ARTICLE OPEN

Aerodynamic effect for collision-free reactive navigation of a small quadcopter

Runze Ding[1,3], Songnan Bai[1,3], Kaixu Dong[1] and Pakpong Chirarattananon[1,2]✉

The small footprint of tiny multirotor vehicles is advantageous for accessing tight spaces, but their limited payload and endurance impact the ability to carry powerful sensory and computing units for navigation. This article reports an aerodynamics-based strategy for a ducted rotorcraft to avoid wall collisions and explore unknown environments. The vehicle uses the minimal sensing system conventionally conceived only for hovering. The framework leverages the duct-strengthened interaction between the propeller wake and vertical surfaces. When incorporated with the flight dynamics, the derived momentum-theory-based model allows the robot to estimate the obstacle's distance and direction without range sensors or vision. To this end, we devised a flight controller and reactive navigation methods for the robot to fly safely in unexplored environments. Flight experiments validated the detection and collision avoidance ability. The robot successfully identified and followed the wall contour to negotiate a staircase and evaded detected obstacles in proof-of-concept flights.

npj Robotics (2023)1:2 ; https://doi.org/10.1038/s44182-023-00002-9

INTRODUCTION

Small aerial vehicles have established their presence in several civilian applications, including additive manufacturing[1], logistics[2], and wildlife surveys[3]. This is attributed to their versatility and unrivaled accessibility to hazardous or remote locations[4]. To accomplish the assignments, the dependence of these robots on human intervention varies according to their sensing and computational ability[5]. Thus far, the levels of control autonomy achieved by different flying robots strongly correlate with their sizes. Multirotor vehicles weighing 200−300 g and above have displayed a certain degree of "cognitive autonomy" as defined in ref. [5]. Equipped with multiple sensors, such as monocular/RGB-D cameras, LiDAR, or ultra-wideband modules, these robots rely on a plethora of information to robustly navigate through cluttered environments and avoid collisions with onboard perception using visual-inertial odometry[6–9].

On the other end of the spectrum, insect-sized aerial robots face immense challenges when it comes to power and control autonomy[10–14]. The aerodynamic and actuation efficiencies, as well as the energy density of batteries, of small vehicles are substantially demoted, resulting in limited payload capacity and endurance[5,14,15]. This is aggravated by the limitations of sensory devices in terms of mass, power, and bandwidth. While sensor quality often degrades with size, the dynamics of small flyers are inherently faster, calling for sensors with higher sampling rates and lower latency for stabilization and control[14,16]. Furthermore, the hardware and computational power required for vision-based localization remains too heavy and energy-intensive for small drones (for example, NVIDIA Jetson Xavier NX module used in ref. [6] weighs 79 g with the power consumption of 10–20 W while NVIDIA Jetson TX2 in ref. [7] weighs 154 g with the power consumption of 8–15 W). For these reasons, sub-gram robots have yet to exhibit sustained untethered flight even at the level of "sensory-motor autonomy" (staying airborne or hovering stably).

In between, rotorcraft in the range of 30–100 g can usually accommodate a minimal sensor suite conventionally deemed sufficient for sensory-motor autonomy, consisting of an inertial measurement unit (IMU, including gyroscope and accelerometer), an optic flow camera, and a range finder (such as a time-of-flight or ToF sensor)[14,17,18]. Without using map-based navigation, mobile and aerial robots can benefit from image motion and optic flow[19,20]. Together, an optic flow camera and an emissive ToF sensor are lightweight and efficient sensors for stabilizing flight speed and altitude, respectively. At this scale, "reactive autonomy" has been accomplished. To traverse through cluttered environments, the robot may employ an insect-inspired wide-field camera, accompanied by an efficient optic flow-based motion-sensitive detector[21–25] or multiple ranger finders[17]. Without dealing with high-resolution images for the construction of detailed maps, smaller robots with limited payload and computational capacities can still avoid collisions efficiently.

Recently, other miniature sensory packages have been introduced to supplement or substitute optic flow and a range finder on top of the minimal sensor suite to enable small robots to sense the surroundings and evade obstacles. In ref. [26], the authors developed an audio extension deck, comprising of a piezoelectric buzzer and four microphones, to imitate bats' echolocation method. The strategy has yielded reliable wall detection with a precision of around 8 cm on a flying robot but with limited accuracy on the wall direction. Another bio-inspired approach is based on the behavior of mosquitoes. The authors in ref. [27] cite the observed nocturnal collision-avoidance mechanism, conjecturing that the self-induced wake, after interacting with a ground or wall, is a cue of mediated response from mechanosensory antennae. They assembled a 9.2-g electronic module featuring five pairs of pressure probes for a palm-sized multirotor platform and relied on differential pressure data to sense the presence of floors and walls. Nevertheless, the device in ref. [27] was unable to measure the distance or direction of the detected surfaces. The use of differential pressure measurements for surface detection with a small flying robot is further investigated in ref. [28]. The study shows that multiple pressure measurements, when combined

[1]Department of Biomedical Engineering, City University of Hong Kong, Tat Chee Avenue, Hong Kong SAR, China. [2]Department of Mechanical Engineering, City University of Hong Kong, Tat Chee Avenue, Hong Kong SAR, China. [3]These authors contributed equally: Runze Ding, Songnan Bai. ✉email: pakpong.c@cityu.edu.hk

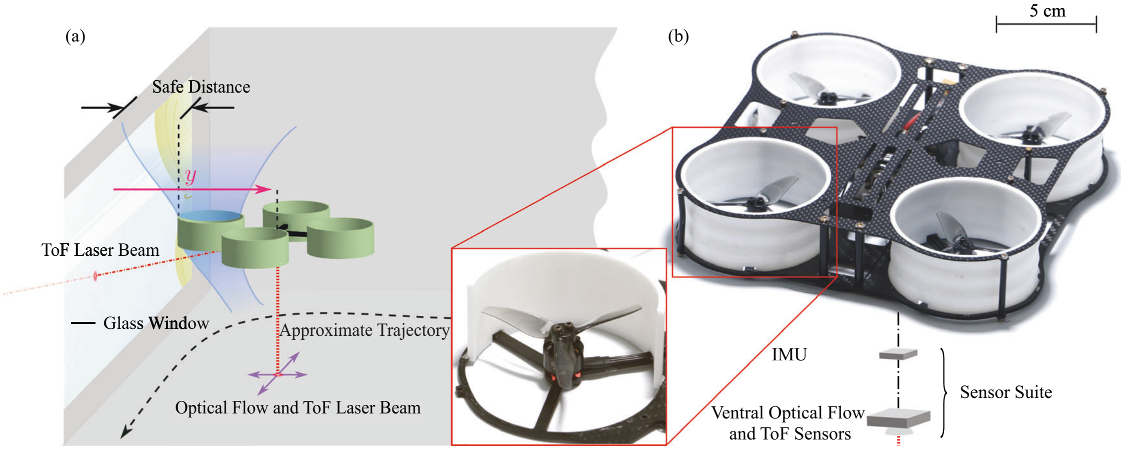

Fig. 1 Collision-free navigation by a rotorcraft with ducted propellers. **a** The robot with only an IMU and ventral optical flow and ToF sensors is able to avoid a wall collision through the sensing of wall-induced aerodynamic force. Unlike a ToF sensor, the aerodynamics-based method is multi-directional and is robust against transparent (glass) or reflective (mirror) surfaces. **b** Photograph of the robot, featuring four ducted rotors.

with empirical models, provide the ground and wall distance estimates in free flight with varying degrees of accuracy. However, the wall distance estimate is relatively imprecise and the detection alone was not sufficient to prevent the robot from colliding with a wall due to the destabilizing aerodynamic interactions. Despite some limitations, unlike ToF sensors and echolocators, the aerodynamics-based techniques in refs. [27,28] are highly power efficient as they involve no emission of electromagnetic or acoustic waves.

Despite the aforementioned developments, robustly avoiding a collision or achieving reactive autonomy remains a challenge for small robots with restricted payload capacity. This study offers a strategy for small rotorcraft to achieve reactive autonomy using a minimal suite of sensors (an IMU with downward-looking optic flow and ToF sensors), which has traditionally been regarded as sufficient only for sensory-motor autonomous flight. In other words, we efficiently equip aerial robots with the ability to estimate the distance and direction of a wall without extra sensors as seen in refs. [17,21,22,26–28]. This enables small aerial vehicles to safely navigate complex environments with minimal sensing.

Similar to refs. [27,28], we exploit the propeller-surface interactions. Previously investigated as ground[29–31], ceiling[32–36], and wall[37–39] effects (collectively known as the proximity effect[27,28,37,40–42]), aerial vehicles exhibit discernible interactions when they operate near horizontal or vertical surfaces that oftentimes undesirably lead to crashes if not properly accounted for[28,36,37,39,43,44]. Nonetheless, the short-range nature of the interactions makes it difficult for a robot to robustly detect, let alone estimate the distance and direction, the presence of a wall in time for control purposes even when multiple barometric sensors are used[27,28]. To overcome the challenge, we introduce a small quadrotor with ducted propellers as depicted in Fig. 1b and Movie 1. Through extensive measurements, we demonstrate that the ducts markedly amplify the wall effect, both in terms of range and magnitude. The results, after integration with the momentum-theory-based thrust model and flight dynamics, permit the robot to reliably estimate the distance and direction of a nearby wall using solely the sensors required to hover. Subsequently, we devise a flight controller for the robot to safely regulate its position against a detected wall without barometers. Hence, the robot attains reactive autonomy, flying and avoiding collisions in both indoor and outdoor settings without extra sensors or visual-inertial navigation.

To summarize, the technical contribution of the work is threefold. First, we conceive an aerodynamics-based framework to realize reactive autonomy with a small robot via a minimal set of sensor suites previously used for realizing hovering flight only. Second, a physics-based model is combined with experimental measurements to elucidate the propeller-wall interaction. Third, the devised thrust model is incorporated with the flight dynamics, leading to reliable in-flight estimation of wall distance and direction. This is accompanied by a flight controller that proficiently stabilizes the robot near a wall, allowing the robot to react, fly, and navigate safely without a collision in various real-world settings.

RESULTS

Ducted rotors and proximity effects

We first employ momentum theory (MT) to predict how the thrust and power of a ducted propeller are influenced by a nearby wall. This differs from several studies that the wall effect was only investigated experimentally[28,37,41]. Then, benchtop experiments were conducted to validate the derived models.

We consider a propeller of radius R fitted in a duct as featured in Fig. 2a. The inner radius of the thin duct is assumed marginally larger than (effectively equal to) the radius of the propeller. This is to reduce the tip loss and, hence, improve the aerodynamic efficiency[45]. A vertical surface is located at the distance d from the ducted propeller as shown in Fig. 2. In a steady state, thrust $\boldsymbol{T}$ is generated and the corresponding aerodynamic power is denoted by P_a. Momentum theory (MT) is applied to derive the relationship between the propelling thrust and aerodynamic power.

For modeling, the flow is assumed quasi-steady, incompressible, inviscid, and irrotational with constant density ρ. These conditions are commonly presumed for similar analyses[34]. Akin to our previous work[38], the scenario depicted by Fig. 2a differs from a standard application of MT[46] thanks to the presence of the wall, which is conjectured to break the axial symmetry of the flow. As a result, it is hypothesized that the infinitesimally thin effective propeller disk that resides within the duct may not necessarily coincide with the physical propeller plane, but tilted by θ_t. Subsequently, the associated thrust vector $\boldsymbol{T}$, perpendicular to the effective propeller disk of area $A = \pi R^2$, deviates from the vertical axis by θ_t. Similarly, the direction of the terminal wake may make a non-zero angle θ_∞ from the vertical. To retain the conservation of energy assumption, the drags induced by the proximate surface and the duct are neglected.

As detailed in Methods, the application of momentum and energy conservation principles leads to the following relation

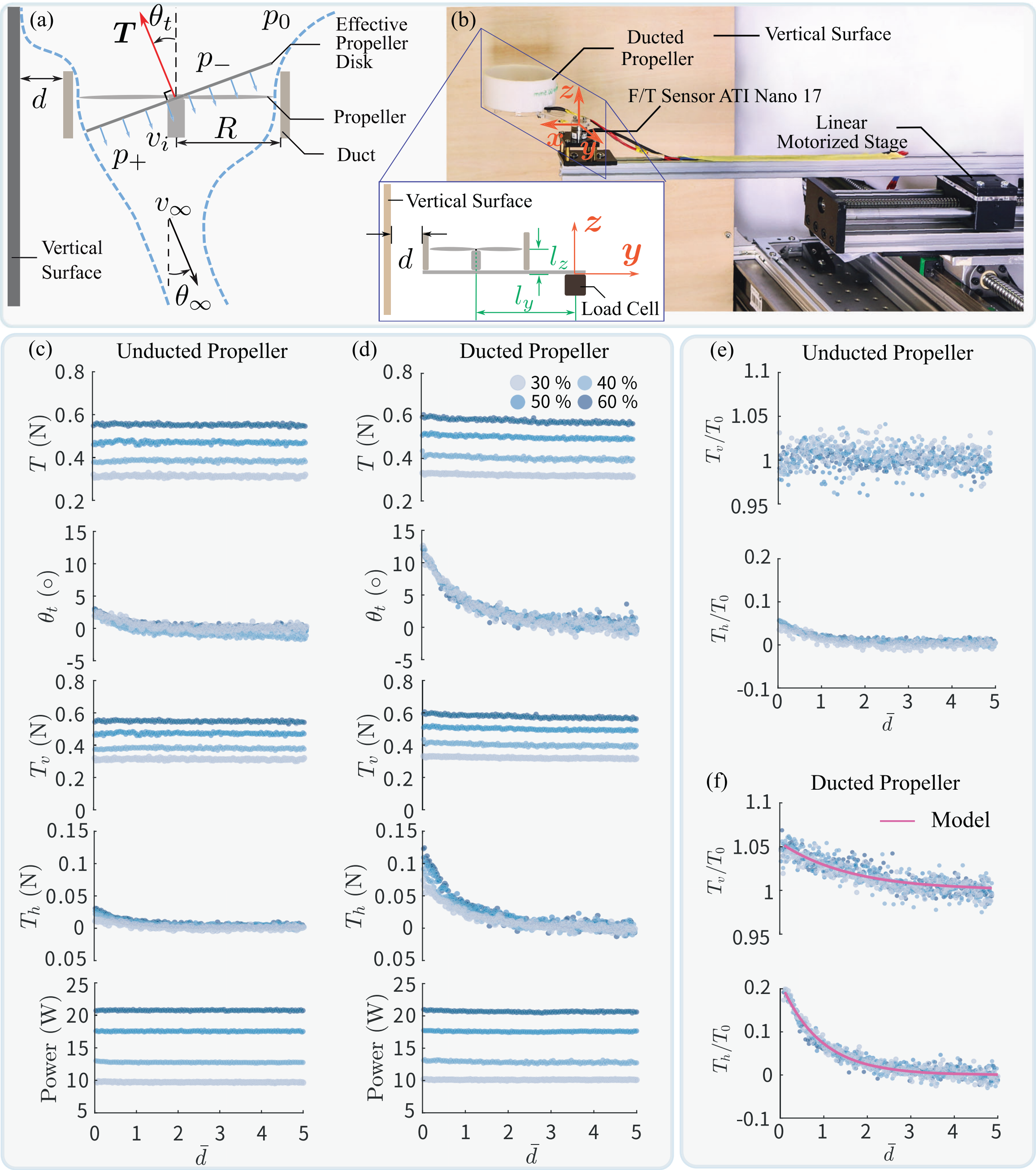

Fig. 2 Experimental measurements of the proximity effects. a A schematic drawing shows the effective propeller disk and the wake interaction when a ducted propeller is in the vicinity of a wall. b Photo of the platform used for the force and power measurements. The ducted propeller is affixed on a load cell and mounted on a motorized stage. Different driving commands and distances to the surface were tested. The inset shows a drawing of the setup from another perspective. **c** Force and power measurements of an unducted propeller. **d** Force and power measurements of a ducted propeller. Four command levels were applied. **e** Normalized forces of the unducted propeller. **f** Normalized forces and fitted models of the proximity effect belonging to the ducted propeller.

between the thrust magnitude T and aerodynamic power P_a.

$$T = \gamma\left(\sqrt{2\rho A P_a}\right)^{\frac{2}{3}}, \qquad (1)$$

where $\gamma = (\cos\theta_\infty / \cos\theta_t)^{\frac{2}{3}}$ can be regarded as a factor capturing the thrust and wake directions. Without the wall or far away, the wake and thrust become vertical: $\theta_t, \theta_\infty \to 0$ and $\gamma \to 1$, and the aerodynamics power reduces to $P_a = T\sqrt{T/2\rho A}$, matching the result of the classical MT. Near the surface, the value of γ deviates from unity and depends on the distance d and physical parameters of the duct and the propeller.

In the context of a multirotor robot, we separately inspect the horizontal T_h and vertical T_v components of the thrust. They can be deduced from Fig. 2a and (1) as

$$\begin{aligned} T_h &= T\sin\theta_t = \gamma_h(\sqrt{2\rho A P_a})^{\frac{2}{3}}, \\ T_v &= T\cos\theta_t = \gamma_v(\sqrt{2\rho A P_a})^{\frac{2}{3}}, \end{aligned} \qquad (2)$$

where $\gamma_h = \gamma\sin\theta_t$ and $\gamma_v = \gamma\cos\theta_t$ are the coefficients of proximity effects defined to capture the effects of the duct and distance to the surface. Introducing the normalized distance $\bar{d} = d/R$, we may express $\gamma_h(\theta_t, \theta_\infty), \gamma_v(\theta_t, \theta_\infty)$ as $\gamma_h(\bar{d}), \gamma_v(\bar{d})$.

Therefore, the relationship between the aerodynamic power and thrust components are captured by $\gamma_h(\bar{d})$ and $\gamma_v(\bar{d})$.

Benchtop characterization of the Wall effect

To validate the MT-based model captured by (2), we devised a benchtop setup shown in Fig. 2. As detailed in Methods, a ducted propeller was mounted on a load cell next to a vertical surface. The setup was placed on top of a linear motorized stage. Force, torque, and power measurements were taken at different distances d from 1 to 200 mm at the step of 1 mm. Four throttle commands (from 40 to 70%) were used at each distance. The experiments were repeated without the duct as benchmark measurements.

The measurements of thrust and electrical power (product of the supplied voltage and measured currents) in two cases (with and without the duct) are plotted against the normalized distance $\bar{d}$ in Fig. 2c and d. For both cases, the electrical powers remain constant for each thrust command regardless of $\bar{d}$. At a large distance ($\bar{d}>4$), compared to the unducted propeller, a slight increase in the thrust generated by the ducted propellers is observed, likely due to the reduced tip-loss as reported in ref. [45]. The increase of around 3−5 g somewhat compensates for the 7-g mass of each duct, making their impact on flight efficiency insignificant when used with the robot.

Next, focusing on the near-wall measurements ($\bar{d}$<3), the proximity effects slightly affect the overall thrust magnitude T. The horizontal thrust T_h grows as $\bar{d} \to 0$ whereas the vertical thrust T_v remains relatively unchanged. As a result, the thrust vector tilts towards the wall as described by positive θ_t. The proximity effects are noticeably more pronounced with the ducted propeller (θ_t as large as 10° versus lower than 5° for the unducted propeller). In other words, the duct markedly strengthens the proximity effects for all throttle commands, making it possible for the robot to use the effect for reactive navigation as shown below.

The findings in the form of (2) cannot be readily applied to the obtained benchtop measurements as the aerodynamic power P_a is unavailable. Nevertheless, as explained in refs. [34,38], with the use of an electronic speed controller (ESC), the mappings between the throttle commands, aerodynamic powers, and electrical powers are one-to-one. This is in accordance with the experimental findings in Fig. 2c and d, which clearly demonstrate a constant electrical power for each throttle command. Furthermore, for each throttle command (and its associated P_a), we elect $T_0 = (\sqrt{2\rho A}P_a)^{\frac{2}{3}}$ to denote the nominal thrust value as the magnitude of T_v far from a vertical surface. Doing so, (2) simplifies to

$$T_h = \gamma_h(\bar{d})T_0, \quad \text{and} \quad T_v = \gamma_v(\bar{d})T_0. \tag{3}$$

Subsequently, $\gamma_h(\bar{d}) = T_h/T_0$ and $\gamma_v(\bar{d}) = T_v/T_0$ can be deduced from the benchtop measurements. For each throttle command, T_0 is evaluated by averaging T_v at large values of $\bar{d}$ ($4.0 \leq \bar{d} \leq 5.0$). As presented in Fig. 2e and f, the values of T_h/T_0 and T_v/T_0 from four throttle commands collapse together, corroborating the MT analysis and (3) on the existence of $\gamma_h(\bar{d})$ and $\gamma_v(\bar{d})$ for both ducted and unducted propellers.

For estimation and control purposes, analytic forms of $\gamma_h(\bar{d})$ and $\gamma_v(\bar{d})$ are preferred. Unfortunately, they cannot be directly derived from MT. While it might be feasible to employ blade-element theory and/or computational fluids dynamics for the task, the outcomes are likely dependent on the blade geometry and other physical parameters[33,34] and become overly complex due to the duct and wall interactions. Hence, we propose empirical models based on the measurements in Fig. 2c and d,

$$\gamma_h(\bar{d}) = a_1 b_1^{\bar{d}}, \quad \gamma_v(\bar{d}) = 1 + a_2 b_2^{\bar{d}} \tag{4}$$

where $a_i > 0$ and $b_i \in (0, 1)$ are numerical coefficients. In this form, γ_h approaches zero and γ_v approaches one when $\bar{d} \to \infty$ as anticipated. The least squares regression gives $\gamma_h = 0.21(0.34)^{\bar{d}}$ and $\gamma_v = 1 + 0.06(0.54)^{\bar{d}}$ with the R squared values of 0.95 and 0.71, respectively. The fitted results are shown in Fig. 2c. Despite a decent fit, the R squared value for γ_v is relatively low. This is because the variation of γ_v (≈0.06) over the entire range of $\bar{d}$ is small in comparison to the spread of data points around the best-fit line (≈± 0.02).

Rotorcraft with ducted propellers

To make use of the proximity effects, a quadrotor with ducted propellers ($R = 38$ mm) presented in Fig. 1b is constructed. The dynamics of the robot when it operates in the vicinity of a vertical surface are described. Then, the details of the prototype are provided.

For dynamic modeling, the robot with mass m and four ducted propellers is assigned the body-fixed frame $\mathcal{B} = \{\boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$ as seen from the top in Fig. 3a. Without loss of generality, the inertial frame, defined as $\mathcal{W} = \{\boldsymbol{x}_W, \boldsymbol{y}_W, \boldsymbol{z}_W\}$, is assumed located on the wall with $\boldsymbol{y}_W$ being the surface normal. The robot's position with respect to $\mathcal{W}$ is denoted by $\boldsymbol{p} = [x, y, z]^T$. The rotation matrix $\boldsymbol{R}$ relates the attitude of the body frame to the inertial frame.

With four rotors, we let subscript $i \in \{1, 2, 3, 4\}$ label the i^{th} propeller as illustrated in Fig. 3a. Further, we let $\boldsymbol{r}_i$ be the vector from the center of mass to the i^{th} propeller defined in $\mathcal{B}$ and $\boldsymbol{e}_i$'s be

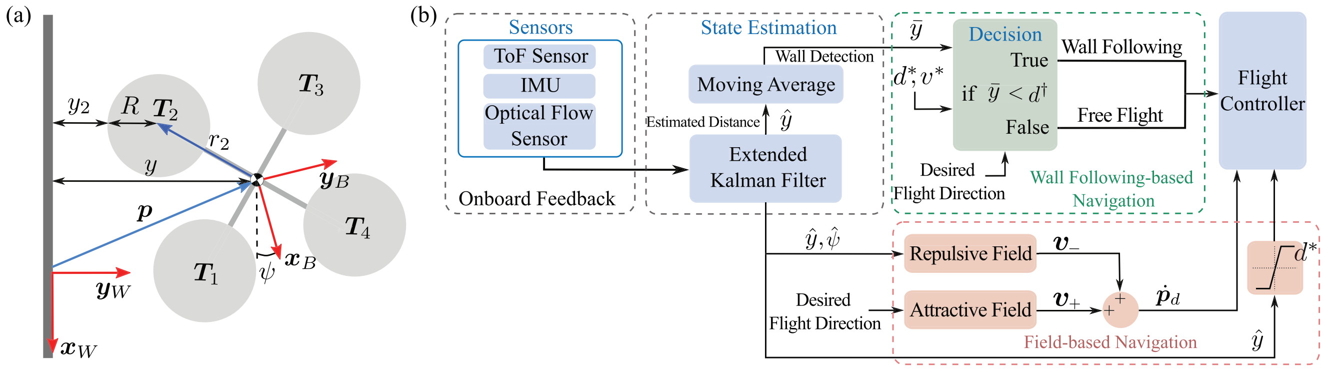

Fig. 3 Wall detection and navigation methods. a Coordinate frames and robot's configuration. The world frame $\mathcal{W}$ is located on the wall, with $\boldsymbol{y}_W$ being the surface's normal. **b** Structure of the reactive navigation methods, consisting of the wall distance estimation and control. The pipeline includes an EKF for wall detection and a flight controller for flight stabilization. The estimation only needs onboard feedback from an IMU and downward-looking optic flow camera and ToF sensor. In wall following-based navigation, the decision module navigates the robot to the goal unless it detects an obstructing surface, in which case it directs the robot to travel along the wall to avoid a collision. Alternatively, the field-based method dynamically evaluates a collision-free path to negotiate obstacles.

basis vectors. It follows that $y_i = y + \boldsymbol{e}_2^T \boldsymbol{R} \boldsymbol{r}_i - R$ is the distance between the i^{th} propeller and the wall.

Due to the anticipated rotor-wall interactions, the force and torque generated by the vehicle are slightly different from conventional rotorcraft in free flight. For the i^{th} propeller with the thrust command $T_{0,i}$, the nominal thrust vector is $T_{0,i}\boldsymbol{e}_3$ with respect to the body frame. To obtain the actual thrust in $\mathcal{W}$, we assume only the vertical component (normal to the wall surface) contributes to the proximity effects (this is valid when the robot is approximately upright). Hence, $T_{0,i}\boldsymbol{e}_3$ must be not only projected but also adjusted as

$$\boldsymbol{T}_{W,i} = T_{0,i}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\gamma_h(y_i) \\ 0 & 0 & \gamma_v(y_i) \end{bmatrix}\boldsymbol{R}\boldsymbol{e}_3 = T_{0,i}\boldsymbol{\Gamma}_i(y_i)\boldsymbol{R}\boldsymbol{e}_3, \tag{5}$$

where we have defined matrix $\boldsymbol{\Gamma}_i(y_i)$ to reflect the influence of the wall. Eq. (5) assumes the proximity effects result in the amplification of the vertical component of the thrust by a factor of $\gamma_v(y_i)$ and adds the horizontal force attracting the propeller to the surface (in the $-y_W$ direction) through $\gamma_h(y_i)$. This wall attraction may destabilize the robot as mentioned in refs. [27,28,43] if not compensated for. Notating $\boldsymbol{T}_W = \sum_{i=1}^4 \boldsymbol{T}_{W,i}$ as the total thrust vector in the inertial frame, the translational dynamics of the vehicle follows

$$m\ddot{\boldsymbol{p}} + mg\boldsymbol{e}_3 = \boldsymbol{T}_W. \tag{6}$$

As anticipated, $\boldsymbol{T}_W$ converges to $\boldsymbol{R}\boldsymbol{e}_3\sum_{i=1}^4 T_{0,i}$ when $y \to \infty$. Meanwhile, the generated torque, in the body frame, can be computed by projecting $\boldsymbol{T}_{W,i}$ back to the body frame and including rotors' locations and spinning directions as

$$\boldsymbol{\tau} = \sum_{i=1}^{4}\left([\boldsymbol{r}_i]_\times + (-1)^{i+1}c_\tau \boldsymbol{I}_{3\times3}\right)\boldsymbol{R}^T\boldsymbol{T}_{W,i} \tag{7}$$

where $[\boldsymbol{r}_i]_\times$ is the matrix representation of $\boldsymbol{r}_i \times$ and c_τ is the thrust-to-torque coefficient accounting for the rotor drag. Each propeller's spinning direction is taken into consideration by $(-1)^{i+1}$. The formulation exploits the fact that the proximity effect does not affect the thrust location as shown in Supplementary Note 1 and Supplementary Fig. 1. Consolidating with (5), (7) can be manipulated into the form $\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{T}_0$, where $\boldsymbol{B}$ is a 3×4 matrix and $\boldsymbol{T}_0 = [T_{0,1}, T_{0,2}, T_{0,3}, T_{0,4}]^T$. Due to the proximity effects, $\boldsymbol{B}$ is not constant but dependent on y and $\boldsymbol{R}$. Lastly, the attitude dynamics of the robot is

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} = \boldsymbol{B}\boldsymbol{T}_0, \tag{8}$$

where $\boldsymbol{J}$ is the moment of inertia, and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ is the body-centric angular velocity.

The prototype with four ducted propellers is shown in Fig. 1b. The actuation components are identical to the parts used for the benchtop measurements. With a flight control board (Crazyflie Bolt), a battery (2S LiPo), and an onboard sensor suite (Crazyflie Flow Deck V2), the robot weighs 280 g and is capable of producing over 440 g of thrust.

Without a camera for visual odometry, the robot is equipped with an IMU. In addition, the Flow Deck consists of downward-looking ToF and optical flow sensors. Consequently, the ToF sensor directly measures the robot's altitude z. Similarly, the optical flow sensor nominally outputs a two-dimensional robot-centric perceived velocity $[n_x, n_y]$, scaled by the distance (altitude)[47], as

$$n_x = (\boldsymbol{e}_1^T\boldsymbol{R}^T\dot{\boldsymbol{p}})z^{-1} - \omega_y, \tag{9}$$

$$n_y = (\boldsymbol{e}_2^T\boldsymbol{R}^T\dot{\boldsymbol{p}})z^{-1} + \omega_x, \tag{10}$$

where the contribution from the rotation (ω_x, ω_y) is taken into consideration. The integration of the IMU, ToF, and optical flow sensors enables the robot-centric translational velocity ($\boldsymbol{e}_1^T\boldsymbol{R}^T\dot{\boldsymbol{p}}, \boldsymbol{e}_1^T\boldsymbol{R}^T\dot{\boldsymbol{p}}$) to be deduced.

Strategy for wall detection and near-wall flight control

Here, we first present the framework for estimating the distance to a wall y and its relative orientation ψ. Without a direct distance measurement or vision, we only rely on inertial and optical flow sensors. This is achieved by an Extended Kalman Filter (EKF). Then, a method is proposed to stabilize the robot at the desired position and orientation with respect to the detected wall.

In order to estimate the distance and direction of a vertical surface nearby, we introduce the state tuple:

$$\mathbf{x} := \{\bar{\boldsymbol{p}}, \dot{\boldsymbol{p}}, \boldsymbol{R}, \boldsymbol{\omega}\} \tag{11}$$

where $\bar{\boldsymbol{p}} = [y, z]^T$ contains the distance to the wall y and the altitude z. Remark that the position of the robot in the horizontal direction parallel to the surface of the wall is not included as it is not observable. Since $\boldsymbol{R}$ is defined with respect to the wall, the surface direction is implicit. The time evolution of $\mathbf{x}$ is provided by the translational (6) and rotational (8) dynamics as described in Methods. After each iteration, they are updated via the onboard feedback through the measurement vector $\mathbf{y} = [\boldsymbol{a}_m^T, \boldsymbol{w}_m^T, \boldsymbol{n}_m^T, z_m]^T$ according to the associated measurement model $\mathbf{y} = \boldsymbol{h}(\mathbf{x}) + \boldsymbol{v}$. The implementation allows us to use only the onboard feedback to estimate $\mathbf{x}$, including the wall distance and direction. The estimated distance can be used to stabilize the position of the robot in the proximity of a wall.

To regulate the position of the robot near the wall, we let $\boldsymbol{p}_d$ denote the desired position and devised a flight control scheme to minimize the position error $\boldsymbol{p} - \boldsymbol{p}_d$. To do so, the effects of the surface on the generated thrust shown in (6) must be dealt with. To simplify the process, we regard the proximity effects as modeling errors or disturbances. To make up for this, a flight controller based on the incremental nonlinear dynamic inversion (INDI) is employed thanks to its inherent ability to compensate for modeling errors[48]. This is because the controller re-adjusts the commands based on the sensory feedback until the errors are eliminated.

Using the estimated wall location, optic flow feedback, and attitude of the robot, the INDI position controller and the inner loop attitude controller determines the individual rotor commands for stabilizing the robot as detailed in Methods. In addition, the yaw control is included for the robot to retain its heading. The cascaded control loops stabilize the vehicle to the desired position setpoint $\boldsymbol{p}_d$ and heading angle ψ_d. The approach only requires the distance and direction of the wall, which are estimated from onboard feedback by the EKF as explained earlier.

To verify the estimation algorithm and the INDI controller using the robot depicted in Fig. 1b, the experiments were first carried out in the laboratory setting for complete ground truth measurements. To do so, the robot communicates with a ground station computer (Ubuntu 18.04) running Python scripts for the position control through radio. To accelerate the implementation, the estimator was first implemented with Python on the ground computer with a sample rate of 100 Hz. Later, we ported the estimator over to the onboard flight controller for the final set of experiments in order to demonstrate the computational efficiency of the scheme. In the laboratory, a motion capture system (MoCap) was employed to provide ground truth position and attitude data.

In-flight distance estimation and INDI position control

To independently verify the effectiveness of the EKF estimator and the INDI controller, we first carried out a near-wall flight with both the estimator and the controller. However, the controller used position feedback from the MoCap. This way, the performance of the estimator and controller can be separately tested.

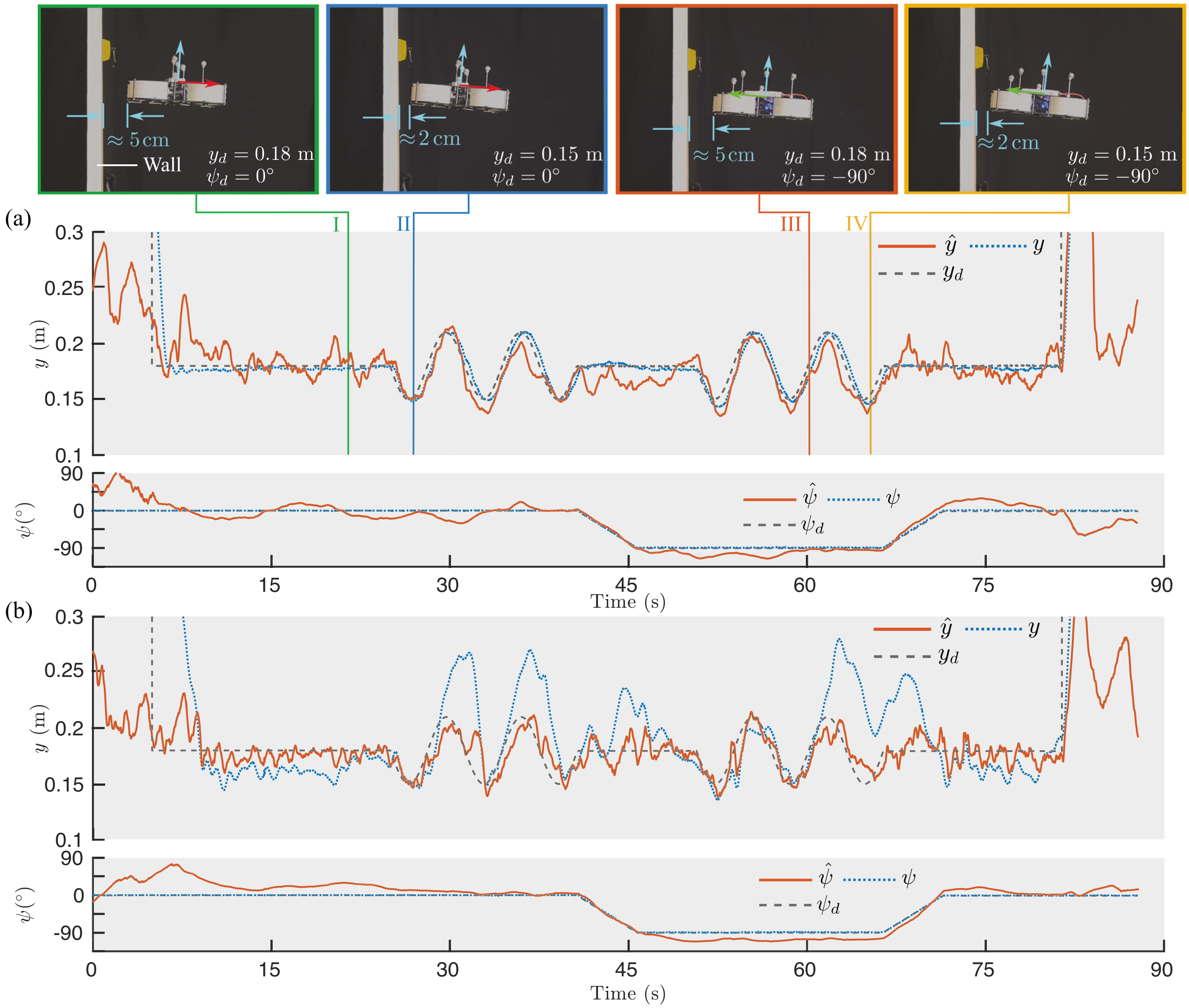

Fig. 4 Estimation and control of wall distance and direction. **a** Plots of the robot's position and orientation (reference, actual, and estimated) with respect to the wall. In this flight, the control was based on MoCap feedback. The trajectory highlights the estimation performance at varying distances and angles. **b** Plots of the robot's position and orientation (reference, actual, and estimated) with respect to the wall. In this flight, the control was entirely from onboard feedback.

After taking off at ≈40 cm from the wall, the robot was commanded by the INDI controller at $t = 5$ s to stay at an altitude of 1.0 m and the distance $y_d = 18$ cm from the wall while maintaining the orientation setpoint of $\psi_d = 0°$ for 15 s. In this configuration, the distance between the surface and the nearest ducted propeller was ~5 cm. At $t = 25$ s, we varied the setpoint position according to the sinusoidal function $y_d = 0.18 - 0.03\sin(t - 25)$ m for $25 \le t \le 41$ s. Thereafter, the setpoint yaw angle ψ_d was varied by 90° in 30 s twice, starting at $t = 41$ s and $t = 66$ s. This translates to the change in the relative wall direction. In between (when $\psi_d = -90°$), the position setpoint repeated the former sinusoidal pattern. The estimated and tracking results are compared with ground truth distance and orientation in Fig. 4a. The distances from the wall to individual propellers (y_i's) are shown in Supplementary Fig. 2.

It can be seen that, regardless of the robot's yaw orientation (wall direction), the estimated distance and yaw angle (Fig. 4a) remained accurate when the robot was within 20 cm from the wall (or ≈10 cm from the wall to the nearest propeller). During this period, the shortest distance between the tip of the robot to the wall was ~2 cm. The root mean square errors (RMSEs) of $\hat{y}$ and $\hat{\psi}$, over a 60-s period ($t = 10$ s to $t = 70$ s), were 1.8 cm and 13.6°. When it comes to the flight performance, the position and yaw errors were also minor for collision avoidance purpose: 1.7 cm and 0.4°. Together, the results manifest the reliability of the proposed aerodynamic model, devised estimator, and the robustness of the implemented INDI controller.

To highlight the advantage of our aerodynamics-based method over the ToF—a lightweight emissive device for detecting nearby obstacles with high accuracy and low cost, we installed a ToF sensor (Crazyflie, Multi-ranger deck) on the side of the robot to measure the wall distance in the direction perpendicular to the robot's heading. Since the ToF sensor is directional and vulnerable to reflective or transparent surfaces, we compared three types of wall surfaces in this experiment: wooden board, transparent plexiglass, and mirror.

The previous flight trajectory was repeated with the three surfaces. The ground truth distance and raw ToF measurements from these flights are plotted in Supplementary Fig. 3 (yaw angle omitted). As anticipated, the ToF sensor takes an accurate

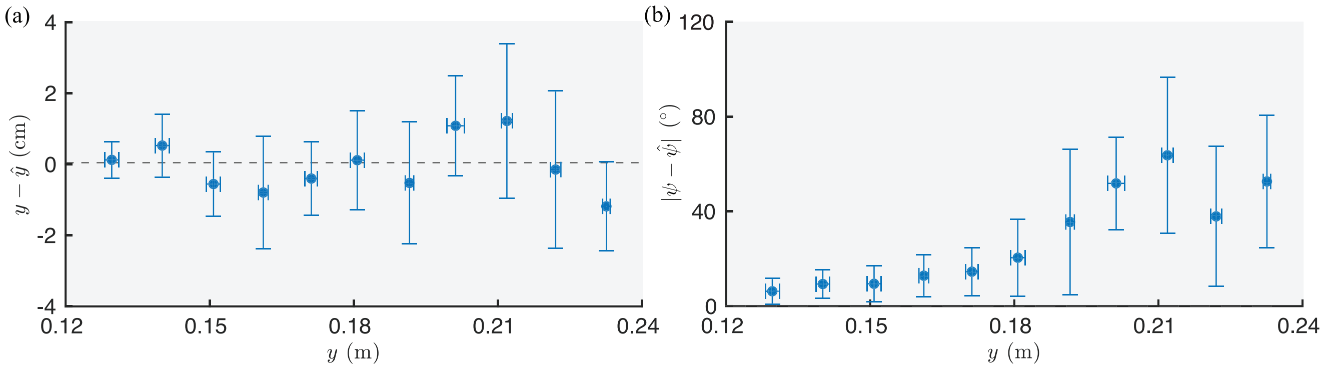

Fig. 5 Estimation errors versus distance from the wall. Error bars indicate the range of one standard deviation. **a** Mean estimation errors of distance are plotted against the average distance from the wall. **b** Mean estimation errors of orientation are plotted against the average distance from the wall.

measurement with a wooden surface. Despite of this, the output was not usable when the robot turned away from the surface owing to the directionality. It is not surprising that the emissive sensor cannot correctly detect a plexiglass sheet or a mirror as the beam is deflected or reflected. These limitations make the proposed aerodynamics-based strategy a more reliable solution in particular circumstances.

Then, we conducted several flights to investigate the relationship between estimation performance and distance. During each flight, the robot remained in place with $\psi_d = 0°$ for 60 seconds. The hovering flight was repeated 11 times with different position setpoints, ranging from 13 to 23 cm. The recorded trajectories are shown in Supplementary Figs. 4–7 and the statistics are displayed in Fig. 5. The results indicate that the distance errors were mostly below 2 cm when the robot was within 20 cm of the wall. As shown in Fig. 5b, relatively small angle errors were also observed when the robot was within this proximity to the wall (absolute errors, instead of signed errors, are shown due to the anticipated symmetry when $\psi_d = 0°$). We believe the errors at large distances were due to the weakened proximity effect. For instance, a distance of 0.2 m from the wall corresponds to $\overline{d} = 2.1$ for the nearest propeller when $\psi_d = 0°$ (Fig. 2f). This limitation in the sensing range affects the reliability of the estimates beyond 20 cm from the wall.

Flights with distance estimation and control

The control autonomy is demonstrated by directly using the estimated state for controlling the distance between the robot and the wall. In this flight (Movie 2), the robot was commanded to a start point with MoCap feedback, and the onboard position controller was activated at $t = 5$ s. However, yaw angle (ψ) and position feedback from the MoCap in the direction parallel to the wall (x direction) were still used in order to restrict the robot to the flight area. The rest of the reference trajectory is identical to the flight in Fig. 4a.

We first inspect the estimator's performance. As detailed in Fig. 4b and Supplementary Fig. 8, the error of the position estimate (the difference between $\hat{y}$ and y) increased noticeably to more than 3 cm when the actual y is over 20 cm in a few occasions. This was due to the diminished wall effect as illustrated in Fig. 5. The estimation errors stayed below 2–3 cm otherwise. Simultaneously, the yaw estimate generally stayed within 20° to the actual yaw angle ψ. Since the yaw estimation $\hat{\psi}$ corresponds to the wall direction and it was not used for control as indicated by (35) (Methods), ψ and $\hat{\psi}$ do not coincide.

To assess the controller's effectiveness, it can be seen that the estimated distance closely follows the references (the error was mostly below 2 cm). Overall, the results validate that the combined controller and estimator are highly reliable when both are implemented simultaneously.

Reactive navigation with aerodynamics-based collision avoidance

We propose two simple navigation methods that leverage the wall detection capability. The first method is inspired by bug algorithms[49], which use wall-following behavior to reactively avoid obstacles and explore the environment or reach the goal. The method consists of a high-level controller that steers the robot toward the destination. If an obstacle is detected, the robot maintains a safe distance and follows the surface tangentially until it can resume its original direction. The second method is based on attractive and repulsive vector fields[50,51]. The robot is attracted to the goal while repelled from any detected surfaces. This method enables the robot to take more direct paths to its destination, but it requires marginally more complex computation than the first approach.

An overview of the wall-following navigation method is depicted in Fig. 3b. The high-level controller makes the decision whether the robot is in free-flight or wall-following mode and it determines the position setpoint ($\boldsymbol{p}_d$ and higher order derivatives) for the INDI controller accordingly. To do so, the EKF estimator constantly performs wall detection. We adopt the moving average of the estimated distance as an indicator, defined as $\overline{y}(t) = (1/\Delta t)\int_{t-\Delta t}^{t}\hat{y}(\tau)\mathrm{d}\tau$ with Δt being the time window. The use of a moving average of $\hat{y}$ instead of $\hat{y}$ directly prevents the controller from erratically switching between the two navigation modes. When $\overline{y}$ falls below the threshold, the robot is instructed to enter the wall following mode by keeping a safe, constant separation $d^* < d^\dagger$ from the surface. This is by commanding the setpoint $\boldsymbol{e}_2^T\boldsymbol{p}_d = d^*, \boldsymbol{e}_2^T\dot{\boldsymbol{p}}_d = 0$ (be reminded that, by our definition, the orientation of the inertial frame $\mathcal{W}$ is with respect to the detected wall and this may change during the flight). Meanwhile, the speed parallel to the wall is controlled to be constant v^*, with the direction towards the ultimate goal or $\boldsymbol{e}_1^T\dot{\boldsymbol{p}}_d = v^*$. From the vehicle's perspective, this means the robot keeps a fixed distance in the direction of the estimated surface (described by $\hat{\psi}$) and a constant speed in the perpendicular direction.

When no surface is detected nearby (estimated wall distance $\overline{y}$ is higher than the threshold $d^\dagger$), the robot is commanded to fly towards the destination or desired direction. In the experiments, we present two feasible schemes for the free-flight mode. The first is when position feedback is available (GPS or MoCap measurements). In this case, the position error $\boldsymbol{p} - \boldsymbol{p}_d$ and their time derivatives required by the controller (Eq. (26)) are directly available. Second, in indoor settings with no positioning, a pilot may prescribe the flight direction when there is no collision risk (

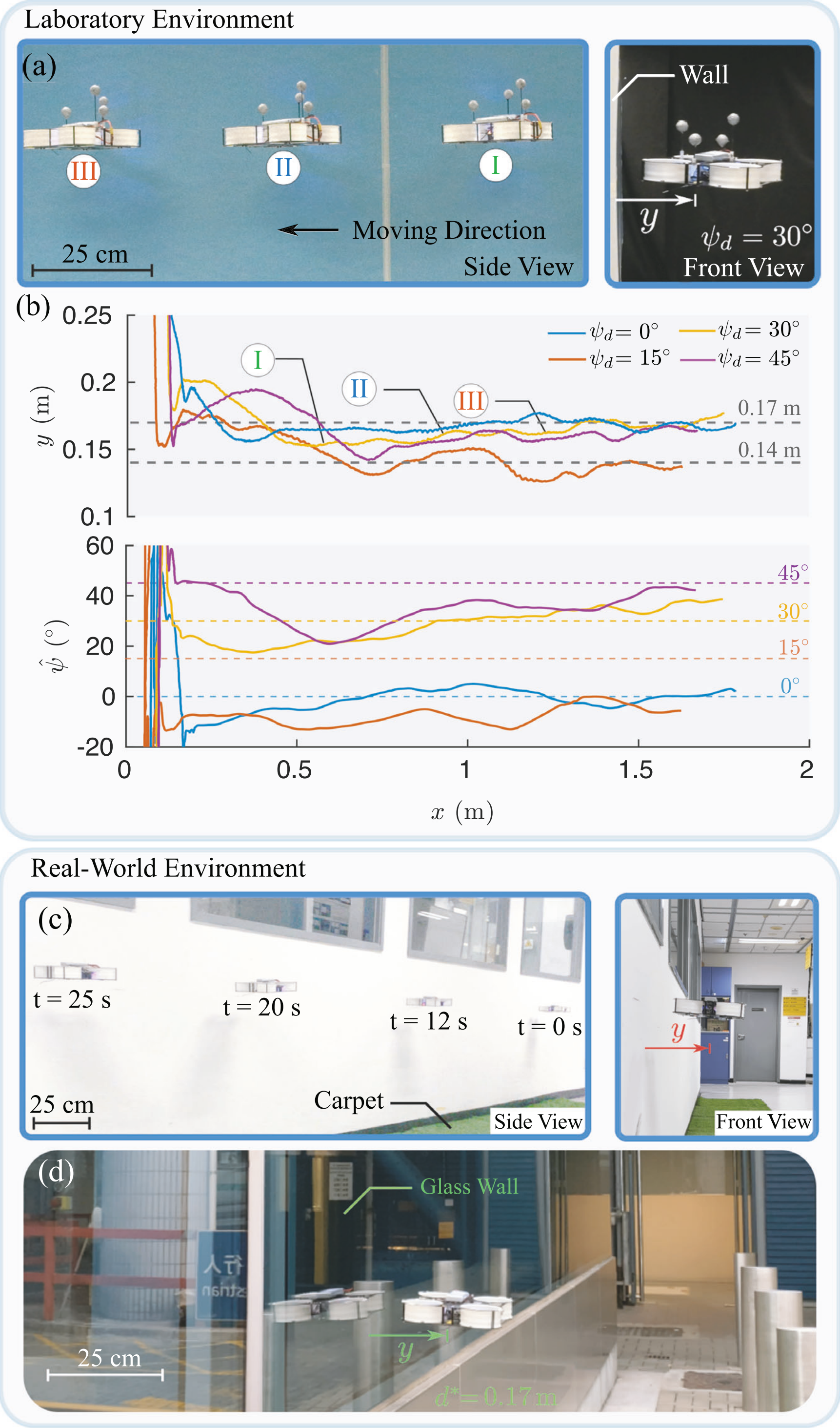

Fig. 6 Wall-tracking maneuvers in both laboratory and real-world environments. a Lateral and frontal photos showing the robot traveling parallel to a vertical wall with onboard feedback when $\psi = 30°$. **b** Trajectories of wall-tracking flights when ψ (wall directions) takes different values. In each flight, the robot approached the wall and traveled along the wall by about 1.8 m, maintaining a constant distance of $d^* = 0.17$ m to the wall. **c** Lateral and frontal photos showing the robot flying parallel to a wall along a corridor with onboard feedback. The robot traveled from right to left by about 2.5 m in 25 s. **d** Photo of an outdoor wall-tracking flight along a transparent and reflective surface.

$\bar{y} < d^{\dagger}$). In this situation, the velocity setpoint $\dot{\boldsymbol{p}}_d$ is chosen and other controller gains ($\boldsymbol{K}_p$ and $\boldsymbol{K}_i$) in (26) are set to zero.

The vector field-based method for collision avoidance is outlined in Fig. 3b. In this case, the setpoint velocity of the INDI controller $\dot{\boldsymbol{p}}_d$ is directly provided as a vector sum of two components, $\boldsymbol{v}_+(\hat{y})$ and $\boldsymbol{v}_-(\hat{y})$, directing the robot towards the goal while repelling it from obstacles.

To pull the robot to the goal, the attractive field is defined as a uniform velocity vector with $|\boldsymbol{v}_+(\hat{y})| = v_+$ and the direction of $\boldsymbol{v}_+$ points to the goal. The third element of $\boldsymbol{v}_+$ is zero for maintaining a constant altitude. We chose $v_{+} = 10\ \mathrm{cm \cdot s^{-1}}$. This becomes the setpoint speed of the robot when it is far from obstacles.

The repulsive field keeps the robot away from a detected obstacle based on the current estimates $\{\hat{y}, \hat{\psi}\}$. The magnitude of $|\boldsymbol{v}_{-}(\hat{y})| = v_{-} = (9\times 10^{5}\ \mathrm{m \cdot s^{-1}})(1.8\times 10^{-35})^{\hat{y}}$ was experimentally tuned such that v_{-} converges to a very high value, $9 \times 10^{5}\ \mathrm{m \cdot s^{-1}}$, as $\hat{y} \to 0$ and diminishes exponentially as $\hat{y} \gg 1$. The vector $\boldsymbol{v}_{-}$ is normal to the surface direction. That is,

$$\boldsymbol{v}_{-}(\hat{y}) = \begin{bmatrix} v_{-}(\hat{y}) \sin\hat{\psi} & v_{-}(\hat{y}) \cos\hat{\psi} & 0 \end{bmatrix}^{T}, \tag{12}$$

such that $\dot{\boldsymbol{p}}_{d} = \boldsymbol{v}_{+} + \boldsymbol{v}_{-}$ is dynamically evaluated online based on the current estimates of an obstacle. In the meantime, the proportional (and integral) component of the INDI position controller in (26) is modified and saturated so that it only

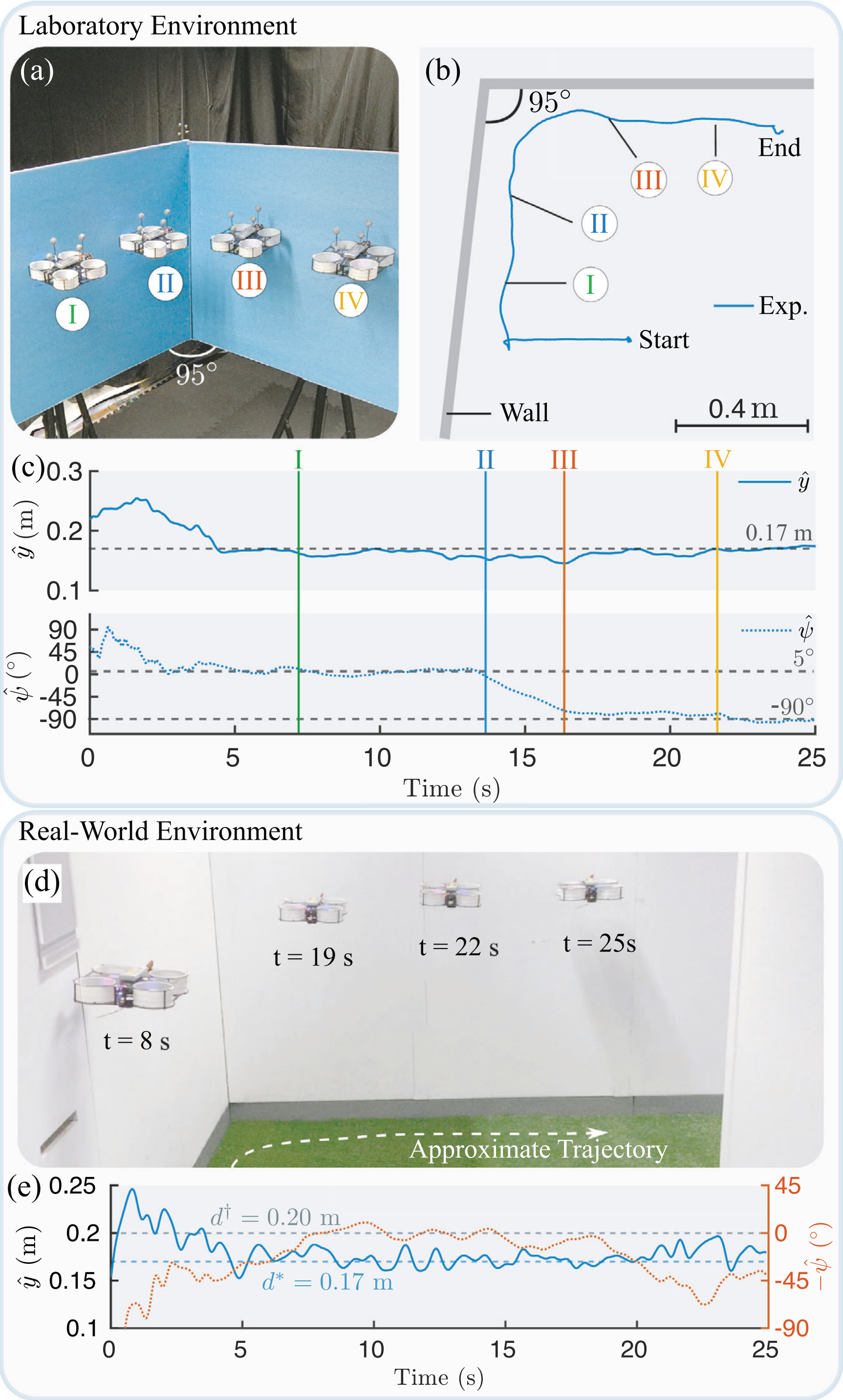

Fig. 7 Wall tracking flights with a corner in both laboratory and real-world environments. **a** Composite image depicting the robot negotiating a corner in a wall-tracking flight. **b** The trajectory of the flight in (**a**). **c** Onboard estimation results of the flight in (**a**). **d** Composite photo showing the wall-tracking flight along a corner in a corridor. **e** Onboard estimation results of the flight in (**d**).

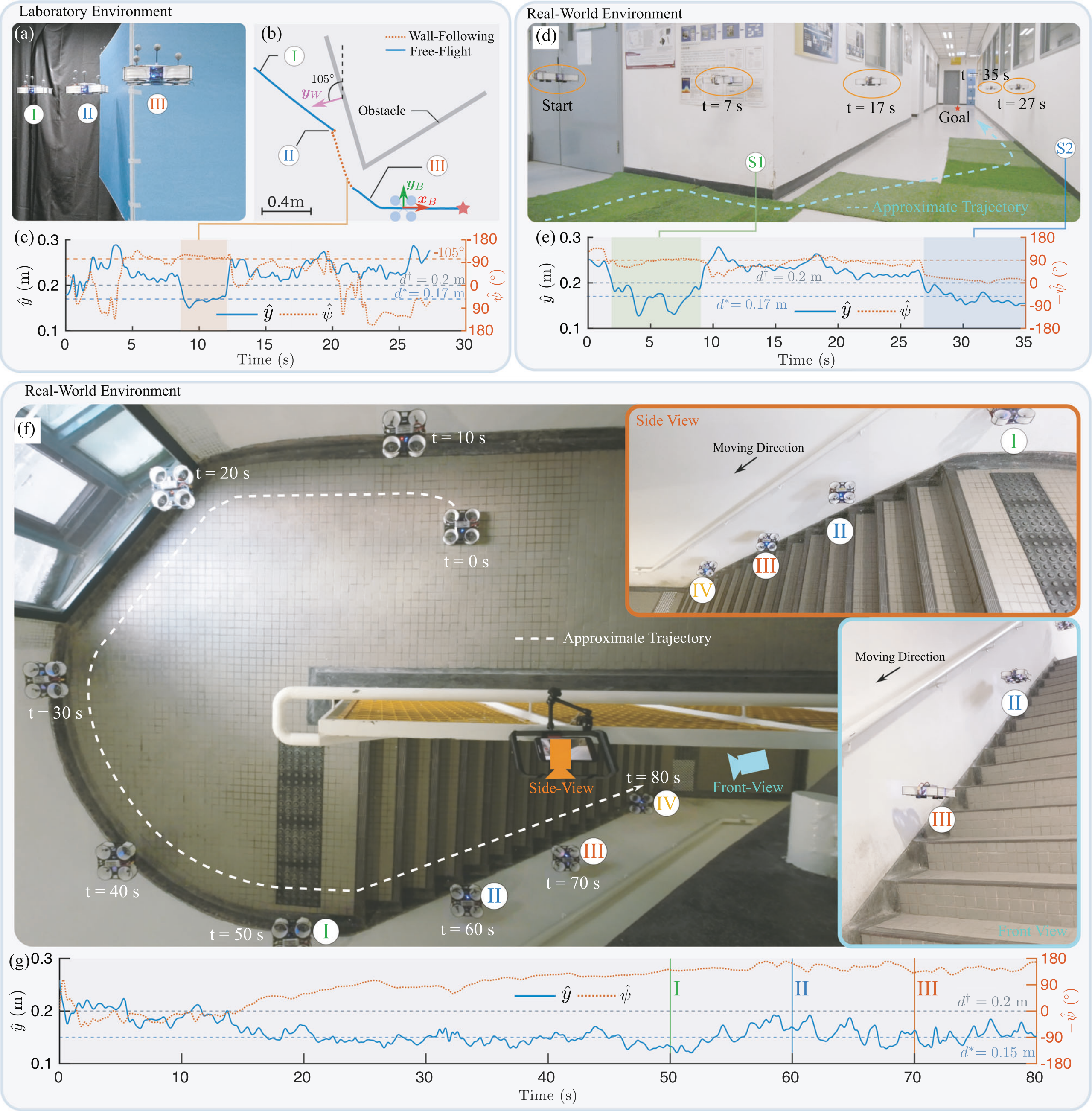

Fig. 8 Collision avoidance and exploring flights in laboratory and real-world environments. a A short flight of the robot bypassing an obstacle to reach the goal in a laboratory. **b** The trajectory of the flight in (**a**). **c** Plots of the estimated wall distance and direction, covering the periods of free-flight and wall-following modes. **d** Photo of a collision-free flight in the corridor. When encountering a wall, the robot traveled along the contour. **e** Plots of the estimated wall distance and direction showing two wall-following sections. **f** A composite image displaying the robot exploring a staircase by wall tracking. The robot took off on the upper floor and followed the wall to safely travel downstairs. **g** Plots of the estimated wall distance and direction illustrating the changing surface direction (relative to the robot's body frame).

proportionally repulses the robot away from the surface when $\hat{y} < d^* = 19$ cm. Unlike the wall-following approach earlier, there is no force directly attracting the robot to the surface. Neglecting the proportional component, the expected trajectory can then be deduced from the vector field: $\dot{\boldsymbol{p}}_d = \boldsymbol{v}_+ + \boldsymbol{v}_-$. The critical distance when $|\boldsymbol{v}_+(\hat{y})| = |\boldsymbol{v}_-(\hat{y})|$ is 17.1 cm. This threshold approximately determines whether the attractive or repulsive field dominates.

Collision avoidance flight

We experimentally apply the proposed collision avoidance and navigation strategy in steps. This begins with the wall following components in both indoor and outdoor settings. After that, a high-level decision-making controller is added to realize wall-following navigation. As suggested by the results in Fig. 4d, we set the time window Δt for $\bar{y}$ as 1.5 s, the decision threshold as $d^\dagger = 20$ cm, and the wall tracking distance of $d^* = 17$ cm. This ensures the robot remains sufficiently close to the wall to yield reliable distance estimates ($\overline{d}$<2.1). The wall tracking speed v^* is either 10 or 25 cm · s^{-1} depending on the experiments. The robot's heading angle was kept constant by choosing $\psi_d = 0°$. We then present a real-world example entailing a robot tracking a wall and traversing a staircase without vision. Lastly, more collision-free flights are

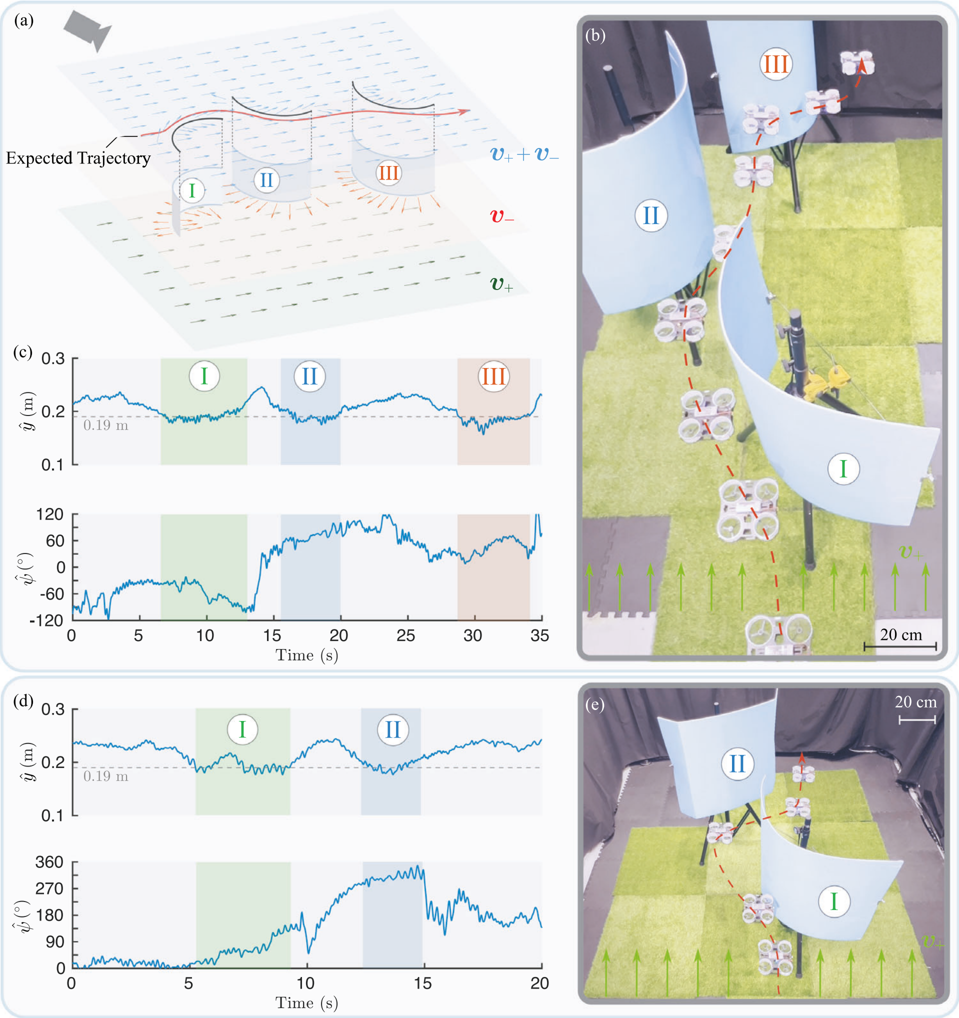

Fig. 9 Field-based collision avoidance flights in cluttered environments. **a** A drawing illustrating the composition of the velocity field (not-to-scale). The resultant field is an interplay between the attractive and repulsive fields. An example of an expected trajectory is shown. **b** A composite image displaying the robot evading three obstacles in the path. **c** Plots of the estimated wall distance and direction during the flight in (**b**). **d** Plots of the estimated wall distance and direction during the flight in (**e**). **e** A composite image showing the robot flying through rounded and flat surface obstacles.

demonstrated with a field-based algorithm in complex scenes with rounded and flat obstacles.

We first verify the wall tracking strategy (Movie 3), employing a sheet of foam board (2.4 × 0.9 m) as an artificial wall. The robot operated autonomously using only onboard feedback and the MoCap was only for ground truth measurements except for the beginning. In the experiment, the robot took off with ~50 cm gap from the wall and maintained an altitude of 1.0 m with the yaw angle $\psi_d = 0°$ by initially relying on the MoCap feedback. It then approached the wall with a constant velocity of 10 $\mathrm{cm \cdot s^{-1}}$ and transitioned to the wall-tracking mode and fully leveraged the EKF estimates when the surface was detected or $\bar{y} < d^{\dagger} = 20$ cm.

As plotted in Fig. 6b, in the wall following regime, the robot glided along the wall for ≈1.5 m at the setpoint speed $v^* = 10$ $\mathrm{cm \cdot s^{-1}}$, retaining the distance of 17 cm (a 5 cm gap between the ducted propeller and wall) using only the onboard feedback. During this period, the RMS position control error $y - y_d$ is 0.4 cm. To inspect the influence of ψ_d on the tracking performance, we repeated the same experiment with $\psi_d = 15°, 30°$, and 45°. The results, presented in Fig. 6a and b, corroborate that the wall direction does not significantly affect the behavior unless the error in the angle estimate is prominent. The RMS errors of the wall distance are 3.2 cm for $\psi_d = 15°$, 1.0 cm for $\psi_d = 30°$, and 1.3 cm for $\psi_d = 45°$. For the case with $\psi_d = 15°$ and $\psi_d = 45°$, the higher errors of 3.2 cm and 1.3 cm are likely associated with the inaccurate angle estimate.

Then, the strategy was applied in the absence of the MoCap system in both indoor and outdoor environments. For indoor conditions, the flight was carried out in the corridor of a building. The robot took off in the middle of the corridor (~2 m wide) and maintained an altitude of 0.8 m using its ventral ToF sensor. It was then remotely commanded to fly toward a wall by a user. After the wall was detected, the robot avoided a collision and followed the wall contour over 6 m in 27 s ($v^* = 25$ $\mathrm{cm \cdot s^{-1}}$, $\psi_d = 0°$) as shown in Fig. 6c. A similar flight was replicated outdoors at a building exterior with a glass (transparent) wall (Fig. 6d). The robot traveled safely along the surface over 2.5 m in 18 s. Together, the outcomes confirm that the distance estimation and control method stays sufficiently robust when the robot moves along the surface. Unlike emissive sensors (ToF), the aerodynamics-based approach is non-directional and material-independent.

Next, we investigate the reaction of the robot when it encounters two surfaces intersecting with an obtuse angle. Since the proximity effect model and the estimation method assume the presence of only one wall, it remains unclear how the estimator and the robot would react when it flies into such corner.

The test was first conducted in the flight arena with two surfaces formed by 120 × 90-cm foam sheets making a 95° corner (approximately perpendicular) as seen in Fig. 7a–c. Similarly, the robot initiated the flight with MoCap feedback, and it traveled along the first wall with $v^* = 10$ cm · s^{-1} at an altitude of 1.0 m using the EKF estimates, with other operating conditions identical to previous experiments. As the vehicle neared the second surface, the estimated $\hat{\psi}$ gradually shifted (Fig. 7c) from 0° to −90°. Thanks to the closed-loop distance control, the estimated distance $\hat{y}$ did not deviate much from the setpoint d^*. The robot kept a safe distance while it negotiated the corner as shown in Fig. 7b. The trend continued as the robot departed the first wall and followed the second wall.

When executed in a building corridor (Fig. 7d), the robot produced a similar reaction. It detected the corner as a gradual change in direction while staying away from both walls, resulting in a collision-free flight. We may conclude that practically the robot deals with a corner as a progressive change in the surface direction. This allows the robot to avoid a collision in such scenarios without explicitly modeling the aerodynamic forces caused by two intersecting walls.

Having demonstrated reliable wall-following maneuvers, we expand the flight to include a simple navigation component through the use of a high-level controller for the robot to avoid obstacles while traveling to the desired location or in the preferred direction. A series of experiments were conducted. This begins with a simulated situation in a MoCap-equipped lab. The robot was provided with real-time position feedback and instructed to head towards the goal that is ≈2.8 m away from the starting point. In between, we placed an obstacle with a convex corner formed by two surfaces as illustrated in Fig. 8a and b.

As visualized in Fig. 8b and c, the robot initially headed directly to the goal after liftoff while the onboard estimation algorithm was also active. When a surface was detected ($\bar{y} < d^{\dagger} = 20$ cm), the high-level controller enabled the wall-tracking mode using the estimated distance and the robot glided along the wall with $v^* = 25$ cm · s^{-1}. After passing through the corner, the estimated distance quickly grew beyond the threshold and the robot reverted back to free-flight mode. Shortly after, it safely reached its destination. The total flight time of 27 s includes 4 s spent in the wall-following mode.

A similar flight was enacted in an indoor corridor without MoCap feedback. As displayed in Fig. 8d and e, the robot attempted to reach the far end of the corridor based on the direction provided by a human pilot. However, it was partially blocked by a wall at the starting location. Without prior knowledge of the surrounding, a velocity command was sent to the robot to pilot it toward the desired position. When encountering the wall ($t = 3$ s), it entered the wall-tracking mode and circumvented it without crashing at the speed $v^* = 20$ cm · s^{-1}. However, without positioning or the Earth's magnetic field, the travel direction along the wall was manually provided by the human pilot. After the wall had disappeared, the robot returned to cruising and approximately maintained its travel direction until it faced another surface at $t = 27$ s. It reliably identified the wall and adjusted its travel direction accordingly before we terminated the flight at $t = 42$ s.

A more challenging navigation task involving an exploratory flight over a staircase is illustrated. In this instance, the robot tracked a curved wall before descending a flight of stairs. In this example, the vehicle followed the wall for an extended period without leaving the surface. As depicted in Fig. 8f, the robot took off on the upper floor and it was piloted to fly at a speed of 10 cm · s^{-1} to find a nearby surface. After identifying the wall, it initiated the wall-tracking process and retained the flight speed of 10 cm · s^{-1} with $d^* = 15$ cm. The vehicle subsequently stayed adjacent to the surface for over 70 s. In this period, it flew along a 1.2-m wide glass panel and gradually executed a complete U-turn (≈1.6 m in radius) in ~30 s. This is feasible as the estimated surface direction was constantly updated. As plotted in Fig. 8g, the estimated wall direction changed from 0° to 180°. At the end of the turn, the robot negotiated the stairs to arrive at the lower level. During the descent through steps, the downward-looking ToF sensor stabilized the perceived altitude, resulting in a relatively choppy motion. Despite this, the distance estimator and controller remained effective (Fig. 8g). This wall-following approach can be readily extended for exploring unknown space or even constructing a simple map as illustrated in ref. [17]. Therefore, this staircase experiment manifests the robustness of the strategy as well as the capability of the robot to autonomously explore a less structured environment in the absence of vision.

Finally, we realized collision-free flights in complex scenes with several isolated obstacles with the proposed field-based algorithm (Movie 4). Motivated by refs. [52,53], we artificially constructed cluttered environments with rounded (radius: 0.8 m, relatively large compared to the size of the robot to comply with the flat-surface assumption of the estimator) and flat obstacles (Fig. 9). Unlike previous examples, in which the estimator was implemented as a Python script on a ground computer for rapid deployment, we ported the EKF-based estimation scheme over to the onboard flight controller (Bitcraze, Crazyflie Bolt 1.1 powered by Cortex-M4, with the code written in C) to manifest the computational efficiency of the method.

In the first scenario, the constant attractive field induces the robot to fly north and negotiate around three rounded obstacles. Based on the environment, we can pre-construct the field map and predict the anticipated trajectory based on the starting location as visualized in Fig. 9a (without the knowledge of the surrounding, the robot is deprived of this information). In the actual flight (Fig. 9b), the repulsive field $\boldsymbol{v}_-$ was computed online based on the current wall estimates (Fig. 9c), permitting it to avoid a collision in the 35-s flight.

The second demonstration involves one rounded obstacle and one planar surface. Similarly, the robot stayed away from any detected surfaces with the minimum $\hat{y}$ in the 20-s flight of 19 cm (Fig. 9d). As illustrated in Fig. 9e, the flat surface re-directed the robot away from the goal direction. The repulsive field forced the robot to circumvent the obstacle before safely reaching the goal.

DISCUSSION

In this work, an aerodynamics-based collision avoidance strategy is proposed for a small ducted rotorcraft. This is accomplished by leveraging the flow interaction between spinning propellers and a surface. The propeller ducts are shown to significantly reinforce the proximity effect and the observation from the benchtop measurements is supported by the momentum theory analysis. The modeled change in vertical and horizontal force components enables us to incorporate the contribution of this surface-induced aerodynamic effect into the flight dynamics. Therefore, we develop a Kalman filter to estimate the distance and orientation of a nearby wall without any visual feedback. The robot solely relies on an IMU and a suite of ventral ToF and optic flow sensors to localize itself with respect to the wall. Following this, an INDI controller is devised for the robot to maneuver safely in the vicinity of a wall, taking advantage of the inherent robustness against modeling errors. With the ability to localize and regulate its position against a detected surface, we employ a simple decision-making rule for the robot to alternate between wall-tracking and free-flight maneuvers based on collision risk. Alternatively, a field-based path-planning method can also be used for obstacle avoidance. Thanks to these, the robot is able to navigate safely with reactive autonomy, using only the same set of sensors required for sensory-motor autonomy. If deployed with established bug algorithms[49] for complete wall-following behaviors, the robot has the potential to be used for exploring unknown environments in swarms as demonstrated in refs. [17,52].

Together, the collision avoidance strategy was thoroughly verified via a series of demonstrations in both controlled and field environments, including an exploratory flight in a staircase. The results convincingly illustrate the reliability and effectiveness of the method. For other rotorcraft to employ the developed obstacle avoidance strategy, one must first construct and verify the near-wall thrust model. This is followed by integrating the obtained force and torque mapping into the dynamics for estimating and avoiding collisions. However, wider adoption of this method is subject to addressing limitations and open questions that remain, such as the constraint on sensing range, the ability to deal with moving obstacles, and scaling of aerodynamic forces to different vehicle sizes. These outstanding issues are briefly discussed as folows.

As evidenced by the flight experiments, the effective wall sensing range is restricted to ~2.1 rotor radii (8 cm or less) from a surface to the nearest rotor. While the range is lower than conventional ToF sensors, the proximity sensing approach is insensitive to surface materials and can detect obstacles in all directions around the vehicle. Therefore, the two sensing modalities could be used complementarily to overcome their respective limitations.

The limited sensing range imposes practical limits on safe flight speed. With only 8 cm of detection range, the robot must have sufficient time to react upon detecting an obstacle in its path. Similarly, a fast-moving obstacle approaching a hovering robot may collide with the robot before it can detect and evade.

To estimate a safe speed limit based on the obstacle detection range of ~8 cm with 4 cm reserved as safety buffer, the robot flying at speed v should stop within 4 cm of detecting an obstacle. Assuming instant detection and no controller delay, decelerating at $-1\ \mathrm{m\cdot s^{-2}}$ (or ≈0.1g) would stop the robot in 4 cm if $v \leq 28$ $\mathrm{cm\cdot s^{-1}}$. In practice, accounting for response latency, an even lower safe speed would be prudent. In flight tests, a 25 $\mathrm{cm\cdot s^{-1}}$ speed yielded collision-free navigation. As outlined, this could be regarded as the speed limit of moving obstacles for a hovering robot.

In addition, it is foreseeable that including the impact of wind on flight dynamics and wind velocity as an unknown state variable could improve the reliability of the vision-less collision avoidance strategy in gusty environments. Furthermore, the recorded trajectory of the robot, in association with the detected surfaces, could be used for reconstructing a simple map when the robot explores an unknown environment similar to the approach in ref. [17].

Lastly, the impact of scaling on the wall effect is considered. The ground effect, which is another form of the proximity effect, has been observed in various vehicles and objects across multiple length scales. Even when using the normalized distance $\overline{d} = d/R$ for analysis, the degree of the effect can still vary with the characteristic length. In a previous study[34], it was found that the ceiling effect was more pronounced for a smaller propeller (with $R = 23$ cm) compared to a larger one (with $R = 50$ cm) after normalization.

To investigate the wall effect further, we conducted tests on two smaller ducted propellers with radii of 2.5 and 2 inches, as described in Supplementary Note 2. The empirical wall effect coefficients shown in Supplementary Fig. 9 suggest that the wall effect may be less pronounced for smaller propellers. However, this is a preliminary finding that warrants further investigation in future studies.

METHODS

Momentum theory for Wall effect

To apply MT, we let p_0 be the atmospheric pressure of the free-stream quiescent air. The propelling thrust is produced by the discontinuity in the pressures immediately above (p_-) and below (p_+) the effective propeller disk according to

$$T = (p_+ - p_-)A, \tag{13}$$

when $A = \pi R^2$ is the disk area. Provided that v_i is the speed of the uniform flow across the propeller disk, the upstream and downstream wakes along the streamline are characterized by Bernoulli equations as

$$p_0 = p_- + \frac{1}{2}\rho v_i^2, \tag{14}$$

$$p_+ + \frac{1}{2}\rho v_i^2 = p_0 + \frac{1}{2}\rho v_\infty^2, \tag{15}$$

where ρ is the air density and v_∞ is the terminal flow velocity far downstream. Combining (13)–(15) to eliminate p_+ and p_- leaves

$$T = \frac{1}{2}\rho A v_\infty^2. \tag{16}$$

To relate v_i to T, the conservation of momentum is taken. The flow rate through the propeller disk is given by $\rho A v_i$. Since the wake far upstream is quiescent and the terminal flow is described by v_∞ and θ_∞, the difference in the vertical momentum of the incoming and outgoing air is $\rho A v_i v_\infty \cos\theta_\infty$. Neglecting the skin drags, the difference in momentum is equated to the vertical component of the thrust as

$$\rho A v_i v_\infty \cos\theta_\infty = T\cos\theta_t. \tag{17}$$

Making use of (16) to eliminate v_∞ from (17) produces

$$v_i = \sqrt{\frac{T}{2\rho A}} \cdot \frac{\cos\theta_t}{\cos\theta_\infty}. \tag{18}$$

Therefore, we obtain the aerodynamic power of the spinning propeller in terms of T as

$$P_a = Tv_i = T\sqrt{\frac{T}{2\rho A}} \cdot \frac{\cos\theta_t}{\cos\theta_\infty}. \tag{19}$$

Inverting the relation of T and P_a results in (1).

Benchtop setup

A single 3-blade propeller with $R = 38$ mm (AVAN, 3024) and a brushless DC motor (EMAX, RS1306B 4000KV) were mounted on top of a multi-axis force/torque sensor (ATI, Nano 17) via a solid structure. The propeller was fitted with a 3D-printed duct that is $H = 35$ mm (0.92R) tall. The space between the propeller tips to the duct's inner surface was 0.5 mm (assumed negligible). As depicted in Fig. 2, the setup was installed above a linear motorized stage. This allows the distance between the ducted propeller and a wall (sheet of plywood, 50 × 130 cm) to be varied. The motor was controlled by an Electrical Speed Controller (Turnigy 20A) with a constant supplied voltage of 8.4 V. In the experiments, a computer running Simulink Real-Time Desktop (MathWorks) with the data acquisition unit (NI PCI-6229) was employed for synchronizing the driving commands and measurements. The current consumed by the ESC was recorded through a Hall effect current sensor (LEM, LTS25-NP). The motorized stage was controlled by a stepper motor (resolution: 25μm) for altering the distance d from 1 to 200 mm at the increment of 1 mm. At each distance, force, torque and current measurements were taken at the rate of 2 kHz. Four motor throttle commands between 40% and 70% were taken. For each data point, the measurement was taken using the average from a

10-s duration after allowing the system to reach a steady state for 20 s. Each point represents a particular throttle and distance.

EKF for State Estimation

In the continuous-time domain, elements in the state evolve according to $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{T})$, in which

$$\begin{aligned} \dot{\bar{\boldsymbol{p}}} &= [\boldsymbol{e}_2, \boldsymbol{e}_3]^T \dot{\boldsymbol{p}}, \\ \ddot{\boldsymbol{p}} &= m^{-1}(\boldsymbol{T}_W - mg\boldsymbol{e}_3), \\ \dot{\boldsymbol{R}} &= \boldsymbol{R}[\boldsymbol{\omega}]_\times, \\ \dot{\boldsymbol{\omega}} &= \boldsymbol{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}), \end{aligned} \tag{20}$$

with the expressions of $\ddot{\boldsymbol{p}}$ and $\dot{\boldsymbol{\omega}}$ follow the translational and rotational dynamics described by (6) and (8). The terms $\boldsymbol{T}_W$ and $\boldsymbol{\tau}$ therein implicitly include the thrust commands $\boldsymbol{T}_0$ and the wall effects (in the form of the wall distance and direction) as detailed earlier. In other words, $\boldsymbol{T}_w$ and τ are dependent on $\bar{\boldsymbol{p}}$ and $\boldsymbol{R}$.

Next, the continuous-time equations (20) are discretized using the forward Euler method to propagate the state:

$$\mathbf{x}_k^- = \mathbf{x}_{k-1}^+ + f(\mathbf{x}_{k-1}^+, \boldsymbol{T}_{k-1})\Delta t + \boldsymbol{w}_{k-1} \tag{21}$$

where k is the time index, Δt denotes a constant time step. The superscripts $(\cdot)^+$ and $(\cdot)^-$ indicate a-posteriori and a-priori estimates. $\boldsymbol{w}_{k-1}$'s is a vector of zero-mean Gaussian white noise representing the process noise, with the associated covariance matrix $\mathbf{Q}_{k-1}$.

Defining $\boldsymbol{P}$ as the covariance matrix of the estimated state, its propagation follows

$$\boldsymbol{P}_k^- = \boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{22}$$

where $\boldsymbol{A}_{k-1}$ is state Jacobian associated with (20).

The predicted state is updated based on the measurements from onboard sensors, including the accelerometer $\boldsymbol{a}_m$ and gyroscopic $\boldsymbol{w}_m$ readings, optical flow sensor output $\boldsymbol{n}_m$, and the ToF reading measurement z_m. Omitting the time index, the measurement model $\mathbf{y} = [\boldsymbol{a}_m^T, \boldsymbol{w}_m^T, \boldsymbol{n}_m^T, z_m]^T = \boldsymbol{h}(\mathbf{x}) + \boldsymbol{v}$ is

$$\begin{aligned} \boldsymbol{a}_m &= \boldsymbol{R}^T(\ddot{\boldsymbol{p}} - g\boldsymbol{e}_3) + \boldsymbol{v}_a \\ \boldsymbol{\omega}_m &= \boldsymbol{\omega} + \boldsymbol{v}_\omega \\ \boldsymbol{n}_m &= [n_x, n_y]^T + \boldsymbol{v}_n, \\ z_m &= z + v_z, \end{aligned} \tag{23}$$

where $\boldsymbol{v}$ is zero-mean Gaussian white noise with the covariance matrices $\mathbf{R}$. Letting $\boldsymbol{H}$ be the Jacobian taken from (23), the Kalman gain $\boldsymbol{K}$ for timestamp k is given by

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \mathbf{R}_k\right)^{-1}, \tag{24}$$

Lastly, we obtain the a-posteriori estimate of the state and its covariance at time step k as

$$\begin{aligned} \mathbf{x}_k^+ &= \mathbf{x}_k^- + \boldsymbol{K}_k\left(\mathbf{y}_k - \boldsymbol{h}(\mathbf{x}_k^-)\right), \\ \boldsymbol{P}_k^+ &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-. \end{aligned} \tag{25}$$

Together, (21), (22), and (25) allow us to use only the onboard feedback to estimate $\mathbf{x}$.

INDI controller

To stabilize the position error, $\boldsymbol{p} - \boldsymbol{p}_d$, a standard proportional-integral-derivative scheme is used to evaluate the target acceleration $\ddot{\boldsymbol{p}}_t$ that would minimize the position error $\boldsymbol{p} - \boldsymbol{p}_d$:

$$\begin{aligned} \ddot{\boldsymbol{p}}_t = \ddot{\boldsymbol{p}}_d &+ \boldsymbol{K}_d(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_d) \\ &+ \boldsymbol{K}_p(\boldsymbol{p} - \boldsymbol{p}_d) + \boldsymbol{K}_i\textstyle\int(\boldsymbol{p} - \boldsymbol{p}_d)\mathrm{d}t \end{aligned} \tag{26}$$

where $\boldsymbol{K}_{(\cdot)}$'s are positive diagonal gains. It can be seen that if $\ddot{\boldsymbol{p}} \to \ddot{\boldsymbol{p}}_t, \boldsymbol{p}$ would converge to $\boldsymbol{p}_d$. Next, to ensure that $\ddot{\boldsymbol{p}} \to \ddot{\boldsymbol{p}}_t$, an INDI framework below is employed to evaluate the desired angular rate $\omega_{d,x}, \omega_{d,y}$, and total thrust $T_{d,0} = \sum_{i=1}^4 T_{0,i}$ as setpoints for the inner controller loop. The INDI controller updates the values of $\omega_{d,x}, \omega_{d,y}$ and $T_{d,0}$ as long as the target acceleration $\ddot{p}_t$ is still unequal to the measured acceleration $\boldsymbol{Ra}_m + g\boldsymbol{e}_3$, rendering the method robust to the unmodeled wall effect and other disturbances.

To begin, we note that the translational dynamics in (6) is dictated by $\boldsymbol{T}_W$. When the proximity effects are neglected, $\boldsymbol{T}_W$ (derived from (5)) reduces to $T_0\boldsymbol{Re}_3$ with $T_0 = \sum_{i=1}^4 T_{0,i}$ being the total nominal thrust. Furthermore, the term $\boldsymbol{Re}_3$ is the unit vector of the $\boldsymbol{z}_B$ axis expressed in the inertial frame. Hence, $T_0\boldsymbol{Re}_3 = T_0\boldsymbol{z}_B$ can be collectively seen as an auxiliary input. Next, the INDI method focuses on the *increment* between the current t_k and previous t_{k-1} time steps of the dynamics (with $t_k - t_{k-1} = \Delta t$), leveraging the discrete-time implementation of the control and sensing instrument. We re-write (6) for two consecutive time steps:

$$m\ddot{\boldsymbol{p}}(t_k) + mg\boldsymbol{e}_3 = (T_0\boldsymbol{z}_B)(t_k), \tag{27}$$

$$m\ddot{\boldsymbol{p}}(t_{k-1}) + mg\boldsymbol{e}_3 = (T_0\boldsymbol{z}_B)(t_{k-1}). \tag{28}$$

The difference between the two equations are

$$m(\ddot{\boldsymbol{p}}(t_k) - \ddot{\boldsymbol{p}}(t_{k-1})) = \Delta(T_0\boldsymbol{z}_B), \tag{29}$$

where $\Delta(T_0\boldsymbol{z}_B) = (T_0\boldsymbol{z}_B)(t_k) - (T_0\boldsymbol{z}_B)(t_{k-1})$ is the change of the auxiliary input from the previous time step. The intermediate goal is to determine $\Delta(T_0\boldsymbol{z}_B)$ that makes the updated acceleration $\ddot{\boldsymbol{p}}(t_c)$ become the target $\ddot{\boldsymbol{p}}_t$ from (26). To accomplish that, it is assumed that $\ddot{\boldsymbol{p}}(t_p)$ can be measured. In this case, $\ddot{\boldsymbol{p}}(t_p)$ is computed from the specific acceleration and current attitude state provided by the IMU from (23) as $\ddot{\boldsymbol{p}}(t_p) = \boldsymbol{Ra}_m + g\boldsymbol{e}_3$. As a result, $\Delta(T_0\boldsymbol{z}_B)$ can be evaluated according to

$$\Delta(T_0\boldsymbol{z}_B) = m(\ddot{\boldsymbol{p}}_t - \boldsymbol{Ra}_m - g\boldsymbol{e}_3). \tag{30}$$

Lastly, to generate the setpoints for the attitude rate controller, we inspect the first-order expansion of $\Delta(T_0\boldsymbol{z}_B)$

$$\begin{aligned} \Delta(T_0\boldsymbol{z}_B) &= \tfrac{\mathrm{d}}{\mathrm{d}t}(T_0\boldsymbol{z}_B)\Delta t \\ &= \left(\omega_y\boldsymbol{x}_B - \omega_x\boldsymbol{y}_B\right)T_0\Delta t + \dot{T}_0\boldsymbol{z}_B\Delta t, \end{aligned} \tag{31}$$

where we have used the fact that $\dot{\boldsymbol{z}}_B = \omega_y\boldsymbol{x}_B - \omega_x\boldsymbol{y}_B$. With three perpendicular vector elements in (31), three setpoints $\{\omega_x, \omega_y, T_0\}$ for t_k can be computed by substituting in $\Delta(T_0\boldsymbol{z}_B)$ from (30):

$$\omega_x(t_k) = -\frac{m}{T_0(t_{k-1})\Delta t}\boldsymbol{y}_B^T(t_{k-1})(\ddot{\boldsymbol{p}}_t - \boldsymbol{Ra}_m - g\boldsymbol{e}_3), \tag{32}$$

$$\omega_y(t_k) = \frac{m}{T_0(t_{k-1})\Delta t}\boldsymbol{x}_B^T(t_{k-1})(\ddot{\boldsymbol{p}}_t - \boldsymbol{Ra}_m - g\boldsymbol{e}_3), \tag{33}$$

$$T_0(t_k) = T_0(t_{k-1}) + m\boldsymbol{z}_B^T(t_{k-1})(\ddot{\boldsymbol{p}}_t - \boldsymbol{Ra}_m - g\boldsymbol{e}_3). \tag{34}$$

Observe that the outcomes are intrinsically robust to modeling errors caused by neglecting the proximity effects as the setpoints produced by (32)–(34) would be updated from the previous time step as long as the target acceleration $\ddot{p}_t$ is unequal to the measured acceleration $\boldsymbol{Ra}_m + g\boldsymbol{e}_3$. This is the consequence of the *incremental* and sensor-based approach.

Since $\{\omega_x(t_k), \omega_y(t_k), T_0(t_k)\}$ serve as setpoints for the inner controller loop, they are subsequently be referred to as $\{\omega_{d,x}, \omega_{d,y}, T_{d,0}\}$.

Inner attitude rate loop

Considering the attitude dynamics, the task is to realize the desired roll and pitch rate $\omega_{d,x}, \omega_{d,y}$ obtained from (32) and (33). For the yaw axis, the angular error from the yaw setpoint ψ_d can be quantified as $\psi - \psi_d$. The yaw angle is calculated by projecting $\boldsymbol{x}_B$ on to the horizontal plane as $\psi = \operatorname{atan2}(\boldsymbol{e}_2^T\boldsymbol{x}_B, \boldsymbol{e}_1^T\boldsymbol{x}_B)$. Then, the rate and yaw errors are simultaneously stabilized if the angular acceleration becomes $\dot{\boldsymbol{\omega}}_d$ in the following form:

$$\dot{\boldsymbol{\omega}}_d = -\boldsymbol{\kappa}_d \begin{bmatrix} \omega_x - \omega_{d,x} \\ \omega_y - \omega_{d,y} \\ \omega_z \end{bmatrix} - \boldsymbol{\kappa}_p \begin{bmatrix} 0 \\ 0 \\ \int \omega_z dt - \psi_d \end{bmatrix} \tag{35}$$

where $\boldsymbol{\kappa}_d$ and $\boldsymbol{\kappa}_p$ are diagonal gain matrices. Observe that the proportional error term $\int \omega_z dt$ can be interpreted as the yaw angle relative to the start of the flight. When the gyroscope drift is insignificant, the robot maintains its absolute (i.e., not with respect to the wall) heading.

Finally, to determine the nominal thrust commands in the form of $\boldsymbol{T}_0$, we consolidate (35) with the attitude dynamics from (8). Then, the setpoint for the total nominal thrust from (34) is incorporated. Together, we yield

$$\boldsymbol{T}_0 = \begin{bmatrix} \boldsymbol{B} \\ 1111 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{J}\dot{\boldsymbol{\omega}}_d + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} \\ T_{d,0} \end{bmatrix}. \tag{36}$$

The result fully determines the thrust commands. Unlike the position controller, the influence of the wall is not neglected but integrated into $\boldsymbol{B}$.

DATA AVAILABILITY

All data generated or analyzed during this study are either included in this published article and its supplementary information files or available from the corresponding author on reasonable request.

Received: 7 January 2023; Accepted: 7 August 2023;
Published online: 26 October 2023

REFERENCES

1. Zhang, K. et al. Aerial additive manufacturing with multiple autonomous robots. *Nature* **609**, 709 (2022).
2. Xie, H., Dong, K. & Chirarattananon, P. Cooperative transport of a suspended payload via two aerial robots with inertial sensing. *IEEE Access* **10**, 81764 (2022).
3. Shah, K., Ballard, G., Schmidt, A. & Schwager, M. Multidrone aerial surveys of penguin colonies in antarctica. *Sci. Robot.* **5**, eabc3000 (2020).
4. Mansouri, S. S., Kanellakis, C., Kominiak, D. & Nikolakopoulos, G. Deploying mavs for autonomous navigation in dark underground mine environments. *Robot. Auton. Syst.* **126**, 103472 (2020).
5. Floreano, D. & Wood, R. J. Science, technology and the future of small autonomous drones. *Nature* **521**, 460 (2015).
6. Zhou, X. et al. Swarm of micro flying robots in the wild. *Sci. Robot.* **7**, eabm5954 (2022).
7. Foehn, P. et al. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Sci. Robot.* **7**, eabl6259 (2022).
8. Sanket, N. J. Active vision based embodied-ai design for nano-uav autonomy, Ph.D. thesis, Univ. Maryland, Coll. Park (2021).
9. Sanket, N. J., Singh, C. D., Fermüller, C. & Aloimonos, Y. Prgflow: unified swap-aware deep global optical flow for aerial robot navigation. *Electron. Lett.* **57**, 614 (2021).
10. Graule, M. A. et al. Perching and takeoff of a robotic insect on overhangs using switchable electrostatic adhesion. *Science* **352**, 978 (2016).
11. FarrellHelbling, E. & Wood, R. J. A review of propulsion, power, and control architectures for insect-scale flapping-wing vehicles. *Appl. Mech. Rev.* **70**, 010801 (2018).
12. Jafferis, N. T., Helbling, E. F., Karpelson, M. & Wood, R. J. Untethered flight of an insect-sized flapping-wing microscale aerial vehicle. *Nature* **570**, 491 (2019).
13. Chen, Y., Xu, S., Ren, Z. & Chirarattananon, P. Collision resilient insect-scale soft-actuated aerial robots with high agility. *IEEE Trans. Robot.* **37**, 1752 (2021).
14. Fuller, S., Yu, Z. & Talwekar, Y. P. A gyroscope-free visual-inertial flight control and wind sensing system for 10-mg robots. *Sci. Robot.* **7**, eabq8184 (2022).
15. Bai, S., He, Q. & Chirarattananon, P. A bioinspired revolving-wing drone with passive attitude stability and efficient hovering flight. *Sci. Robot.* **7**, eabg5913 (2022).
16. Healy, K., McNally, L., Ruxton, G. D., Cooper, N. & Jackson, A. L. Metabolic rate and body size are linked with perception of temporal information. *Anim. Behav.* **86**, 685 (2013).
17. McGuire, K., DeWagter, C., Tuyls, K., Kappen, H. & de Croon, G. C. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Sci. Robot.* **4**, eaaw9710 (2019).
18. Tan, S., Zhong, S. & Chirarattananon, P. A one-step visual-inertial ego-motion estimation using photometric feedback. *IEEE/ASME Trans. Mechatron.* **27**, 12–23 (2022).
19. Aloimonos, J., Weiss, I. & Bandyopadhyay, A. Active vision. *Int. J. Comput. Vision* **1**, 333 (1988).
20. Bajcsy, R., Aloimonos, Y. & Tsotsos, J. K. Revisiting active perception. *Auton. Robots* **42**, 177 (2018).
21. Escobar-Alvarez, H. D., Ohradzansky, M., Keshavan, J., Ranganathan, B. N. & Humbert, J. S. Bioinspired approaches for autonomous small-object detection and avoidance. *IEEE Trans. Robot.* **35**, 1220 (2019).
22. Xiao, F., Zheng, P., di Tria, J., Kocer, B. B. & Kovac, M. Optic flow-based reactive collision prevention for mavs using the fictitious obstacle hypothesis. *IEEE Robot. Autom. Lett.* **6**, 3144 (2021).
23. Sanket, N. J., Singh, C. D., Ganguly, K., Fermüller, C. & Aloimonos, Y. Gapflyt: active vision based minimalist structure-less gap detection for quadrotor flight. *IEEE Robot. Autom. Lett.* **3**, 2799 (2018).
24. de Croon, G. C. Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy. *Bioinspir. Biomim.* **11**, 016004 (2016).
25. de Croon, G. C., Dupeyroux, J., Fuller, S. & Marshall, J. Insect-inspired ai for autonomous robots. *Sci. Robot.* **7**, eabl6334 (2022).
26. Dümbgen, F., Hoffet, A., Kolundžija, M., Scholefield, A. & Vetterli, M. Blind as a bat: Audible echolocation on small robots. *IEEE Robot. Autom. Lett.* **8**, 1271–1278 (2023).
27. Nakata, T. et al. Aerodynamic imaging by mosquitoes inspires a surface detector for autonomous flying vehicles. *Science* **368**, 634 (2020).
28. Britcher, V. & Bergbreiter, S. Use of a MEMS differential pressure sensor to detect ground, ceiling, and walls on small quadrotors. *IEEE Robot. Autom. Lett.* **6**, 4568 (2021).
29. Davis, E. & Pounds, P. E. I. Passive position control of a quadrotor with ground effect interaction. *IEEE Robot. Autom. Lett.* **1**, 539 (2016).
30. Kan, X. et al. Analysis of ground effect for small-scale uavs in forward flight. *IEEE Robot. Autom. Lett.* **4**, 3860 (2019).
31. Conyers, S. A., Rutherford, M. J. & Valavanis, K. P. An empirical evaluation of ground effect for small-scale rotorcraft. *IEEE Int. Conf. Robot. Autom*. 1244–1250 (IEEE, 2018).
32. Jimenez-Cano, A. E., Sanchez-Cuevas, P. J., Grau, P., Ollero, A. & Heredia, G. Contact-based bridge inspection multirotors: design, modeling, and control considering the ceiling effect. *IEEE Robot. Autom. Lett.* **4**, 3561 (2019).
33. Hsiao, Y. H. & Chirarattananon, P. Ceiling effects for surface locomotion of small rotorcraft. *IEEE/RSJ Int. Conf. Intell. Rob. Syst. (IROS)* 6214–6219 (IEEE, 2018).
34. Hsiao, Y. H. & Chirarattananon, P. Ceiling effects for hybrid aerial–surface locomotion of small rotorcraft. *IEEE/ASME Trans. Mechatron.* **24**, 2316 (2019).
35. Conyers, S. A., Rutherford, M. J. & Valavanis, K. P. An empirical evaluation of ceiling effect for small-scale rotorcraft. *Int. Conf. Unmanned Aircraft Syst. (ICUAS)* 243–249 (IEEE, 2018).
36. Kocer, B. B., Tiryaki, M. E., Pratama, M., Tjahjowidodo, T. & Seet, G. G. L. Aerial robot control in close proximity to ceiling: A force estimation-based nonlinear mpc. *IEEE/RSJ Int. Conf. Intell. Rob. Syst. (IROS)* 2813–2819 (IEEE, 2019).
37. Wang, L., Zhou, B., Liu, C. & Shen, S. Estimation and adaption of indoor ego airflow disturbance with application to quadrotor trajectory planning. *IEEE Int. Conf. Rob. Autom. (ICRA)* 384–390 (IEEE, 2021).
38. Ding, R., Hsiao, Y. H., Jia, H., Bai, S. & Chirarattananon, P. Passive wall tracking for a rotorcraft with tilted and ducted propellers using proximity effects. *IEEE Robot. Autom. Lett.* **7**, 1581 (2022).
39. Wang, L., Xu, H., Zhang, Y. & Shen, S. Neither fast nor slow: How to fly through narrow tunnels. *IEEE Robot. Autom. Lett.* **7**, 5489 (2022).
40. Conyers, S. A. Empirical evaluation of ground, ceiling, and wall effect for small-scale rotorcraft, Ph.D. thesis, University of Denver (2019).
41. McKinnon, C. D. & Schoellig, A. P. Estimating and reacting to forces and torques resulting from common aerodynamic disturbances acting on quadrotors. *Robot. Auton. Syst.* **123**, 103314 (2020).
42. Carter, D. J., Bouchard, L. & Quinn, D. B. Influence of the ground, ceiling, and sidewall on micro-quadrotors. *AIAA J.* **59**, 1398 (2021).
43. Tu, Z., Fei, F., Zhang, J. & Deng, X. Acting is seeing: Navigating tight space using flapping wings. *IEEE Int. Conf. Rob. Autom. (ICRA)* 95–101 (IEEE, 2019).
44. Mulgaonkar, Y. et al. The tiercel: a novel autonomous micro aerial vehicle that can map the environment by flying into obstacles. *IEEE Int. Conf. Rob. Autom. (ICRA)* 7448–7454 (IEEE, 2020).
45. Deng, S., Wang, S. & Zhang, Z. Aerodynamic performance assessment of a ducted fan uav for vtol applications. *Aerosp. Sci. Technol.* **103**, 105895 (2020).
46. Bangura, M. & Mahony, R. Thrust control for multirotor aerial vehicles. *IEEE Trans. Robot.* **33**, 390 (2017).

47. Herisse, B., Russotto, F.-X., Hamel, T. & Mahony, R. Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow. *IEEE/RSJ Int. Conf. Intell. Rob. Syst. (IROS)* 801–806 (IEEE, 2008).
48. Smeur, E. J., de Croon, G. C. & Chu, Q. Gust disturbance alleviation with incremental nonlinear dynamic inversion. *IEEE/RSJ Int. Conf. Intell. Rob. Syst. (IROS)* 5626–5631 (IEEE, 2016).
49. McGuire, K. N., de Croon, G. C. & Tuyls, K. A comparative study of bug algorithms for robot navigation. *Robot. Auton. Syst.* **121**, 103261 (2019).
50. Nelson, D. R., Barber, D. B., McLain, T. W. & Beard, R. W. Vector field path following for miniature air vehicles. *IEEE Trans. Robot.* **23**, 519 (2007).
51. Zhou, D. & Schwager, M. Vector field following for quadrotors using differential flatness. *IEEE Int. Conf. Rob. Autom. (ICRA)* 6567–6572 (IEEE, 2014).
52. Soria, E., Schiano, F. & Floreano, D. Predictive control of aerial swarms in cluttered environments. *Nat. Mach. Intell.* **3**, 545 (2021).
53. Majumdar, A., Farid, A. & Sonar, A. Pac-bayes control: learning policies that provably generalize to novel environments. *Int. J. Robot. Res.* **40**, 574 (2021).

ACKNOWLEDGEMENTS

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (project no. CityU 11207621) and the Shenzhen-Hongkong-Macau Science and Technology Project (Category C) (project no. SGDX20220530111401009).

AUTHOR CONTRIBUTIONS

R.D. and P.C. conceived the ideas and designed the study. R.D. and K.D. designed and conducted the aerodynamic force measurements. R.D. and P.C. analyzed and interpreted the results. R.D. designed, fabricated, and tested the prototypes. R.D. and P.C. designed the flight experiments. R.D. prepared the flight experiments. R.D. and S.B. performed the flight experiments. R.D. and P.C. wrote and revised the manuscripts. All authors read and approved the final manuscript.

COMPETING INTERESTS

All authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44182-023-00002-9.

Correspondence and requests for materials should be addressed to Pakpong Chirarattananon.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.