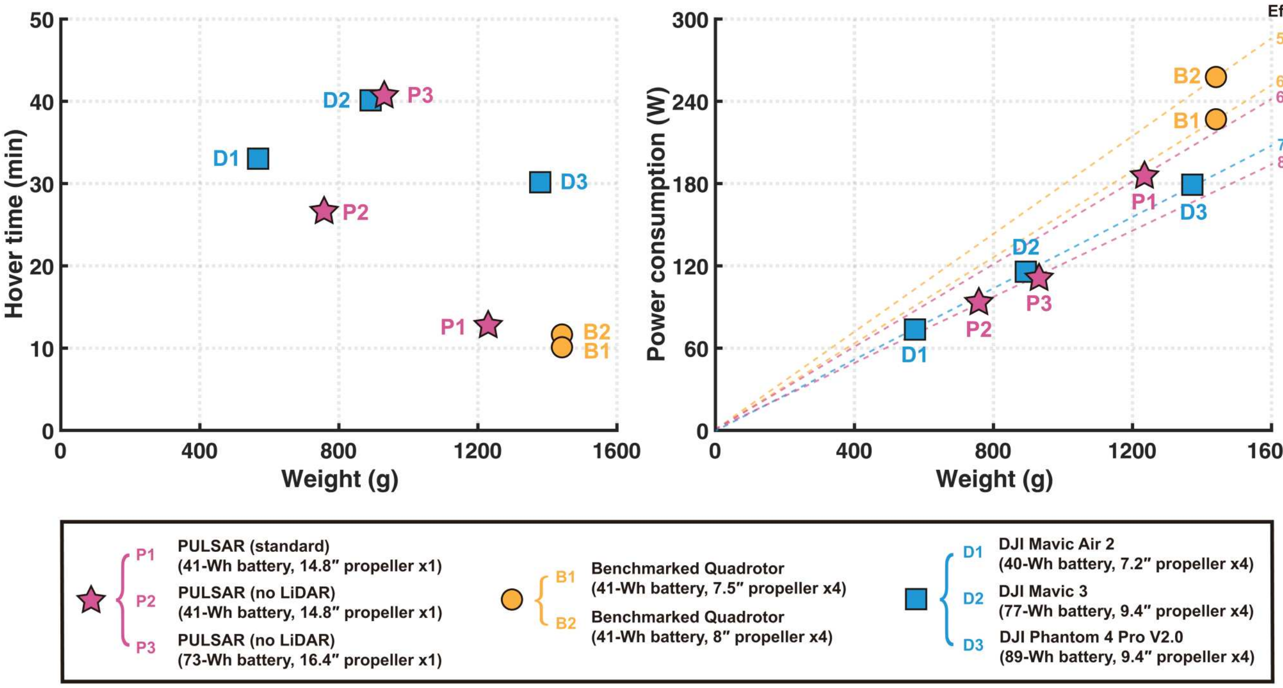


Fig. 4. Hover time, power consumption, and efficiency comparisons of PULSAR, the benchmarked quadrotors, and the commercial UAVs. The data of PULSAR and the benchmarked quadrotor are from experiments, and the data of the commercial UAVs are from DJI official specifications.

pitch and roll commands along with the uncontrolled yaw angle and rate, a natural result of the self-rotation design of PULSAR. Figure 6D shows the three control actions controlling pitch, roll, and altitude, respectively. These three actions were synthesized into a single command (Fig. 6D, iv) for the motor to execute.

Robustness to external disturbances

When operating in outdoor environments, UAVs are usually susceptible to certain disturbances, such as wind gusts. To validate PULSAR's robustness to such external disturbances, we used a fan to produce a wind gust and measured the displacement of PULSAR from its initial hovering position (movie S5). Figure 7 shows the experimental setup and results. In the beginning, PULSAR was hovering at a height of 1 m and a position of 0.7 m in front of the fan (trajectory commands switched to T3 in Fig. 3A). The fan at this distance created a wind gust with speeds up to 4.5 m/s for PULSAR. As the wind was applied, PULSAR was pushed away from the hovering position by a maximum of 15.6 cm. The position error caused the flight controller to estimate and to compensate for wind disturbance, bringing PULSAR back to its original position after 11.5 s. Similarly, when the fan was off, the redundant compensation action led to a displacement up to 18.0 cm in the opposite direction. This position error caused the flight controller to adjust the compensation action and, after 7.2 s, brought PULSAR back to the original position. During the response to wind applied on the x direction, PULSAR's position in y and z was slightly perturbed because of the coupling effect, causing a maximum position error of 8.2 cm for y and 4.1 cm for z. Moreover, the pitch and roll responses tightly tracked their respective desired values computed by the outer-loop position controllers, whereas the yaw was freely rotating. This experiment suggests that PULSAR is able to maintain its stability and hovering position in the presence of external disturbances, making it suitable for operation in real-world environments.

Autonomous navigation in unknown, GNSS-denied environments

To verify the full autonomous navigation ability of PULSAR, we performed a waypoint navigation experiment in a wooded environment of 54 m by 26 m (Figs. 1D and 8A). In the experiment, the trajectory command was computed in real time by the onboard trajectory planner (trajectory commands switched to T1 in Fig. 3A), which had no prior knowledge of the environment except for eight waypoints scattered in the area (star points in Fig. 8A). As the flight proceeded, the trajectory planner automatically generated a smooth trajectory (orange path in Fig. 8A) from the UAV current position to the next waypoint, without colliding with any obstacles (e.g., trunks, branches, and tree leaves). The planned trajectory was then tracked in real time by the onboard controller with small errors (blue path in Fig. 8A). The total flight time was about 125.7 s over the 63-m trajectory, during which PULSAR flew fully autonomously without any human intervention or piloting. Moreover, a 3D point cloud map of the environment was obtained during the flight. Benefiting from the extended FoV of PULSAR, the built map had points uniformly distributed in all horizontal directions, instead of all lying within the small conical sensor FoV, leading to a more efficient exploration of the environment. We conducted the experiment successfully in the same wooded environment in both day and night (see Fig. 1, C and D, and movie S6) and in other environments, such as a cave (movie S7). The various successful flights suggest that PULSAR has a robust navigation ability in unknown, GNSS-denied environments.

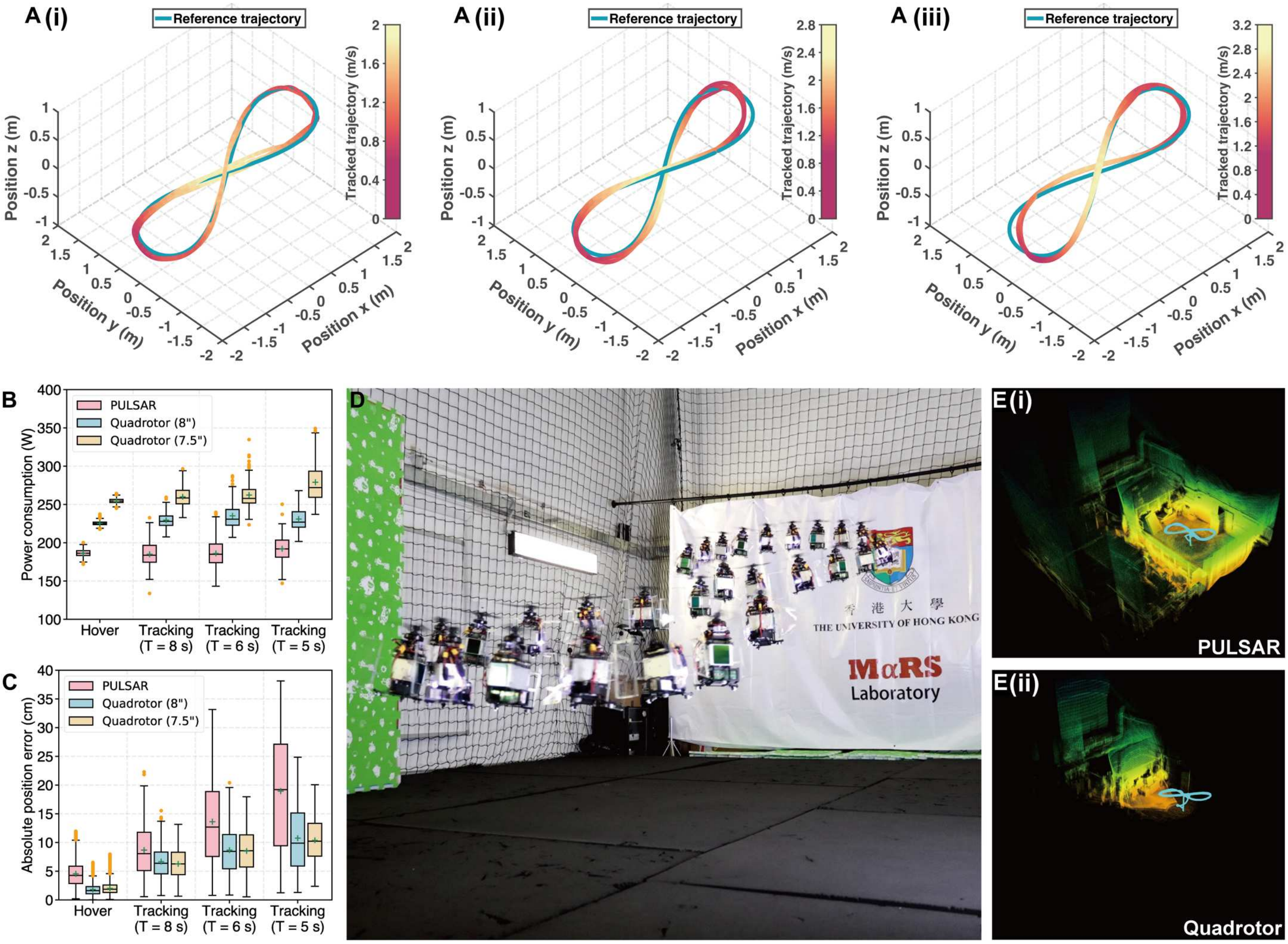


Fig. 5. Indoor trajectory tracking. (**A**) The trajectory under tracking is a figure "8" path with different periods, *T*. The green line represents the reference trajectory, whereas the line colored by velocity represents the actual flight path. Each trajectory is tracked for five cycles, where the middle three are displayed for better visualization of the speed. A(i), A(ii), and A(iii) are the trajectories with period *T* of 8, 6, and 5 s, respectively. (**B**) Power consumption during hover and trajectory tracking. The hover power is filtered by a moving average filter of window size 6. (**C**) Absolute position error of hover and trajectory tracking. On each box plot of (B) and (C), the central mark indicates the median, and the green "+" symbol indicates the mean. The bottom and top edges of the box represent the 25th and 75th percentiles, respectively. The whiskers extend to the maximum and minimum values excluding the outliers, which are plotted individually using the orange points. (**D**) Overlaid snapshots of PULSAR when it is tracking the trajectory with period *T* = 8 s. (**E**) 3D maps built by the two types of UAVs during trajectory tracking.

Figure 8 (B and C) shows the behavior of PULSAR in response to balls from two different directions. In both cases, the images on the left are overlaid snapshots of PULSAR and the ball captured at successive moments. PULSAR successfully detected the ball once it faced the ball due to self-rotation. Then, it moved to one side to avoid the ball while maintaining a safe margin with other static structures (e.g., trees). The images on the right are third-person views from the global point cloud maps built by PULSAR, including the detected ball position (gray ball markers), ball trajectory (yellow arrow), point measurements in the current LiDAR FoV, and the complete UAV trajectory (blue path) from takeoff to hovering and then to avoid the ball. These results imply that PULSAR is able to perform agile motions and to perceive the environment (both dynamic obstacles and static structures) in all horizontal directions beyond the sensor's original FoV.

The computation time of the navigation modules running on the onboard ARM processor is shown in Fig. 8D. The average computation times of the LiDAR-inertial odometry and the trajectory planner were 9.48 and 8.42 ms, respectively, which were well below the 20-ms period (i.e., 50 Hz) and suggested a real-time performance of the navigation module. Note that the trajectory planner and the odometry ran in parallel (see Materials and Methods), so a real-time performance would only require each individual module to take no more than 20 ms.

Avoidance of dynamic obstacles from different directions

To demonstrate the advantage brought by the extended FoV through self-rotation, we tested PULSAR's ability to avoid dynamic obstacles in an outdoor environment (movie S8). We threw a ball from two orthogonal directions that could not be detected if PULSAR did not have self-rotation due to the 70° conical

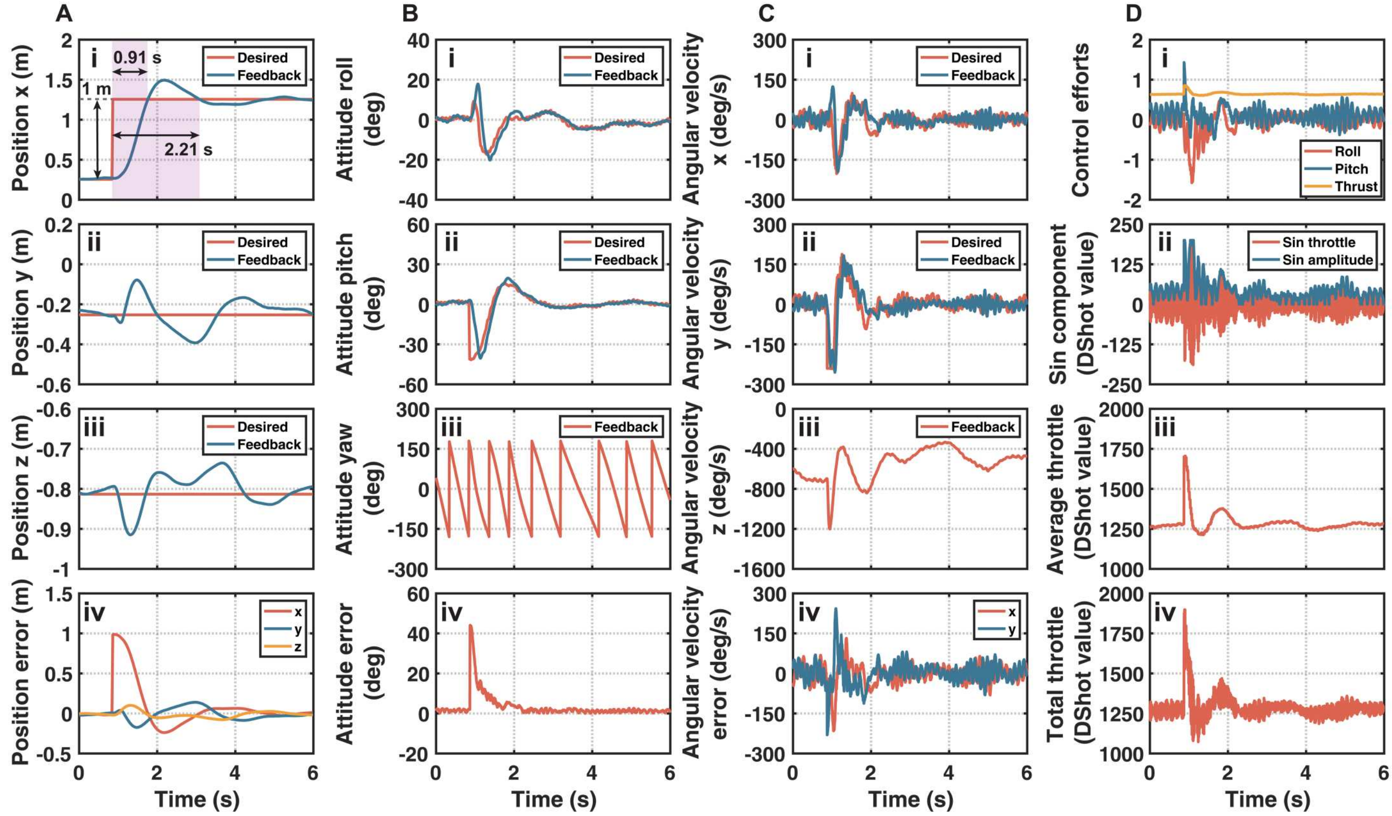


Fig. 6. Response of PULSAR to a step position command in the *x* direction. (**A**) Position response: (i to iii) Position responses in *x*, *y*, and *z*, respectively, and (iv) the position errors in *x*, *y*, and *z*. (**B**) Attitude response: (i and ii) Responses in the UAV roll and pitch angles, (iii) the UAV yaw angle, and (iv) attitude errors defined as the intersection angle between $\mathbf{z}^{\mathcal{I}}_{\mathcal{B}}$ and $\mathbf{z}^{\mathcal{I}}_{\mathcal{B},d}$, which excludes the yaw angle difference. (**C**) Angular velocity responses: (i and ii) Angular velocity response in *x* and *y* represented in inertial frame $\mathcal{I}$, respectively, and (iii) angular velocity in *z* represented in body frame $\mathcal{B}$, which indicates the self-rotation of the UAV body. (**D**) Control actions: (i) The desired moment $\mathbf{M}^{\mathcal{B}}_{d}$ (see Eq. 14, which has components in roll and pitch only) and the thrust $f_{T,d}$ (normalized by the maximal thrust $f_{T,\max}$; see Eq. 11). (ii) The amplitude of the sinusoidal throttle u_s and the complete sinusoidal throttle modulated on the rotor angular position ϕ, $u_s \sin(\phi - \alpha)$. (iii) The average throttle u_a and (iv) the total throttle u_t output to the ESC. Note that this is the only actuator command.

sensor FoV. The ball had a diameter of 32 cm, a size similar to PULSAR, and approached at a speed of around 5.8 m/s. We threw the ball such that it would have hit PULSAR if no evasive maneuver was taken. Consequently, to avoid this collision, PULSAR must detect the incoming ball along with static obstacles in the environment and generate a safe target position for execution (trajectory commands switched to T2 in Fig. 3A).

We conducted the experiment multiple times, and the vehicle managed to detect and avoid the ball most of the time. The major failure cases were caused by short triggering distances, which is the distance of the ball that triggers PULSAR to execute an evasive maneuver. When a ball arrived at the triggering distance, it might just miss the LiDAR FoV and hit the UAV body before it could be detected in the next revolution (after 0.3 s due to ~2.7-Hz self-rotation rate). In practice, this drawback could be trivially overcome by increasing the triggering distance. In the experiment, the triggering distance was set to 4 m because the ball cannot travel a longer distance in the air due to gravity.

The computation time of the navigation modules is shown in Fig. 8E. The average computation times of the odometry and dynamic obstacle detector and planner were 11.35 and 1.15 ms, respectively, which lead to a small detection latency and reliable 50-Hz real-time running.

DISCUSSION

PULSAR's key features

PULSAR has a unique design compared with existing single-actuator UAVs. The dSAW proposed in (*28*) used a servo to drive the cyclic pitch of a wing flap during an air-induced self-rotation. Because of the lack of propellers, the dSAW cannot perform powered flight and only works by dropping from a preset altitude. The single-motor designs (*29*–*33*) are most similar to PULSAR in terms of the number of actuators. Among them, Piccoli and Yim (*29*) used the motor to provide the lift only, leading to uncontrolled attitude and lateral motion. The designs (*30*–*33*) achieved full 3D position control using only one motor, similar to PULSAR, but their flying and actuation principles were completely different. Specifically, Piccoli and Yim (*30*) and Zhang *et al.* (*31*) installed the motor on one side of the UAV frame, a position that is displaced from the center of mass (CoM). Then, with a normal propeller attached, the noncentral motor at rotation provided a thrust that, on the one hand, lifted the UAV's altitude and, on the other hand,

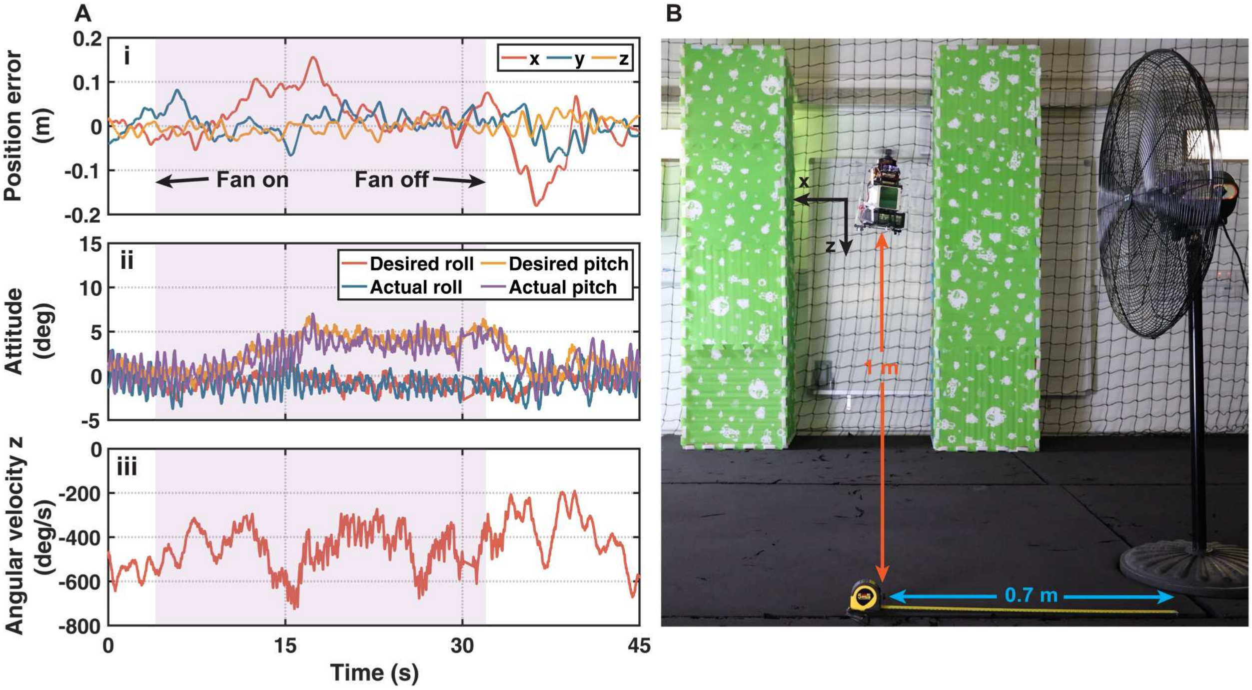


Fig. 7. Wind disturbance rejection. (**A**) The position error (i), attitude (pitch and roll) response (ii), and angular velocity in the *z* axis (iii). The position error (i) is the difference between the trajectory command (which is the specified hovering position) and the position feedback. The angular velocity *z* (iii) indicates PULSAR's self-rotation. (**B**) Experiment setup. The UAV is hovering 0.7 m away from the fan and at a height of 1 m. The fan creates a wind gust of 4.5 m/s at the hovering position of PULSAR.

produced a pitch torque that inclined the thrust from the vertical direction. Meanwhile, the counter-torque of the motor caused the UAV, hence the inclined thrust, to spin. In the case of hovering, the motor speed was kept constant; then, the inclined thrust at spinning will contribute to a net force along the vertical direction only. To move the UAV along a horizontal direction, the motor speed was increased at the moment the thrust was inclined to that direction, hence actuating the lateral motion once every revolution of the UAV self-rotation. On the other hand, Win *et al.* (*32*, *33*) installed the motor on a one-side wing of the UAV. The motor provided a thrust driving the wing to rotate, which produced an aerodynamic lift that controlled the UAV's altitude. To move the UAV along a horizontal direction, the motor speed (hence thrust) was increased or decreased at the respective location, which drove the wing to decelerate (lower lift) or accelerate (higher lift). The differential aerodynamic lift led to a moment that actuated the UAV pitch or roll, hence controlling the UAV lateral motion once every revolution of the UAV self-rotation. In contrast to (*30*–*33*), the lateral motion of PULSAR is actuated at every revolution of the propeller rotation, which is at a much higher rate than the body self-rotation (4700 rpm of the propeller versus 160 rpm of the body). The increased actuation rate enables more accurate trajectory tracking, rapid step response, and robust disturbance rejection for PULSAR, which were not demonstrated previously (*30*–*33*).

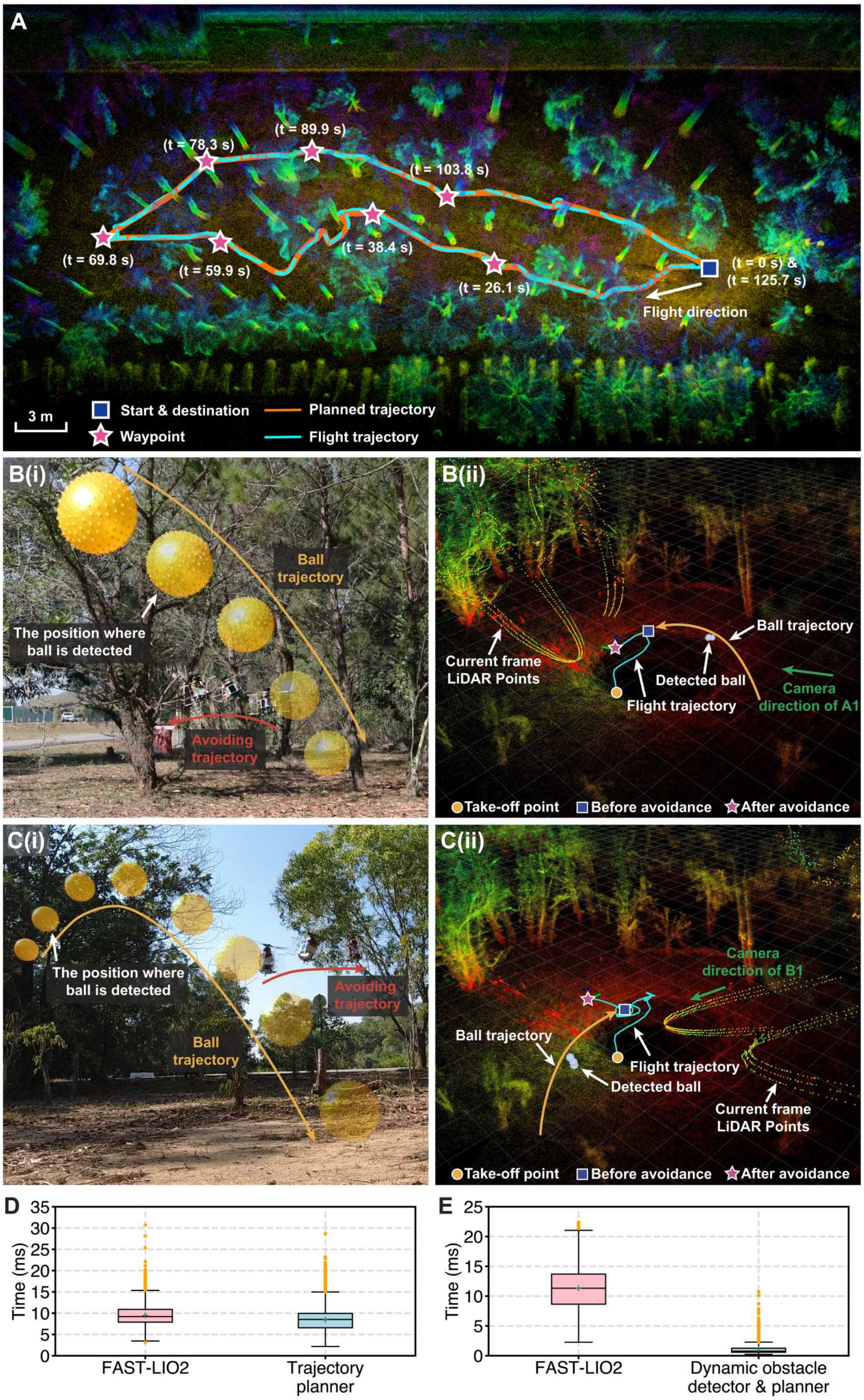
PULSAR is also a self-rotating UAV that can navigate autonomously in unknown environments. By leveraging the high-rate LiDAR point measurements, PULSAR can robustly estimate its full state in the presence of fast FoV changes, requiring no external instrumentation. With the LiDAR sensor, PULSAR is also able to build a 3D point cloud map of the environment, from which both static and dynamic obstacles can be detected and avoided in real time. Consequently, PULSAR can safely navigate in a variety of GNSS-denied environments (e.g., woods, caves, and tunnels), an ability that was rarely demonstrated in existing self-rotating UAVs.

The swashplateless mechanism used by PULSAR is not the first time that it was used on a micro UAV. Previous works (*35*, *36*, *38*) used the swashplateless mechanism on coaxial-rotor UAVs (CRUAVs) with two propellers. Besides the apparent differences in the self-rotation ability and number of actuators, which were the primary motivation of this work, PULSAR also differs from these works in many implementation details. One major improvement in PULSAR is the decreased friction in the swashplateless mechanism, eliminating the deadband phenomenon (i.e., no moment output when the input command is lower than a threshold) in CRUAVs. Another improvement is the simplified motor controller implementation. CRUAVs used a dedicated motor ESC to drive the swashplateless mechanism, whereas, in PULSAR, the driving strategy is implemented on the flight controller without any such ESC. All electronics in PULSAR, including the ESC, flight controller, LiDAR sensor, and onboard computer, are available off-the-shelf, making the implementation much easier.

Energy efficiency

PULSAR has a greater energy efficiency when compared with the benchmarked quadrotors. According to the momentum theory (*39*, *40*), a UAV has its ideal hover efficiency $\eta = \sqrt{2\rho A}/\sqrt{mg}$ (g/W), where m, A, ρ, and g are total mass, total propeller disk area, air density, and the gravity acceleration, respectively. For a quadrotor

Fig. 8. Outdoor experiments with LiDAR sensor. (**A**) Autonomous waypoint navigation of PULSAR in an outdoor wood environment. The start point and the destination point are the same. The time shown on each waypoint is the time that PULSAR arrives at that waypoint. During the flight, a 3D point cloud map of the environment is simultaneously built. (**B** and **C**) Dynamic ball avoidance with PULSAR. The ball was thrown from two orthogonal directions whose avoiding processes are shown in (B) and (C), respectively. In both cases, overlaid snapshots on the left show the ball trajectory, the position of the ball when it is detected by PULSAR, and the avoiding trajectory executed by PULSAR. Images on the right show third-person views of the environment map, current LiDAR point measurements, the detected ball position and its trajectory, and the complete UAV trajectory before and after avoiding the ball. Poses of the camera capturing snapshots on the left are indicated by the green arrows. The environment map is simultaneously built by the LiDAR-inertial odometry system running on the onboard computer. (**D**) Computation times of the odometry and trajectory planner in the autonomous waypoint navigation experiment. (**E**) Computation times of the odometry and dynamic obstacle detector and planner in the dynamic obstacle avoidance experiment.



that preserves the same total disk area and weight as PULSAR, it should have the same power consumption and efficiency. However, in practice, four small propellers often have around 5.79 to 13.61% lower efficiency than one big propeller (fig. S6), due to the different Reynolds number and propeller geometry that are not accounted for by momentum theory (*39*). Then, installing the four propellers on the quadrotor airframe would further introduce rotor-to-rotor interactions (*41*–*43*) and rotor-to-body interactions (*44*), which would cause a further efficiency drop (for the 7.5-inch propeller quadrotor, the drop is about 5.99%). In contrast, PULSAR eliminates this rotor-to-rotor interaction and mitigates the rotor-to-body interaction by placing the UAV body under the propeller hub, leading to an efficiency drop of merely 2.88%. Next, a quadrotor UAV has more component weight (due to the higher number of propulsion systems) and structure weight (due to the distributed motor location on four arms), which bring more power consumption and efficiency drop. In our case, the 7.5-inch propeller quadrotor has 206 g more weight than PULSAR (table S3), which leads to a 4.35% efficiency drop. Considering all the above factors and the power of onboard avionic devices (table S4), PULSAR has an overall efficiency that is 17.5% higher than the 7.5-inch propeller quadrotor (which has a similar total disk area) and 4.07% higher than the 8-inch propeller quadrotor (which has a larger total disk area), respectively. More detailed analyses and comparison results are in the Supplementary Materials.

Besides the custom quadrotors, PULSAR also exhibits efficiency advantages over commercial quadrotor UAVs with smaller weights and larger sizes. The smaller weights and larger sizes already place commercial quadrotor UAVs in an advantageous position, but PULSAR still achieved a higher efficiency (see the Supplementary Materials for more details).

Mapping efficiency

The mapping efficiency of PULSAR is mainly attributed to two factors: First, the increased energy efficiency enables longer-time tasks, avoiding frequent returns and recharges (*6*, *45*); second, the extended FoV via self-rotation measures more volumes per unit time by naturally distributing the sensor measurements equally around the UAV without any active planning on the yaw angle. To quantify the FoV extension by self-rotation, we used the concept of solid angle (fig. S10). The current LiDAR on PULSAR (Livox AVIA) has a conical FoV of about 70° in both vertical and horizontal directions, resulting in an approximate solid angle of 1.40 sr at static. By self-rotation, the solid angle reaches 7.21 sr (57.4% of omnidirection), which is 5.15 times higher. In the case of a 360° LiDAR (e.g., Velodyne Puck LITE), the vertical FoV is ±15°, leading to a solid angle of 3.25 sr at static. If the LiDAR is installed on PULSAR vertically, considering some area that is blocked by the UAV structure, an FoV with a solid angle of 10.66 sr (84.8% of omnidirection) can be achieved via the self-rotation, which is 3.28 times higher than the original FoV. The results quantified the extension of FoV through self-rotation. In the current implementation of PULSAR, the Livox AVIA LiDAR was used because of its long sensing distance and small weight (498 g).

Agility

PULSAR achieved a good level of agility, as demonstrated in the experiments of trajectory tracking and step position response. This enabled PULSAR to perform challenging tasks like avoiding high-speed dynamic obstacles, which are possible only in quadrotor UAVs (*46*, *47*) to date. For existing self-rotation UAVs (*20*–*33*) and other underactuated UAVs with fewer than four propellers (*40*, *48*–*52*), the main focuses were on design concept, vehicle configuration, and flying feasibility, whereas agility and the ability to avoid dynamic obstacles were not considered or demonstrated. Only the dSAW proposed in (*28*) demonstrated an agile motion that involved only unpowered diving.

Although it has a good level of agility, PULSAR has a thrust-to-weight ratio that is obviously lower than those of quadrotors. For the two benchmarked quadrotors, their four propellers can provide much greater total thrust at the cost of high power consumption, leading to a thrust-to-weight ratio of up to 2.7 (7.5-inch propellers) and 3.5 (8-inch propellers), which are 1.29 and 1.67 times higher than that of PULSAR (2.1 thrust-to-weight ratio), respectively. The low thrust-to-weight ratio has prevented PULSAR from executing extremely agile maneuvers, such as flips and racing. Further increasing the thrust-to-weight ratio is possible by increasing the propeller size or decreasing the weight; nevertheless, the former would prevent the UAV from flying in tight spaces, whereas the latter requires more lightweight sensors suitable for navigation at high rotation rates.

Scalability

PULSAR can be adapted to different scales because of the good scalability of the swashplateless mechanism, which has been adequately verified by experiments conducted on prototypes with diameters from 0.1 to 1 m (*53*). Other parts of PULSAR, including the body structure, vanes, landing gears, and battery, are either manufactured with common materials and fabrication techniques or commercial off-the-shelf components, which all scale well. The scalability of PULSAR along with its symmetric geometry allows for carrying various payloads of different weights as required by the task. For instance, by adopting motion blur correction (*20*) or ego-motion compensation (*54*), a mini-size high-speed RGB camera or event camera can be installed for a smaller version of PULSAR, whereas larger-scale equipment such as radar, zoom camera, or LiDAR camera system can be carried by a bigger version.

Potential applications

The high energy efficiency, autonomous navigation ability, and extended sensor FoV of PULSAR make it very suitable for exploration tasks, such as environment surveying, search and rescue, disaster relief, terrain mapping, and automatic 3D reconstruction. Environments in these tasks are often unknown and GNSS-denied (e.g., tree canopy, cave, tunnel, and post-disaster buildings) and involve both static structures and dynamic objects (e.g., birds and animals). As demonstrated in the experiments, PULSAR can operate safely in these real-world environments and acquire environmental data rapidly, both day and night, giving timely and reliable feedback for decision-making.

MATERIALS AND METHODS

Working principle of the swashplateless mechanism

The working principle of the swashplateless mechanism is illustrated in Fig. 2 (E and F) and movie S1. Because of the two tilted passive hinges, the blade pitch can be changed cyclically by modulating an acceleration impulse to the motor at each rotor revolution. When the impulse is applied, the rotor starts accelerating from its present position (i.e., angle α) to a new position (i.e., angle $\alpha + \lambda_0$). Because of the blade's inertia, its rotation will lag from the rotor during the rotor acceleration by rotating along the hinge, leading to a lag angle δ. Then, because of the 45°-tilted and asymmetric hinge directions, the lag angle causes the positive blade (the red one) to twist below the propeller disk, leading to an increased pitch angle, whereas the other blade (the blue one) twists above the disk, leading to a decreased pitch angle (Fig. 2F). The differential changes of the blade pitch angles then cause unequal thrusts on the two blades, which lastly produce a net momentary moment perpendicular to the blades' feathering axes (the purple arrow in Fig. 2E). Similarly, when the rotor is at an opposite position, a deceleration impulse is applied; the decelerating rotor then causes a decreased pitch angle on the positive blade and an increased pitch angle on the negative blade. The net effect is another momentary moment that has the same direction as the former one. Consequently, during one rotor revolution, two momentary moments with the same orientation can be produced to actuate the UAV attitude. The summation of the two moments remains within the propeller disk, with its orientation angle β determined by the rotor angle where the acceleration starts (i.e., angle α) and its magnitude determined by the magnitude of the impulse.

Limited by the rotor inertia, impulse acceleration and deceleration cannot be achieved in practice. Instead, a smoother sinusoidal motor speed profile is adopted. To produce a sinusoidal motor speed, the throttle command u_t is designed as

$$u_t = u_a + u_s \sin(\varphi - \alpha) \tag{1}$$

where u_a is the average throttle to maintain an average motor speed, u_s is the amplitude of the sinusoidal throttle, and ϕ is the current rotor angle measured by the magnetic encoder.

In Eq. 1, the average throttle u_a and the amplitude of sinusoidal throttle u_s determine the propeller thrust and moment, respectively, as below:

$$f_T = k_a u_a, \quad M_C = k_s u_s \tag{2}$$

where f_T is the propeller thrust and M_C is the magnitude of the moment. The linear relations in Eq. 2 hold at the hovering condition with roughly constant coefficients k_a and k_s determined from the data of the swashplateless mechanism in the Supplementary Materials (fig. S2).

Last, the rotor acceleration position α in Eq. 1 determines the orientation angle of the moment (Fig. 2E), as below:

$$\beta = \alpha + \lambda_0 - \delta + \pi/2 \approx \alpha + \lambda_0 + \pi/2 \tag{3}$$

In practice, the lag angle δ is very small and can be deemed as zero, and the angle λ_0 is constant and calibrated in advance.

Dynamic modeling

PULSAR consists of two rigid parts: One part consists of the UAV body and motor stator (referred to as the body part), and the other consists of the propeller and motor rotor (referred to as the rotor part). To model the UAV dynamics, we defined two coordinate frames (fig. S11): The first one is an inertial (ground-fixed) coordinate frame $\mathcal{I}$ whose z axis is the same as the direction of the gravity vector, and the second one is a body-fixed coordinate frame $\mathcal{B}$ attached to the body part with origin at the CoM of the whole UAV and aligned with the principal axis of inertia. Let $\boldsymbol{\omega}^{\mathcal{B}} = [\omega_x^{\mathcal{B}}, \omega_y^{\mathcal{B}}, \omega_z^{\mathcal{B}}]^T$ be the body's angular velocity. Then, the angular momentum $\mathbf{L}^{\mathcal{I}}$ of PULSAR observed in the inertial frame $\mathcal{I}$ is

$$\mathbf{L}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{B}}(\mathbf{I}_O^{\mathcal{B}}\boldsymbol{\omega}^{\mathcal{B}} + \mathbf{I}_R^{\mathcal{B}}\boldsymbol{\Omega}^{\mathcal{B}}) \tag{4}$$

where $\mathbf{R}^{\mathcal{I}\mathcal{B}}$ is the rotation matrix and its superscript indicates the coordinate transformation from $\mathcal{B}$ to $\mathcal{I}$; $\boldsymbol{\Omega}^{\mathcal{B}} = [0,0,\sigma]^T$, with σ being the rotor speed with respect to the stator; $\mathbf{I}_O^{\mathcal{B}}$ is the inertia matrix of the whole UAV including both rotor and body parts; and $\mathbf{I}_R^{\mathcal{B}}$ is the inertia matrix of the rotor part only, where the reference points O and R are the CoM of the whole UAV and the rotor part, respectively (fig. S11).

According to the angular momentum theorem, the derivative of $\mathbf{L}^{\mathcal{I}}$ is equal to the exerted moment $\mathbf{M}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{B}}\mathbf{M}^{\mathcal{B}}$, which implies

$$\mathbf{M}^{\mathcal{B}} = \lfloor \boldsymbol{\omega}^{\mathcal{B}} \rfloor (\mathbf{I}_O^{\mathcal{B}}\boldsymbol{\omega}^{\mathcal{B}} + \mathbf{I}_R^{\mathcal{B}}\boldsymbol{\Omega}^{\mathcal{B}}) + (\mathbf{I}_O^{\mathcal{B}}\dot{\boldsymbol{\omega}}^{\mathcal{B}} + \mathbf{I}_R^{\mathcal{B}}\dot{\boldsymbol{\Omega}}^{\mathcal{B}}) \tag{5}$$

where $\mathbf{M}^{\mathcal{B}} = [M_x^{\mathcal{B}}, M_y^{\mathcal{B}}, M_{\text{prop}}^{\mathcal{B}} + M_{\text{body}}^{\mathcal{B}}]$, $M_{\text{prop}}^{\mathcal{B}}$, and $M_{\text{body}}^{\mathcal{B}}$ are the moments induced by air drag exerted to the rotor part and body part, respectively; $M_x^{\mathcal{B}}$ and $M_y^{\mathcal{B}}$ are the moments in the body x and y axis generated by the swashplateless mechanism, respectively; and $\lfloor \cdot \rfloor$ takes the elements of a vector to form a skew-symmetric matrix. According to the previous analysis,

$$M_x^{\mathcal{B}} = M_C \cos\beta, \quad M_y^{\mathcal{B}} = M_C \sin\beta \tag{6}$$

where M_C is the moment magnitude determined by Eq. 2 and β is the moment orientation angle determined by Eq. 3.

Because PULSAR is designed to rotate freely along its body z axis, we focus on the angular dynamics in the x and y axes only, obtaining (see detailed derivation in the Supplementary Materials)

$$\mathbf{M}_{xy}^{\mathcal{B}} = \mathbf{I}_O^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{xy}^{\mathcal{B}} - \lfloor \mathbf{L}_{\text{gyro}}^{\mathcal{B}} \rfloor \boldsymbol{\omega}_{xy}^{\mathcal{B}} \tag{7}$$

where $\mathbf{M}_{xy}^{\mathcal{B}} = [M_x^{\mathcal{B}}, M_y^{\mathcal{B}}, 0]^T$, $\boldsymbol{\omega}_{xy}^{\mathcal{B}} = [\omega_x^{\mathcal{B}}, \omega_y^{\mathcal{B}}, 0]^T$, and $\mathbf{L}_{\text{gyro}}^{\mathcal{B}} = [0, 0, (I_{O,z}^{\mathcal{B}} - I_{O,y}^{\mathcal{B}})\omega_z^{\mathcal{B}} + I_{R,z}^{\mathcal{B}}\sigma]^T$.

The kinematics model contains rotation motion and linear motion. Instead of describing the full rotation $\mathbf{R}^{\mathcal{I}\mathcal{B}}$, we only need to describe the kinematics of the z axis of the body frame $\mathcal{B}$ (i.e., $\mathbf{z}_{\mathcal{B}}^{\mathcal{I}}$) because the rotation along $\mathbf{z}_{\mathcal{B}}^{\mathcal{I}}$ is free, and only the rotation perpendicular to $\mathbf{z}_{\mathcal{B}}^{\mathcal{I}}$ will affect the UAV's linear motion. As proved in the Supplementary Materials, the rotation kinematics is

$$\dot{\mathbf{z}}_{\mathcal{B}}^{\mathcal{I}} = \boldsymbol{\omega}_{xy}^{\mathcal{I}} \times \mathbf{z}_{\mathcal{B}}^{\mathcal{I}} \tag{8}$$

where $\boldsymbol{\omega}_{xy}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{B}}\boldsymbol{\omega}_{xy}^{\mathcal{B}}$. Last, the linear motion is described as

$$\dot{\mathbf{v}}^{\mathcal{I}} = -\frac{f_T}{m}\mathbf{z}_{\mathcal{B}}^{\mathcal{I}} + \mathbf{g}^{\mathcal{I}} \tag{9}$$

$$\dot{\mathbf{p}}^{\mathcal{I}} = \mathbf{v}^{\mathcal{I}} \tag{10}$$

where m is the total mass of PULSAR, $\mathbf{g}^{\mathcal{I}}$ is the gravity vector, and f_T is the propeller thrust determined by Eq. 2.

Trajectory tracking control

The trajectory tracking control of PULSAR adopts a standard dual-loop cascaded controller structure (fig. S12), where the outer-loop position controller is a proportional controller and the inner-loop velocity controller is a PID controller. The acceleration feedforward of the trajectory is added, with the acceleration command computed by the velocity controller to produce the total desired acceleration $\mathbf{a}_d^{\mathcal{I}}$, which is then substituted into Eq. 9 to obtain the desired thrust $f_{T,d}$ and attitude $\mathbf{z}_{\mathcal{B},d}^{\mathcal{I}}$

$$\mathbf{f}_d^{\mathcal{I}} = m\ (\mathbf{a}_d^{\mathcal{I}} - \mathbf{g}^{\mathcal{I}}),\ f_{T,d} = \|\mathbf{f}_d^{\mathcal{I}}\|,\ \mathbf{z}_{\mathcal{B},d}^{\mathcal{I}} = -\frac{\mathbf{f}_d^{\mathcal{I}}}{f_{T,d}} \tag{11}$$

Note that we only have to specify the body z axis of the UAV without restricting its yaw angle, which is free rotating for PULSAR. To track the desired attitude $\mathbf{z}_{\mathcal{B},d}^{\mathcal{I}}$, an angular velocity of the body frame $\mathcal{B}$ that rotates $\mathbf{z}^{\mathcal{I}}_{\mathcal{B}}$ (the current attitude) toward $\mathbf{z}^{\mathcal{I}}_{\mathcal{B},d}$ should be generated. In PULSAR, the desired angular velocity $\boldsymbol{\omega}^{\mathcal{I}}_{xy,d}$ is designed as

$$\boldsymbol{\omega}^{\mathcal{I}}_{xy,d} = k^a_p(\mathbf{z}^{\mathcal{I}}_{\mathcal{B}} \times \mathbf{z}^{\mathcal{I}}_{\mathcal{B},d}) \tag{12}$$

where k^a_p is a proportional gain of the attitude controller and $\mathbf{z}^{\mathcal{I}}_{\mathcal{B}}$ is the body z axis estimated by the navigation module. As proved in the Supplementary Materials, when the desired angular velocity $\boldsymbol{\omega}^{\mathcal{I}}_{xy,d}$ computed in Eq. 12 is accurately tracked and the gain k^a_p is positive, $\mathbf{z}^{\mathcal{I}}_{\mathcal{B}}$ converges to the desired direction $\mathbf{z}^{\mathcal{I}}_{\mathcal{B},d}$, achieving the control objective.

To track the desired angular velocity $\boldsymbol{\omega}^{\mathcal{I}}_{xy,d}$ computed in Eq. 12, we transform it into the body frame $\mathcal{B}$ where the angular velocity dynamics is modeled by

$$\boldsymbol{\omega}^{\mathcal{B}}_{xy,d} = \left(\mathbf{R}^{\mathcal{I}\mathcal{B}}\right)^T \boldsymbol{\omega}^{\mathcal{I}}_{xy,d} = k^a_p\left(\mathbf{R}^{\mathcal{I}\mathcal{B}}\right)^T \lfloor \mathbf{z}^{\mathcal{I}}_{\mathcal{B}} \rfloor \mathbf{z}^{\mathcal{I}}_{\mathcal{B},d} = k^a_p \lfloor \mathbf{z}^{\mathcal{B}}_{\mathcal{B}} \rfloor \mathbf{z}^{\mathcal{B}}_{\mathcal{B},d} \tag{13}$$

which naturally leads $\boldsymbol{\omega}^{\mathcal{B}}_{xy,d}$ to have components only in x and y because $\mathbf{z}^{\mathcal{B}}_{\mathcal{B}} = [0, 0, 1]^T$. Then, a PI controller is used to track the desired angular velocity, leading to a control law

$$\mathbf{M}^{\mathcal{B}}_{xy,d} = \mathbf{K}^r_p \tilde{\boldsymbol{\omega}}^{\mathcal{B}}_{xy} + \mathbf{K}^r_i \int_0^t \tilde{\boldsymbol{\omega}}^{\mathcal{B}}_{xy} d\tau \tag{14}$$

where $\mathbf{M}^{\mathcal{B}}_{xy,d} = \left[M^{\mathcal{B}}_{x,d}, M^{\mathcal{B}}_{y,d}, 0\right]^T$ is the desired moment, $\tilde{\boldsymbol{\omega}}^{\mathcal{B}}_{xy} = \boldsymbol{\omega}^{\mathcal{B}}_{xy,d} - \boldsymbol{\omega}^{\mathcal{B}}_{xy}$ is the angular velocity error, both are vectors in the body frame $\mathcal{B}$, and $\mathbf{K}^r_p$ and $\mathbf{K}^r_i$ are two gain matrices that are tuned appropriately. Because both $\boldsymbol{\omega}^{\mathcal{B}}_{xy,d}$ and $\boldsymbol{\omega}^{\mathcal{B}}_{xy}$ have a zero z component, the computed moment $\mathbf{M}^{\mathcal{B}}_{xy,d}$ will naturally have a zero z component too. This moment, if executed successfully, will lead the angular velocity $\boldsymbol{\omega}^{\mathcal{B}}_{xy}$, which is subject to the first-order dynamics in Eq. 7, to converge to the desired value $\boldsymbol{\omega}^{\mathcal{B}}_{xy,d}$.

Mixer for driving swashplateless mechanism

The computed desired thrust $f_{T,d}$ (Eq. 11) and desired moment $\mathbf{M}^{\mathcal{B}}_{xy,d}$ (Eq. 14) are used to generate the total motor throttle command u_t in the form of Eq. 1. Specifically, replacing f_T with $f_{T,d}$ and M_C with $\|\mathbf{M}^{\mathcal{B}}_{xy,d}\|$ in Eq. 2, the average throttle and the amplitude of the sinusoidal throttle are

$$u_a = \frac{f_{T,d}}{k_a}, \quad u_s = \frac{\|\mathbf{M}^{\mathcal{B}}_{xy,d}\|}{k_s} \tag{15}$$

and the moment orientation angle is

$$\beta = \mathrm{atan2}(\mathrm{M}^{\mathcal{B}}_{\mathrm{y,d}}, \mathrm{M}^{\mathcal{B}}_{\mathrm{x,d}}) \tag{16}$$

which then determines the sinusoidal phase $\alpha = \beta - \lambda_0 - \pi/2$ according to Eq. 3. Last, the computed u_s, u_a, and α along with the current rotor angle ϕ measured by the magnetic encoder lead to the total throttle u_t, which is sent to the ESC for execution.

Software framework overview

As shown in Fig. 3A, the software framework is divided into two parts running on the respective hardware. The first part is the flight control, including the controllers, estimator, and mixer, which runs on the flight controller board at multiple frequencies: 50 Hz for position control, 200 Hz for attitude control, 800 Hz for angular velocity control, and 910 Hz for the mixer. This part has only one output: the total throttle command, which indicates the single-actuated characteristic of PULSAR. The second part is the navigation module, which consists of odometry and trajectory planning and runs on the onboard ARM computer at 50 Hz. The data flow from the onboard computer to the flight controller is the UAV states estimated by the LiDAR-inertial odometry and the trajectory commands generated by three means: The preset trajectory library contains trajectories planned offline, the trajectory planner plans a smooth and obstacle-free trajectory in real time according to the actual perception of the environment, and the dynamic obstacle detector and planner detects dynamic obstacles and generates an evasive trajectory. Different experiments will choose the trajectory command according to the task requirement.

The communication framework of software modules was ROS Noetic running in Ubuntu 20.04. All the software modules were implemented in C++. The flight control firmware of PULSAR was developed on the basis of PX4 V1.11.2. The communication between the onboard computer and the flight controller was based on MAVROS (*55*).

LiDAR-inertial odometry

The full-state estimation of PULSAR was realized by FAST-LIO2 (*56*), an efficient and robust LiDAR-inertial odometry framework. FAST-LIO2 estimates the UAV state (i.e., position, velocity, and attitude including the self-rotation) and updates the local map in the inertial (ground-fixed) frame (see the description in the "Dynamic modeling" section) whose heading is chosen as the initial UAV heading. We set the LiDAR scan rate to 50 Hz, which implies a 50-Hz state estimation and map update. Such a high-frequency map update enables FAST-LIO2 to tightly track even very aggressive motions, such as the self-rotation of PULSAR. The estimated state includes the UAV position, velocity, and attitude, which are fed to an extended Kalman filter (EKF) implemented on the flight controller. The EKF will further fuse the states with the onboard inertial measurement unit (IMU) to refine the UAV state at a higher frequency for the use of the controllers that are implemented on the same flight controller. Note that the state estimation of PULSAR's EKF is dependent on both the IMU and additional sensor (i.e., LiDAR or external motion capture system). Without the additional sensor, the estimation does not work properly. The details of the EKF are described in the Supplementary Materials.

Incremental *k*-dimensional forest

Besides odometry, another fundamental requirement for autonomous flight and dynamic obstacle avoidance is a map that represents both static and dynamic obstacles in the flying environments. We developed a map, called incremental *k*-dimensional forest (ikd-forest, see Fig. 3B), which is a collection of incremental *k*-dimensional trees (ikd-trees) (*57*), each contained in a voxel (cubic side length is 1 m). An ikd-tree first downsamples the points in the corresponding voxel by retaining only the centermost point according to a prescribed resolution (0.1 m). Then, the retained points are organized into a *k*-dimensional tree structure for efficient nearest neighbor search. An ikd-tree has the advantages of incremental updates (insert and delete) and dynamic rebalancing, which are all inherited by the ikd-forest. Furthermore, when compared with an ikd-tree that builds all points into a large tree, the ikd-forest achieves higher efficiency because the tree size of each ikd-tree in each voxel is reduced substantially.

On each node of an ikd-tree, it saves the point coordinates. To distinguish points on dynamic objects from those on static ones, two extra temporal characteristics are also saved on the node: the hitting point counter and the last hitting point timestamp. The hitting points counter records the number of points that are too close (i.e., within the resolution 0.1 m) to the point on the node (i.e., the centermost point), although these non-centermost points have been removed by the downsampling. The last hitting point timestamp denotes the timestamp of the last point hitting the node.

Trajectory generation and autonomous flight

To generate a safe trajectory in an environment with obstacles, we deployed a path planning module on the onboard computer, running at 50 Hz by receiving the point clouds measured by the onboard LiDAR. The path planning module was implemented on the basis of a time-accumulated local map and a kinodynamic A* search algorithm (*58*). To achieve a faster nearest neighbor search for obstacle avoidance in path planning, the ikd-forest was used to maintain the time-accumulated local map that only considers points that appeared in recent scans (indicated by the last hitting point timestamp). Furthermore, a "receding planning horizon" strategy (*59*) was used where the planner's starting position was taken from the current trajectory that is 20 ms after the current time. The "receding planning horizon" strategy removes the dependence of the planner on the odometry so that they can run in parallel. The planned trajectory is then transmitted to the flight controller for tracking. Once a collision is detected on the current trajectory or the UAV is far from the trajectory under tracking, a replan module is triggered and a new trajectory will be replanned to ensure flight safety in a complex environment. When the UAV's current position is within 1 m of the target waypoint, the waypoint is deemed as arrived, and the next waypoint will be used as the new target point triggering the above planning process.

Dynamic obstacle avoidance

A detector was designed to detect dynamic obstacles approaching the UAV. Because the dynamic obstacles are present only for a short period and appear at different positions in successive scans, points collected in recent scans (indicated by the last hitting point timestamp) and that have small hitting point counters are considered dynamic obstacles, whereas the rest of the points are considered static obstacles. In experiments, the threshold of the timestamp and the counter was well tuned to achieve a reliable detection performance. Once a dynamic obstacle is detected, a target point is generated such that its distance to the UAV is shorter than the nearest static points in the map and that it lies in a direction orthogonal to the object's incoming direction. The first condition ensures that the space between the UAV and the target point does not have any static obstacles, and the second condition facilitates the UAV to evade the dynamic obstacles. Last, the generated target point is sent to the flight controller for onboard control.

Supplementary Materials

This PDF file includes:
Supplementary Text
Figs. S1 to S13
Tables S1 to S5
References (*61*–*64*)

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S10

REFERENCES AND NOTES

1. M. Hasanzade, O. Shadeed, E. Koyuncu, Deep reinforcement learning based aggressive collision avoidance with limited fov for unmanned aerial vehicles, in *AIAA SCITECH 2022 Forum* (AIAA, 2022), p. 2043.
2. Y. Zhao, L. Yan, Y. Chen, H. Xie, B. Xu, Faep: Fast autonomous exploration planner for uav equipped with limited fov sensor. arXiv:2202.12507 [cs.RO] (25 February 2022).
3. C. M. Eaton, E. K. Chong, A. A. Maciejewski, Robust UAV path planning using pomdp with limited fov sensor, in *2017 IEEE Conference on Control Technology and Applications (CCTA)* (IEEE, 2017), pp. 1530–1535.
4. P. Theodorakopoulos, S. Lacroix, Uav target tracking using an adversarial iterative prediction, in *2009 IEEE International Conference on Robotics and Automation* (IEEE, 2009), pp. 2866–2871.
5. C. Yu, X. Chen, Leader-follower formation for UAVS with fovs constraint, in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)* (IEEE, 2021), vol. 1, pp. 1119–1124.
6. M. Miiller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, W. Stürzl, Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 3701–3708.
7. W. Gao, K. Wang, W. Ding, F. Gao, T. Qin, S. Shen, Autonomous aerial robot using dual-fisheye cameras. *J. Field Robot.* **37**, 497–514 (2020).
8. A. Gurtner, D. G. Greer, R. Glassock, L. Mejias, R. A. Walker, W. W. Boles, Investigation of fish-eye lenses for small-uav aerial photography. *IEEE Trans. Geosci. Remote Sens.* **47**, 709–721 (2009).
9. M. Tarhan, E. Altuğ, Ekf based attitude estimation and stabilization of a quadrotor uav using vanishing points in catadioptric images. *J. Intell. Robot. Syst.* **62**, 587–607 (2011).
10. V. M. Respall, D. Devitt, R. Fedorenko, Unmanned aerial vehicle path planning for exploration mapping, in *2020 International Conference Nonlinearity, Information and Robotics (NIR)* (IEEE, 2020), pp. 1–6.
11. J. Zhu, J. Zhu, X. Wan, C. Xu, Downside hemisphere object detection and localization of mav by fisheye camera, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (IEEE, 2018), pp. 532–537.
12. G. Kulathunga, R. Fedorenko, A. Klimchik, Regions of interest segmentation from LiDAR point cloud for multirotor aerial vehicles, in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2020), pp. 1213–1220.
13. A. Harmat, M. Trentini, I. Sharf, Multi-camera tracking and mapping for unmanned aerial vehicles in unstructured environments. *J. Intell. Robot. Syst.* **78**, 291–317 (2015).
14. D. Wierzbicki, Multi-camera imaging system for UAV photogrammetry. *Sensors* **18**, 2433 (2018).
15. P. Gohl, D. Honegger, S. Omari, M. Achtelik, M. Pollefeys, R. Siegwart, Omnidirectional visual obstacle detection using embedded FPGA, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 3938–3943.
16. G. Zhou, L. Fang, K. Tang, H. Zhang, K. Wang, K. Yang, Guidance: A visual sensing platform for robotic applications, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2015)*, (IEEE, 2015), pp. 9–14.
17. S. Zhao, H. Zhang, P. Wang, L. Nogueira, S. Scherer, Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments, in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021), pp. 8729–8736.
18. B. MacAllister, J. Butzke, A. Kushleyev, H. Pandey, M. Likhachev, Path planning for non-circular micro aerial vehicles in constrained environments, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3933–3940.
19. M. Karimi, M. Oelsch, O. Stengel, E. Babaians, E. Steinbach, Lola-slam: Low-latency lidar slam using continuous scan slicing. *IEEE Robot. Autom. Lett.* **6**, 2248–2255 (2021).
20. S. Jameson, K. Fregene, M. Chang, N. Allen, H. Youngren, J. Scroggins, Lockheed Martin's SAMARAI nano air vehicle: Challenges, research, and realization, in *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (AIAA, 2012), p. 584.
21. S. Bai, P. Chirarattananon, Design and take-off flight of a samara-inspired revolving-wing robot, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 6070–6076.

22. S. Bai, Q. He, P. Chirarattananon, A bioinspired revolving-wing drone with passive attitude stability and efficient hovering flight. *Sci. Robot.* **7**, eabg5913 (2022).
23. A. Safaee, S. Z. Moussavi, M. B. Menhaj, Design and construction of monocopter and its nonlinear control using photo diode array. *U. Porto J. Eng.* **4**, 34–41 (2018).
24. E. R. Ulrich, D. J. Pines, J. S. Humbert, From falling to flying: The path to powered flight of a robotic samara nano air vehicle. *Bioinspir. Biomim.* **5**, 045009 (2010).
25. K. Fregene, S. Jameson, D. Sharp, H. Youngren, D. Stuart, Development and flight validation of an autonomous mono-wing UAS, in *American Helicopter Society Forum* (Vertical Flight Society, 2010).
26. J. T. Isaacs, C. Magee, A. Subbaraman, F. Quitin, K. Fregene, A. Teel, U. Madhow, J. Hespanha, GPS-optimal micro air vehicle navigation in degraded environments, in *2014 American Control Conference* (IEEE, 2014), pp. 1864–1871.
27. D. Sharp, C. Stoneking, K. Fregene, Micro air vehicle based navigation aiding in degraded environments, in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)* (IEEE, 2016), pp. 305–312.
28. S. K. H. Win, L. S. T. Win, D. Sufiyan, G. S. Soh, S. Foong, An Agile samara-inspired single-actuator aerial robot capable of autorotation and diving. *IEEE Trans. Robot.* **38**, 1033–1046 (2022).
29. M. Piccoli, M. Yim, Passive stability of a single actuator micro aerial vehicle, in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 5510–5515.
30. M. Piccoli, M. Yim, Piccolissimo: The smallest micro aerial vehicle, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 3328–3333.
31. W. Zhang, M. W. Mueller, R. D'Andrea, A controllable flying vehicle with a single moving part, in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 3275–3281.
32. L. S. T. Win, S. K. H. Win, D. Sufiyan, G. S. Soh, S. Foong, Achieving efficient controlled flight with a single actuator, in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (IEEE, 2020), pp. 1625–1631.
33. S. K. H. Win, L. S. T. Win, D. Sufiyan, S. Foong, Design and control of the first foldable single-actuator rotary wing micro aerial vehicle. *Bioinspir. Biomim.* **16**, 066019 (2021).
34. S. Sun, G. Cioffi, C. De Visser, D. Scaramuzza, Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events. *IEEE Robot. Autom. Lett.* **6**, 580–587 (2021).
35. J. Paulos, M. Yim, An underactuated propeller for attitude control in micro air vehicles, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013), pp. 1374–1379.
36. J. Paulos, M. Yim, Flight performance of a swashplateless micro air vehicle, in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 5284–5289.
37. H. Huang, G. M. Hoffmann, S. L. Waslander, C. J. Tomlin, Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering, in *2009 IEEE International Conference on Robotics and Automation* (IEEE, 2009), pp. 3277–3282.
38. J. Paulos, B. Caraher, M. Yim, Emulating a fully actuated aerial vehicle using two actuators, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 7011–7016.
39. S. Gudmundsson, *General Aviation Aircraft Design: Applied Methods and Procedures* (Butterworth-Heinemann, ed. 2, 2021).
40. Y. Qin, W. Xu, A. Lee, F. Zhang, Gemini: A compact yet efficient Bi-copter UAV for indoor applications. *IEEE Robot. Autom. Lett.* **5**, 3213–3220 (2020).
41. D. Barcelos, A. Kolaei, G. Bramesfeld, Aerodynamic interactions of quadrotor configurations. *J. Aircr.* **57**, 1074–1090 (2020).
42. D. Shukla, N. Komerath, Multirotor drone aerodynamic interaction investigation. *Drones* **2**, 43 (2018).
43. W. Zhou, Z. Ning, H. Li, H. Hu, An experimental investigation on rotor-to-rotor interactions of small UAV propellers, in *35th AIAA Applied Aerodynamics Conference* (AIAA, 2017), p. 3744.
44. D. Verstraete, R. MacNeill, The effects of blockage on the performance of small propellers, in *20th Australasian Fluid Mechanics Conference* (Australasian Fluid Mechanics Society, 2016).
45. P. Foehn, A. Romero, D. Scaramuzza, Time-optimal planning for quadrotor waypoint flight. *Sci. Robot.* **6**, eabh1221 (2021).
46. D. Falanga, K. Kleber, D. Scaramuzza, Dynamic obstacle avoidance for quadrotors with event cameras. *Sci. Robot.* **5**, eaaz9712 (2020).
47. B. He, H. Li, S. Wu, D. Wang, Z. Zhang, Q. Dong, C. Xu, F. Gao, Fast-dynamic-vision: Detection and tracking dynamic objects with event and depth sensing, in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021), pp. 3071–3078.
48. O. Carholt, E. Fresk, G. Andrikopoulos, G. Nikolakopoulos, Design, modelling and control of a single rotor UAV, in *2016 24th Mediterranean Conference on Control and Automation (MED)* (IEEE, 2016), pp. 840–845.
49. J. Buzzatto, M. Liarokapis, An agile, coaxial, omnidirectional rotor module: On the development of hybrid, all terrain robotic rotorcrafts, in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (IEEE, 2020), pp. 162–168.
50. Q. Zhang, Z. Liu, J. Zhao, S. Zhang, Modeling and attitude control of bi-copter, in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)* (IEEE, 2016), pp. 172–176.
51. B. Li, L. Ma, D. Huang, Y. Sun, A flexibly assembled and maneuverable reconfigurable modular multirotor aerial vehicle. *IEEE ASME Trans. Mechatron.* **27**, 1704–1714 (2022).
52. W. Hao, B. Xian, T. Xie, Fault-tolerant position tracking control design for a tilt tri-rotor unmanned aerial vehicle. *IEEE Trans. Ind. Electron.* **69**, 604–612 (2022).
53. J. J. Paulos, M. Yim, Scalability of cyclic control without blade pitch actuators, in *2018 AIAA Atmospheric Flight Mechanics Conference* (AIAA, 2018), pp. 1–18.
54. T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, D. Scaramuzza, Event-based motion segmentation by motion compensation, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2019), pp. 7244–7253.
55. Mavros,https://github.com/mavlink/mavros.
56. W. Xu, Y. Cai, D. He, J. Lin, F. Zhang, FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Trans. Robot.* **38**, 2053–2073 (2022).
57. Y. Cai, W. Xu, F. Zhang, ikd-Tree: An incremental K-D tree for robotic applications. arXiv:2102.10808 [cs.RO] (22 February 2021).
58. F. Kong, W. Xu, Y. Cai, F. Zhang, Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots. *IEEE Robot. Autom. Lett.* **6**, 7869–7876 (2021).
59. A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon path planning for 3d exploration and surface inspection. *Auton. Robots* **42**, 291–306 (2018).
60. N. Chen, Data of PULSAR, Zenodo (2023); https://doi.org/10.5281/zenodo.7656487.
61. APC propeller database,https://www.apcprop.com/technical-information/performance-data/.
62. B. W. McCormick, *Aerodynamics, Aeronautics, and Flight Mechanics* (John Wiley & Sons, 1994).
63. H. Glauert, *Aerodynamic Theory* (Springer, 1935), pp. 169–360.
64. R. W. Deters, G. K. Ananda Krishnan, M. S. Selig, Reynolds number effects on the performance of small-scale propellers, in *32nd AIAA Applied Aerodynamics Conference* (AIAA, 2014), p. 2151.

Acknowledgments: We would like to thank Y. Ren and G. Lu for help with the mechanical design and helpful discussions. **Funding:** This work was supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) (no. 17206920), the Hong Kong Research Grants Council (RGC) Early Career Scheme (ECS) (no. 27202219), and a DJI research donation.

Author contributions: N.C. and W.X. proposed the initial idea of the research. N.C. designed and manufactured the UAV prototype, formulated the dynamic model with F.Z., and implemented all software modules for the UAV with the help of F.K., W.X., and Y.C. The experiments were designed by N.C. and performed by N.C., F.K., W.X., Y.C., D.H., and Y.Q. N.C. and H.L. contributed to the data acquisition on a test stand. N.C. finished all data analyses. N.C. and F.Z. wrote the manuscript with the advice of all authors. F.Z. provided funding and supervised the research. **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 and in an online database (*60*). The design files and source code are also available on GitHub at https://github.com/hku-mars/PULSAR.

Submitted 19 August 2022
Accepted 21 February 2023
Published 15 March 2023
10.1126/scirobotics.ade4538

Science Robotics

A self-rotating, single-actuated UAV with extended sensor field of view for autonomous navigation

Nan Chen, Fanze Kong, Wei Xu, Yixi Cai, Haotian Li, Dongjiao He, Youming Qin, and Fu Zhang

Sci. Robot. **8** (76), eade4538. DOI: 10.1126/scirobotics.ade4538

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.