

An autonomous single-actuator UAV with omnidirectional field of view, high agility, and collision resistance

Nan Chen¹, Haotian Li¹, Yunfan Ren¹, Guozheng Lu¹,
Fangcheng Zhu¹, Jiarong Lin¹, Fu Zhang^{1*}

¹Department of Mechanical Engineering, The University of Hong Kong,
Pokfulam Road, Hong Kong, China.

*Corresponding author(s). E-mail(s): fuzhang@hku.hk;

Contributing authors: cnchen@connect.hku.hk;
haotianli@connect.hku.hk; renyf@connect.hku.hk; gzlu@connect.hku.hk;
zhufc@connect.hku.hk; zivlin@connect.hku.hk;

1.2 Voltage modulation

To generate the acceleration and deceleration of the motor smoothly, we modulated the motor voltage in a sinusoidal form. Specifically, a sinusoidal voltage is added to the motor voltage command, which is

$$u_t = u_n + u_a \sin(\varphi - \gamma) \quad (\text{S1})$$

where u_t is the total voltage, u_n is the nominal voltage, u_a is the amplitude of the sinusoidal voltage, φ is the motor angle measured by the magnetic encoder, and γ is the rotor angle where the acceleration begins (i.e., the rotor acceleration position).

According to Supplementary Fig. 1a, the rotor acceleration position γ can be represented as

$$\gamma = \beta - \pi/2 + \delta_0 - \lambda \quad (\text{S2})$$

where β is the direction of the generated net moment, λ is the lag angle of moment direction caused by various factors such as the inertia of SLM, encoder measurement delay, and software processing time. We neglect the very small lag angle δ_0 and substitute Equation (S2) into Equation (S1), yielding

$$\begin{aligned} u_t &= u_n + u_a \sin(\varphi - \beta + \pi/2 + \lambda) \\ &= u_n + u_a \cos(\varphi - \beta + \lambda). \end{aligned} \quad (\text{S3})$$

It is noted that the motor angle φ is measured by the magnetic encoder in the body frame $\mathcal{B}$, but we need to generate the moment in the fictional frame $\mathcal{F}$. Since the pure yaw rotation between the frame $\mathcal{B}$ and frame $\mathcal{F}$ is known as $\mathbf{R}^{\mathcal{F}\mathcal{B}}$ (see the Supplementary Note 6 for details of calculation), we denote this yaw angle as ψ_{yaw} . Then, we can calculate the motor angle in frame $\mathcal{F}$ as $\varphi + \psi_{\text{yaw}}$, and finally yield the modulated voltage as

$$u_t = u_n + u_a \cos(\varphi + \psi_{\text{yaw}} - \beta + \lambda). \quad (\text{S4})$$

In Equation (S4), φ and ψ_{yaw} are known and β can be calculated according to the desired moment direction. In the real system, since λ depends on the motor speed Ω_R , we represent it as $\lambda = g_\lambda(\Omega_R)$ where $g_\lambda(\cdot)$ is a nonlinear function that will be determined through actual experiments. Here, with given u_n , u_a , known $g_\lambda(\cdot)$, and measured Ω_R , the total voltage u_t can be fully determined.

1.3 Mapping of controller outputs

Through actual experiments on the test stand (see Supplementary Note 2), we found that the moment magnitude $\|\mathbf{M}_{xy}^{\mathcal{F}}\|$ mainly depends on the amplitude of the sinusoidal throttle u_a and the motor speed Ω_R , as well as the propeller thrust f_T , which mainly depends on the nominal throttle u_n , such that we can define the following mapping

$$\|\mathbf{M}_{xy}^{\mathcal{F}}\| = g_M(\Omega_R, u_a) \cdot u_a, \quad f_T = g_f(u_n) \quad (\text{S5})$$

where $g_M(\cdot)$ and $g_f(\cdot)$ are two nonlinear functions. If functions $g_M(\cdot)$ and $g_f(\cdot)$ are determined, with measured Ω_R and the desired moment of the controllers $\mathbf{M}_{xy,d}^{\mathcal{F}}$ and $f_{T,d}$, we can determine the u_a and u_n by function inversion.

To generate a moment in a desired direction, we can specify the moment direction β as the direction of the controller's desired moment $\mathbf{M}_{xy,d}^{\mathcal{F}}$, which is

$$\beta = \arctan\left(\frac{M_{y,d}^{\mathcal{F}}}{M_{x,d}^{\mathcal{F}}}\right) \quad (\text{S6})$$

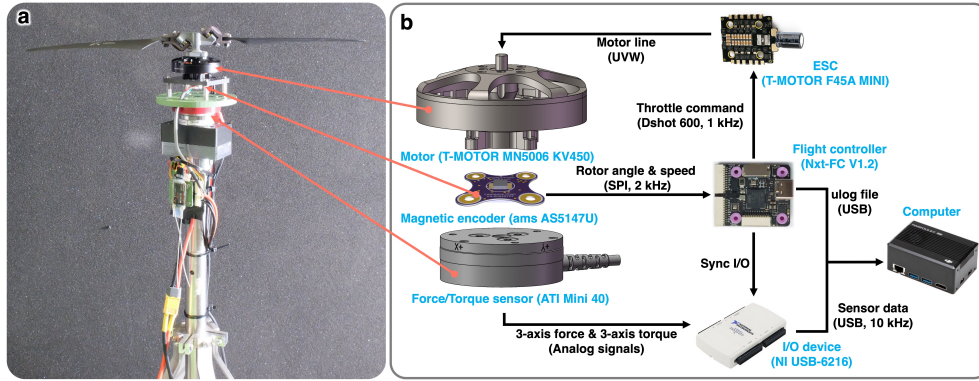
Here, we introduced the mapping of controller outputs, namely that the desired moment $\mathbf{M}_{xy,d}^{\mathcal{F}}$ generated by the angular velocity controller is mapped to the amplitude of sinusoidal voltage u_a and the moment direction β , as well as the desired thrust $f_{T,d}$ generated by the on-manifold model predictive control (OMMPC) is mapped to

the nominal voltage u_n . The unknown functions $g_M(\cdot)$ and $g_f(\cdot)$ will be determined using data from actual experiments of the propulsion system.

2 Actuator modeling

2.1 Motor states measurement

According to Equation (S4), the voltage modulation is based on the angle of the rotor. Meanwhile, some SLM's static models (e.g., functions $g_\lambda(\cdot)$ and $g_M(\cdot)$) are related to the rotor speed. Hence, the angle and speed of the rotor are two key states, which are measured by a magnetic encoder. We installed it at the bottom of the motor as shown in the Supplementary Fig. 2 and Supplementary Fig. 17. The encoder (chip type ams AS5147U) measures the rotor angle and rotational speed at the same time with a high frequency of 2000 Hz and transmits the data to the FCU via high-speed Serial Peripheral Interface (SPI).



Supplementary Fig. 2: Structure and internal connection of the test stand. **a** Overall structure of the test stand. **b** Detailed internal connection of the components.

2.2 Test stand for the propulsion system

To obtain the static models (i.e., functions $g_\lambda(\cdot)$, $g_M(\cdot)$, and $g_f(\cdot)$) and the dynamic model (i.e., the transient response model of moments) of SLM, we built a test stand for data collection as shown in Supplementary Fig. 2. We installed the propulsion system (including the motor, propeller, and SLM) to a high-bandwidth 6-axis force/torque sensor (ATI Mini 40) which can measure both 3D forces and moments dynamically. The output of the force/torque sensor is transmitted to a computer via NI's USB-6216 I/O device with a 10-kHz sampling frequency. The data are finally recorded to files with the timestamp of the computer for further analysis.

In addition to forces and moments, the voltage and current are measured by the electronic speed controller (ESC), and the rotor angle and speed are measured by the magnetic encoder. Such measured data and actuation commands (e.g., desired amplitude and direction of moment) are all recorded in the log file of the FCU with a timestamp from the FCU. Since the timestamps of the computer and FCU are different, to realize time synchronization, the FCU generates an additional step signal (internally recorded in the log file) on a Sync I/O such that it can be captured by the computer. Since the very small time delay from signal generation to signal capture can be neglected, once the step signal from the FCU's log file and that of the computer's recorded data are aligned on the time axis, all data can be synchronized, which is fundamental to performing identification for the dynamic model of SLM.

2.3 Static modeling and compensation of SLM

2.3.1 Data collection for static modeling

According to Equation (S2), we neglect δ_0 and set $\gamma = -\pi/2$ such that the actual moment direction $\beta = \lambda$. In this situation, the generated moment satisfies

$$M_x = \|\mathbf{M}_{xy}^{\mathcal{F}}\| \cos(\lambda), \quad M_y = \|\mathbf{M}_{xy}^{\mathcal{F}}\| \sin(\lambda). \quad (\text{S7})$$

and we can measure the moment $\mathbf{M}_{xy}$ to determine λ .

In FCU's software, the total throttle u_t is mapped from [0%, 100%] to the DShot value [0, 2000]. To cover various operating conditions of the SLM, during the data collection process, we specified a range of variation of the nominal voltage u_n from 500 to 1700 DShot value with a step of 200 to change the rotation speed of the propeller Ω_R , and a range of variation of the sinusoidal voltage amplitude u_a from 0 to 300 DShot value with a step of 50 to change the moment amplitude $\|\mathbf{M}_{xy}^{\mathcal{F}}\|$. The collected data are shown in Supplementary Fig. 3.

2.3.2 Static modeling and compensation

According to the data collected from the test stand, we can obtain the static model of SLM (i.e., functions $g_\lambda(\cdot)$, $g_M(\cdot)$, and $g_f(\cdot)$). We assume that g_λ and g_M have the following polynomial form as

$$\begin{aligned} \lambda &= g_\lambda(\Omega_R) \approx \mathbf{k}_1^T \cdot \mathbf{r}_1 \\ \frac{\|\mathbf{M}_{xy}^{\mathcal{F}}\|}{u_a} &= g_M(\Omega_R, u_a) \approx \mathbf{k}_2^T \cdot \mathbf{r}_2 \end{aligned} \quad (\text{S8})$$

where $\mathbf{k}_1 \in \mathbb{R}^4$ and $\mathbf{k}_2 \in \mathbb{R}^7$ are two coefficient vectors, $\mathbf{r}_1 = [\Omega_R^3, \Omega_R^2, \Omega_R, 1]^T \in \mathbb{R}^4$, $\mathbf{r}_2 = [\Omega_R^3, \Omega_R^2, \Omega_R, \Omega_R^2 u_a, \Omega_R u_a, u_a, 1]^T \in \mathbb{R}^7$. Note that the values of Ω_R (unit is rpm) are divided by 1000, and the value of u_a (unit is the DShot value) is divided by 100 when determining $\mathbf{r}_1$ and $\mathbf{r}_2$ to avoid too small values of $\mathbf{k}_1$ and $\mathbf{k}_2$.

Since M_x and M_y can be directly measured by the force sensor as $M_{x,m,i}$ and $M_{y,m,i}$ (i being i -th measurement), we can define the i -th errors as

$$\begin{aligned} \delta_{1,i} &= \|\mathbf{M}_{xy,m,i}\| \cos(\lambda_i) - M_{x,m,i} \\ &= \|\mathbf{M}_{xy,m,i}\| \cos(\mathbf{k}_1^T \cdot \mathbf{r}_{1,i}) - M_{x,m,i} \\ \delta_{2,i} &= \|\mathbf{M}_{xy,m,i}\| \sin(\lambda_i) - M_{y,m,i} \\ &= \|\mathbf{M}_{xy,m,i}\| \sin(\mathbf{k}_1^T \cdot \mathbf{r}_{1,i}) - M_{y,m,i} \\ \delta_{3,i} &= \|\mathbf{M}_{xy,i}\| - \|\mathbf{M}_{xy,m,i}\| \\ &= u_{a,i} \mathbf{k}_2^T \cdot \mathbf{r}_{2,i} - \|\mathbf{M}_{xy,m,i}\| \end{aligned} \quad (\text{S9})$$

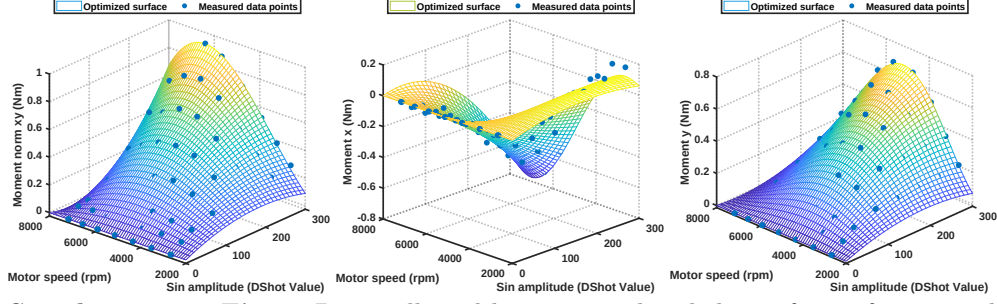
where $\|\mathbf{M}_{xy,m,i}\| = \sqrt{M_{x,m,i}^2 + M_{y,m,i}^2}$. The values of $\mathbf{r}_{1,i}$ and $\mathbf{r}_{2,i}$ can be calculated by the i -th value of $\Omega_{R,i}$ measured by the encoder and the specified value of $u_{a,i}$. Hence, the only unknown parameters are $\mathbf{k}_1$ and $\mathbf{k}_2$. To determine them, an optimization problem is formulated as

$$\min_{\mathbf{k}_1, \mathbf{k}_2} \sum_{i=0}^N (\delta_{1,i}^2 + \delta_{2,i}^2 + \delta_{3,i}^2) \quad (\text{S10})$$

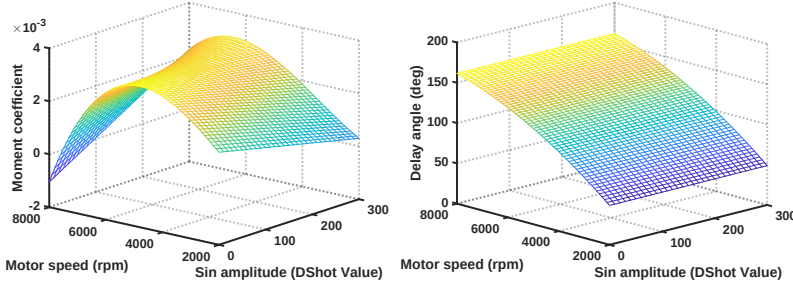
where N is the total data number.

2.3.3 Static modeling result

The optimization converged with a minimum value of the objective function of 0.1106 and the unknown parameters are determined as $\mathbf{k}_1 = [1.761 \times 10^{-5}, -0.027, 0.5969, -0.2343]$ and $\mathbf{k}_2 = [-0.0032, 0.0134, 0.0905, 0.0065, -0.0406, 0.0151, -0.06]$. The determined functions are shown in Fig. 4. The $g_M(\cdot)$ is a highly nonlinear surface depending on the motor speed and the sin amplitude, and the $g_\lambda(\cdot)$ obviously depends on the



Supplementary Fig. 3: Data collected by test stand and the surfaces of optimized results.



Supplementary Fig. 4: Functions of moment gain $g_M(\cdot)$ and delay angle $g_\lambda(\cdot)$ obtained by optimization.

motor speed. We also used the optimized $\mathbf{k}_1$ and $\mathbf{k}_2$ to calculate the moment norm and moment components in both x and y axes, and the results are shown in Fig. 3. The calculated results are very close to the measured data points, which indicates that the static model is accurate.

In the actual system, based on the static model, we can directly calculate the delay angle λ for delay compensation according to the measured rotor speed Ω_R . Similarly, given the measured rotor speed Ω_R and the norm of the desired moment $\|\mathbf{M}_{xy}^{\mathcal{F}}\|$, the desired $u_{a,d}$ can be solved.

2.4 Dynamic modeling and identification of SLM

2.4.1 Model assumption for dynamic modeling

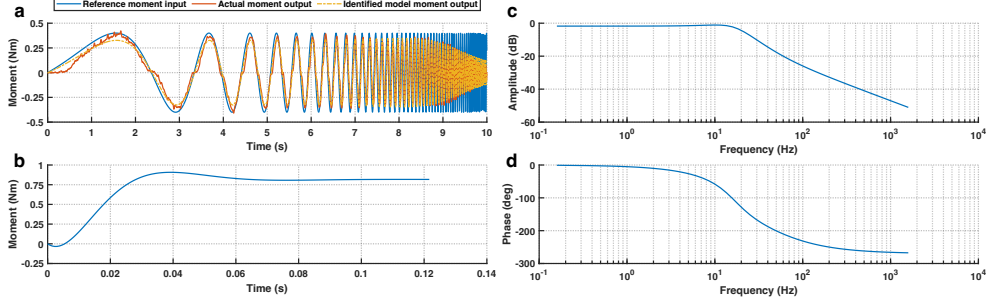
Since the nonlinearity of the SLM can be well compensated through the static model, we can consider the dynamic model of SLM as a linear model. We assume that the dynamic model is decoupled and identical in x and y axes, and hence it can be represented by a diagonal transfer function matrix $\mathbf{G}_a(s)$ as

$$\mathbf{M}_{xy}(s) = \mathbf{G}_a(s)\mathbf{M}_{xy,d}(s) = \begin{bmatrix} G_a(s) & 0 \\ 0 & G_a(s) \end{bmatrix} \mathbf{M}_{xy,d}(s) \quad (\text{S11})$$

where $\mathbf{M}_{xy}(s) = [M_x(s), M_y(s)]^T$ and $\mathbf{M}_{xy,d}(s) = [M_{x,d}(s), M_{y,d}(s)]^T$. $G_a(s)$ is a single-input-single-output (SISO) transfer function where the input is the desired moment amplitude and the output is the actual moment amplitude.

2.4.2 Data collection for dynamic modeling

We used the test stand to collect data for dynamic modeling. Specifically, a chirp signal was employed as the input to stimulate the system under a situation where the rotor speed is the same as the value when the UAV is in a hover state. We only stimulated the x -axis moment since we assumed it is identical in the two axes. On the test stand, the chirp signal (i.e., input) is recorded in the log file of the FCU, and the generated moments (i.e., output) are recorded on the computer. The timestamps of different devices are synchronized via the step signal as mentioned before.



Supplementary Fig. 5: The data used for identification and the identified model. **a** Input data, output data, and model output. **b** Unit step response of the identified model. **c** Amplitude characteristic of Bode diagram of the identified model. **d** Phase characteristic of Bode diagram of the identified model.

2.4.3 Identification result

Before identification, the collected input and output data shown in Supplementary Fig. 5a are down-sampled to the same sampling frequency. Then, we can perform model identification in the frequency domain to determine the transfer function of the actuator via the System Identification Toolbox in MATLAB. The identified transfer function $G_a(s)$ is

$$G_a(s) = \frac{-27.97s + 9028}{s^2 + 121.9s + 11060} \quad (\text{S12})$$

which is an underdamped second-order transfer function with a right half-plane (RHP) zero. Hence, the moment generation dynamics of the SLM is a non-minimum-phase system.

The unit step response and the Bode diagram of the identified model are shown in Supplementary Fig. 5(b-d). In Supplementary Fig. 5(b), we can see a negative adjustment in the time domain due to the non-minimum phase characteristic, and the bandwidth of the actuator is about 17.4 Hz in the frequency domain. Furthermore, we plotted the output of the identified model stimulated by the same input data (i.e., reference moment input) in Supplementary Fig. 5a. It can be seen that the model output is very close to the actual moment output, indicating that the identified model is accurate.

3 Trajectory tracking control on S^2 manifold

3.1 Minimal parameterization of input

The angular velocity $\omega_{\mathcal{F}}^{\mathcal{I}} \in \mathbb{R}^3$ in Equation (15) is overparameterized for a point in S^2 manifold since the valid degrees of freedom (DoF) of $\omega_{\mathcal{F}}^{\mathcal{I}}$ should be 2 due to an unnecessary component parallel to $\mathbf{n}^{\mathcal{I}}$. To achieve minimal parameterization (i.e., the number of parameters is equal to the DoF), the remaining components perpendicular to $\mathbf{n}^{\mathcal{I}}$ can be represented in the tangent space of $\mathbf{n}^{\mathcal{I}}$ as a $\mathbb{R}^2$ vector, which is

$$\bar{\omega} = (\mathbf{B}(\mathbf{n}^{\mathcal{I}}))^{\top} \omega_{\mathcal{F}}^{\mathcal{I}} \in \mathbb{R}^2 \quad (\text{S13})$$

where the overline of $\bar{\omega}$ represents that it is a $\mathbb{R}^2$ vector in the tangent space. $\mathbf{B}(\mathbf{n}^{\mathcal{I}}) = [\mathbf{b}_1, \mathbf{b}_2] \in \mathbb{R}^{3 \times 2}$ is a basis matrix, which can be chosen arbitrarily as long as it forms an orthonormal basis in the tangent plane of $\mathbf{n}^{\mathcal{I}}$, namely, satisfying the constraints of $\mathbf{b}_1 \perp \mathbf{b}_2 \perp \mathbf{n}^{\mathcal{I}}$ and $\|\mathbf{b}_1\| = \|\mathbf{b}_2\| = 1$.

The basis matrix $\mathbf{B}(\mathbf{n}^\mathcal{I})$ can be defined as a function of $\mathbf{n}^\mathcal{I}$. This function is not unique as long as it satisfies the constraints, and hence we define it as

$$\mathbf{R}(\mathbf{n}^\mathcal{I}) = \begin{cases} \text{Exp}(\frac{[\mathbf{w}_3]^\mathcal{I} \mathbf{n}^\mathcal{I}}{\|[\mathbf{w}_3]^\mathcal{I} \mathbf{n}^\mathcal{I}\|} \cdot \theta), \theta = \text{atan2}(\|[\mathbf{w}_3]^\mathcal{I} \mathbf{n}^\mathcal{I}\|, \mathbf{w}_3^\mathcal{T} \mathbf{n}^\mathcal{I}) & \text{if } [\mathbf{w}_3]^\mathcal{I} \mathbf{n}^\mathcal{I} \neq \mathbf{0}, \\ \mathbf{I}, & \text{if } \mathbf{n}^\mathcal{I} = \mathbf{w}_3, \\ \text{undefined}, & \text{if } \mathbf{n}^\mathcal{I} = -\mathbf{w}_3, \end{cases} \quad (\text{S14})$$

$$\mathbf{B}(\mathbf{n}^\mathcal{I}) = \mathbf{R}(\mathbf{n}^\mathcal{I})[\mathbf{w}_1, \mathbf{w}_2]$$

where $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ are arbitrary standard orthogonal bases. The reason why the undefined point exists is that the rotation axis is non-unique when $\mathbf{n}^\mathcal{I} = -\mathbf{w}_3$. For convenience, we can choose the standard orthogonal basis $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ exactly the same as the basis of the inertia frame $\mathcal{I}$ which are $\mathbf{w}_1 = \mathbf{e}_1 = [1, 0, 0]^\mathcal{T}, \mathbf{w}_2 = \mathbf{e}_2 = [0, 1, 0]^\mathcal{T}, \mathbf{w}_3 = \mathbf{e}_3 = [0, 0, 1]^\mathcal{T}$.

With the definition above, the continuous system expressed by Equation (15) can be rewritten in the following form

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}}^\mathcal{I} \\ \dot{\mathbf{v}}^\mathcal{I} \\ \dot{\mathbf{n}}^\mathcal{I} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^\mathcal{I} \\ a_T \mathbf{n}^\mathcal{I} + \mathbf{g}^\mathcal{I} \\ [\mathbf{B}(\mathbf{n}^\mathcal{I}) \cdot \bar{\boldsymbol{\omega}}]^\mathcal{I} \mathbf{n}^\mathcal{I} \end{bmatrix} \quad (\text{S15})$$

Compared with Equation (15), the dimension of input $\mathbf{u} = [a_T, \bar{\boldsymbol{\omega}}^\mathcal{T}]^\mathcal{T}$ is reduced from 4 to 3, which is identical to the valid DoFs of the input. Hence, the system input is now minimally parameterized.

3.2 Augmentation of integration state

To deal with the unmodeled factors (e.g., air drag) and time-varying variables (e.g., thrust coefficient reduction due to battery voltage dropping), we augmented the system state with integration of position error $\mathbf{e}_p^\mathcal{I} = \int (\mathbf{p}^\mathcal{I} - \mathbf{p}_d^\mathcal{I}) dt$, which can achieve the integrator-like effect in the OMMPC. The resulting augmented dynamic model is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{e}}_p^\mathcal{I} \\ \dot{\mathbf{p}}^\mathcal{I} \\ \dot{\mathbf{v}}^\mathcal{I} \\ \dot{\mathbf{n}}^\mathcal{I} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^\mathcal{I} - \mathbf{p}_d^\mathcal{I} \\ \mathbf{v}^\mathcal{I} \\ a_T \mathbf{n}^\mathcal{I} + \mathbf{g}^\mathcal{I} \\ [\mathbf{B}(\mathbf{n}^\mathcal{I}) \cdot \bar{\boldsymbol{\omega}}]^\mathcal{I} \mathbf{n}^\mathcal{I} \end{bmatrix} \quad (\text{S16})$$

Note that $\mathbf{p}_d^\mathcal{I}$ is not defined in the state. It is a known and time-varying variable, such that the system becomes a nonlinear time-varying system.

3.3 Operators definition of S^2 manifold

For $SO(3)$ manifold, the mapping function $\mathbb{R}^3 \mapsto SO(3) : \text{Exp}(\cdot) = \exp([\cdot])$, where $[\cdot]$ is a skew-symmetric matrix, is an exponential map converting a Lie algebra (i.e., a $\mathbb{R}^3$ vector on the tangent space of the initial point of $SO(3)$ manifold) to a point on $SO(3)$ manifold. Similarly, with the tangent space defined above, we can define the operators for the S^2 manifold, which are the addition $\boxplus$ and subtraction $\boxminus$ as

$$\boxplus : S^2 \times \mathbb{R}^2 \mapsto S^2, \quad \boldsymbol{\alpha} \boxplus \boldsymbol{\delta} = \text{Exp}(\mathbf{B}(\boldsymbol{\alpha}) \cdot \boldsymbol{\delta}) \cdot \boldsymbol{\alpha} \quad (\text{S17})$$

$$\boxminus : S^2 \times S^2 \mapsto \mathbb{R}^2, \quad \boldsymbol{\beta} \boxminus \boldsymbol{\alpha} = (\mathbf{B}(\boldsymbol{\alpha}))^\mathcal{T} \cdot (\theta \frac{[\boldsymbol{\alpha}] \boldsymbol{\beta}}{\|[\boldsymbol{\alpha}] \boldsymbol{\beta}\|}) \quad (\text{S18})$$

$$\theta = \text{atan2}(\|[\boldsymbol{\alpha}] \boldsymbol{\beta}\|, \boldsymbol{\alpha}^\mathcal{T} \boldsymbol{\beta})$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in S^2$ (i.e., they are $\mathbb{R}^3$ vectors with a constraint $\|\boldsymbol{\alpha}\| = \|\boldsymbol{\beta}\| = 1$), $\boldsymbol{\delta} \in \mathbb{R}^2$ located on the tangent space of $\boldsymbol{\alpha}$, θ being the intersection angle between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

3.4 System discretization

To discretize the system in Equation (S16) with a fixed time step Δt , first, we expressed the states and inputs at time k as

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{e}_{p,k}^{\mathcal{I}} \\ \mathbf{p}_k^{\mathcal{I}} \\ \mathbf{v}_k^{\mathcal{I}} \\ \mathbf{n}_k^{\mathcal{I}} \end{bmatrix} \in \mathcal{M}, \quad \mathbf{u}_k = \begin{bmatrix} a_{T,k} \\ \bar{\boldsymbol{\omega}}_k \end{bmatrix} \in \mathbb{R}^3 \quad (\text{S19})$$

Then, according to the first-order discretization $\dot{\mathbf{x}} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t}$, the state at time $k+1$ can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k \boxplus \Delta t \dot{\mathbf{x}} \quad (\text{S20})$$

Combining Equation (S16) and the definition of $\boxplus$ for S^2 manifold, we can define

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{p}_k^{\mathcal{I}} - \mathbf{p}_{k,d}^{\mathcal{I}} \\ \mathbf{v}_k^{\mathcal{I}} \\ a_{T,k} \mathbf{n}_k^{\mathcal{I}} + \mathbf{g}^{\mathcal{I}} \\ \bar{\boldsymbol{\omega}}_k \end{bmatrix}_{11 \times 1} \quad (\text{S21})$$

such that $\Delta t \dot{\mathbf{x}}$ satisfies the definition of δ in Equation (S17). Finally, we can write the discrete system as

$$\mathbf{x}_{k+1} = \mathbf{x}_k \boxplus \Delta t \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{e}_{p,k}^{\mathcal{I}} + \Delta t(\mathbf{p}_k^{\mathcal{I}} - \mathbf{p}_{k,d}^{\mathcal{I}}) \\ \mathbf{p}_k^{\mathcal{I}} + \Delta t \mathbf{v}_k^{\mathcal{I}} \\ \mathbf{v}_k^{\mathcal{I}} + \Delta t(a_{T,k} \mathbf{n}_k^{\mathcal{I}} + \mathbf{g}^{\mathcal{I}}) \\ \text{Exp}(\mathbf{B}(\mathbf{n}_k^{\mathcal{I}}) \cdot \Delta t \bar{\boldsymbol{\omega}}_k) \mathbf{n}_k^{\mathcal{I}} \end{bmatrix}_{12 \times 1} \quad (\text{S22})$$

It is noted that the system states of the discrete dynamic model in Equation (S22) evolve on a compound manifold as below

$$\mathcal{M} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times S^2 \quad (\text{S23})$$

whose dimension is $\dim(\mathcal{M}) = 11$.

3.5 Error definition

3.5.1 Error definition of state

The trajectory planner can generate a position trajectory with its high-order derivatives (i.e., $\mathbf{p}^{\mathcal{I}}$, $\mathbf{v}^{\mathcal{I}}$, $\mathbf{a}^{\mathcal{I}}$, $\mathbf{j}^{\mathcal{I}}$). According to the differential flatness, they can be converted to the trajectory of the desired state $\mathbf{x}_{d,k} = [(\mathbf{e}_{p,d,k}^{\mathcal{I}})^{\mathcal{T}}, (\mathbf{p}_{d,k}^{\mathcal{I}})^{\mathcal{T}}, (\mathbf{v}_{d,k}^{\mathcal{I}})^{\mathcal{T}}, (\mathbf{n}_{d,k}^{\mathcal{I}})^{\mathcal{T}}]^{\mathcal{T}}$ ($\mathbf{e}_{p,d,k}^{\mathcal{I}} \equiv \mathbf{0}$) and the desired input $\mathbf{u}_d = [a_{T,d,k}, (\bar{\boldsymbol{\omega}}_{d,k})^{\mathcal{T}}]^{\mathcal{T}}$. Referring to prior work [68, 69], we define the error state in the local tangent space (i.e., an open set in Euclidean space) of the compound manifold $\mathcal{M}$ around each state point $\mathbf{x}_{d,k}$ as

$$\delta \mathbf{x}_k \triangleq \mathbf{x}_k \boxminus \mathbf{x}_{d,k} \in \mathbb{R}^{11} \quad (\text{S24})$$

or expanding for each component of the state as

$$\delta \mathbf{e}_{p,k}^{\mathcal{I}} \triangleq \mathbf{e}_{p,k}^{\mathcal{I}} \boxminus \mathbf{e}_{p,d,k}^{\mathcal{I}} = \mathbf{e}_{p,k}^{\mathcal{I}} - \mathbf{e}_{p,d,k}^{\mathcal{I}} \in \mathbb{R}^3 \quad (\text{S25a})$$

$$\delta \mathbf{p}_k^{\mathcal{I}} \triangleq \mathbf{p}_k^{\mathcal{I}} \boxminus \mathbf{p}_{d,k}^{\mathcal{I}} = \mathbf{p}_k^{\mathcal{I}} - \mathbf{p}_{d,k}^{\mathcal{I}} \in \mathbb{R}^3 \quad (\text{S25b})$$

$$\delta \mathbf{v}_k^{\mathcal{I}} \triangleq \mathbf{v}_k^{\mathcal{I}} \boxminus \mathbf{v}_{d,k}^{\mathcal{I}} = \mathbf{v}_k^{\mathcal{I}} - \mathbf{v}_{d,k}^{\mathcal{I}} \in \mathbb{R}^3 \quad (\text{S25c})$$

$$\delta \mathbf{n}_k^{\mathcal{T}} \triangleq \mathbf{n}_k^{\mathcal{T}} \boxminus \mathbf{n}_{d,k}^{\mathcal{T}} = (\mathbf{B}(\mathbf{n}_{d,k}^{\mathcal{T}}))^{\top} \cdot \left(\theta \frac{[\mathbf{n}_{d,k}^{\mathcal{T}}] \mathbf{n}_k^{\mathcal{T}}}{\|[\mathbf{n}_{d,k}^{\mathcal{T}}] \mathbf{n}_k^{\mathcal{T}}\|} \right) \in \mathbb{R}^2 \quad (\text{S25d})$$

As we can see, the dimension of the error state $\delta \mathbf{x}_k$ is 11, which is equal to the dimension of the compound manifold $\dim(\mathcal{M}) = 11$.

3.5.2 Error definition of input

For the scalar thrust, the error can be simply defined as

$$\delta a_{T,k} = a_{T,k} \boxminus a_{T,d,k} = a_{T,k} - a_{T,d,k} \in \mathbb{R}^1 \quad (\text{S26})$$

namely $a_{T,k} = a_{T,d,k} + \delta a_{T,k} \in \mathbb{R}^1$.

The angular velocity $\bar{\boldsymbol{\omega}}_k$ and $\bar{\boldsymbol{\omega}}_{d,k}$ are both $\mathbb{R}^2$ vectors, however, they are represented in different local tangent spaces of $\mathbf{n}_k^{\mathcal{T}}$ and $\mathbf{n}_{d,k}^{\mathcal{T}}$, respectively. Since their base vectors are different, their error cannot be defined as the direct subtraction of $\bar{\boldsymbol{\omega}}_k$ and $\bar{\boldsymbol{\omega}}_{d,k}$. Here, we define the error $\delta \bar{\boldsymbol{\omega}}_k$ implicitly by $\boxplus$ operation rather than $\boxminus$ operation in the local tangent space of $\mathbf{n}_{d,k}^{\mathcal{T}}$, leading to

$$\bar{\boldsymbol{\omega}}_k = \bar{\boldsymbol{\omega}}_{d,k} \boxplus \delta \bar{\boldsymbol{\omega}}_k = (\mathbf{B}(\mathbf{n}_k^{\mathcal{T}}))^{\top} \cdot \mathbf{B}(\mathbf{n}_{d,k}^{\mathcal{T}}) \cdot (\bar{\boldsymbol{\omega}}_{d,k} + \delta \bar{\boldsymbol{\omega}}_k) \in \mathbb{R}^2 \quad (\text{S27})$$

where $\delta \bar{\boldsymbol{\omega}}_k \in \mathbb{R}^2$. Note that Equation (S27) includes two space transformations. One is from the local tangent space of $\mathbf{n}_{d,k}^{\mathcal{T}}$ to $\mathbb{R}^3$, and another is from $\mathbb{R}^3$ to the local tangent space of $\mathbf{n}_k^{\mathcal{T}}$. The second transformation may cause the loss of the component parallel to $\mathbf{n}_k^{\mathcal{T}}$, and hence $\delta \bar{\boldsymbol{\omega}}_k$ cannot be represented by $\bar{\boldsymbol{\omega}}_k$ and $\bar{\boldsymbol{\omega}}_{d,k}$ by the inverse operation of Equation (S27). As a result, the error definition of input can be represented in a compact form as

$$\mathbf{u}_k = \mathbf{u}_{d,k} \boxplus \delta \mathbf{u}_k \in \mathbb{R}^3 \quad (\text{S28})$$

3.6 Error-state system dynamics

We need to establish the dynamics of the error-state system and control the error state $\delta \mathbf{x}_k$ converging to zero. With the error state defined in Equation (S24), the error input defined in Equation (S28), the actual trajectory $(\mathbf{x}_k, \mathbf{u}_k)$, and the reference trajectory $(\mathbf{x}_{k,d}, \mathbf{u}_{k,d})$ satisfying Equation (S22), we can calculate the differential of the error state as

$$\begin{aligned} \delta \mathbf{x}_{k+1} &= \mathbf{x}_{k+1} \boxminus \mathbf{x}_{d,k+1} \\ &= [\mathbf{x}_k \boxplus \Delta t \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)] \boxminus [\mathbf{x}_{d,k} \boxplus \Delta t \mathbf{f}(\mathbf{x}_{d,k}, \mathbf{u}_{d,k})] \\ &= [(\mathbf{x}_{d,k} \boxplus \delta \mathbf{x}_k) \boxplus \Delta t \mathbf{f}(\mathbf{x}_{d,k} \boxplus \delta \mathbf{x}_k, \mathbf{u}_{d,k} \boxplus \delta \mathbf{u}_k)] \boxminus [\mathbf{x}_{d,k} \boxplus \Delta t \mathbf{f}(\mathbf{x}_{d,k}, \mathbf{u}_{d,k})] \end{aligned} \quad (\text{S29})$$

Comparing the error-state system dynamics in Equation (S29) with the continuous system dynamics in Equation (S16), the dimension of the error state reduces by one, leading to an expected minimum parameterization. More importantly, both the error state and error input are vectors expressed in Euclidean space, which means that the constraint of the manifold is successfully removed.

Additionally, for the error-state system dynamics in Equation (S29), it is clear that we are required to define both $\boxplus$ and $\boxminus$ operations for the state $\mathbf{x}_k$, but we only need to define $\boxplus$ operation for the input $\mathbf{u}_k$, and hence the definition of Equation (S27) is feasible.

3.7 Linearization of error-state system dynamics

The error-state system dynamics in Equation (S29) is a nonlinear system without manifold constraints and over-parameterization. To achieve fast solving of MPC problem, we adopted a linearization strategy. With first-order linearization, the error-state system dynamics can be written as

$$\delta \mathbf{x}_{k+1} = \mathbf{F}_{\mathbf{x}_k} \delta \mathbf{x}_k + \mathbf{F}_{\mathbf{u}_k} \delta \mathbf{u}_k \quad (\text{S30})$$

where $\mathbf{F}_{\mathbf{x}_k} = \frac{\partial \delta \mathbf{x}_{k+1}}{\partial \delta \mathbf{x}_k} \big|_{\delta \mathbf{x}_k=0}$, $\mathbf{F}_{\mathbf{u}_k} = \frac{\partial \delta \mathbf{x}_{k+1}}{\partial \delta \mathbf{u}_k} \big|_{\delta \mathbf{u}_k=0}$ are two Jacobian matrices. They can be calculated by the chain rule analytically [47] or through numerical differentiation. In our implementation, we used numerical differentiation to calculate the Jacobian matrices, and the accuracy is sufficient to ensure the precise solution of the optimization problem.

3.8 Singularity conditions of basis matrix $\mathbf{B}(\mathbf{n}^{\mathcal{I}})$

For the definition of basis matrix $\mathbf{B}(\mathbf{n}^{\mathcal{I}})$ in Equation (S14), the term $\frac{|\mathbf{w}_3| \mathbf{n}^{\mathcal{I}}}{\|\mathbf{w}_3\| \mathbf{n}^{\mathcal{I}}}$ is singular when $\mathbf{n}^{\mathcal{I}} = -\mathbf{w}_3$. Considering $\mathbf{n}^{\mathcal{I}}$ is close to $\mathbf{n}_d^{\mathcal{I}}$ when control converges, we can use a different basis according to $\mathbf{n}_d^{\mathcal{I}}$ as

$$[\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3] = \begin{cases} [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] & \text{if } n_{d,z}^{\mathcal{I}} \geq 0, \\ [\mathbf{e}_1, -\mathbf{e}_2, -\mathbf{e}_3] & \text{if } n_{d,z}^{\mathcal{I}} < 0. \end{cases} \quad (\text{S31})$$

This definition indicates that the selection of the basis is dynamic to ensure the z axis of the basis vector (i.e., $\mathbf{w}_3$) is located on the same hemisphere as $\mathbf{n}_d^{\mathcal{I}}$, which can avoid $\mathbf{n}^{\mathcal{I}} = -\mathbf{w}_3$. In each control cycle, the basis can only be set once before the differential flatness calculation and MPC optimization, so the matrix $\mathbf{B}(\mathbf{n}^{\mathcal{I}})$ is still differentiable during the following optimization process.

3.9 Singularity conditions of operator $\boxminus$

As the definition shown in Equation (S18), if $\boldsymbol{\alpha}$ is parallel with $\boldsymbol{\beta}$, the term $\frac{|\boldsymbol{\alpha}| \boldsymbol{\beta}}{\|\boldsymbol{\alpha}\| \|\boldsymbol{\beta}\|}$ will be singular. Hence, when the intersection angle of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is zero, we can define $\boldsymbol{\beta} \boxminus \boldsymbol{\alpha} = \mathbf{0}_{2 \times 1}$ to avoid singularity.

If the intersection angle is π , the rotation axis cannot be determined, and the singularity happens. However, for the error-state system controlled by the OMMPC, the error between $\mathbf{x}_k$ and $\mathbf{x}_{k,d}$ should be small rather than a large value of π . Hence, this singularity can be avoided.

4 LiDAR odometry

The LIO integrated into PULSAR II's system is Swarm-LIO [54], which supports single UAV state estimation to UAV swarm state estimation. Swarm-LIO is an EKF-based framework that fuses two sensor input sources: IMU data and a LiDAR point cloud. In PULSAR II, the IMU data (acceleration and angular velocity) are sampled at 200 Hz from the IMU of the flight control unit (FCU) and used as model inputs for EKF propagation. The frame frequency of the LiDAR point cloud is configurable, typically 10 to 100 Hz. To ensure space overlap between two adjacent frames of LiDAR point clouds during high-speed self-rotation and to keep the frame frequency as high as possible, we set the frequency to 50 Hz for PULSAR II. Each time a new LiDAR point cloud frame arrives, Swarm-LIO uses the raw points to update the EKF state according to point-to-plane residuals.

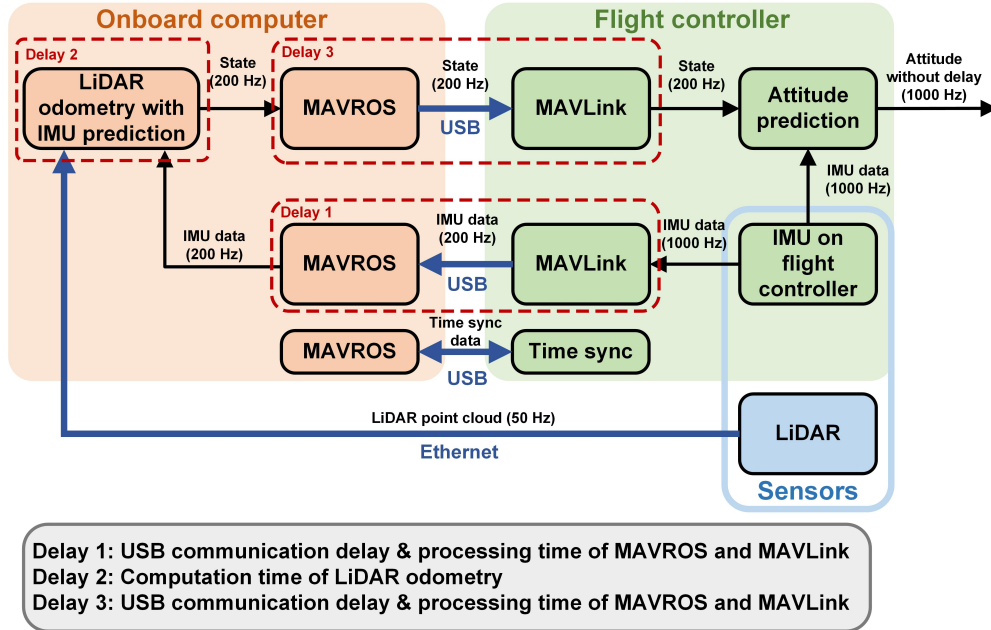
Since Swarm-LIO's inertia coordinate frame (i.e., all states are estimated with respect to this coordinate frame) is defined by the first frame of IMU data, we ensured the z axis of FCU's IMU sensor is carefully aligned with the z axis of PULSAR II (i.e., the self-rotation axis) during UAV design, to reduce the coupling with self-rotation. If they are not well aligned, the estimated position and z -axis direction may have sinusoidal variation. Then, the distance between the origin of Swarm-LIO's inertia coordinate frame and the origin of UAV's body frame along the z axis is well compensated according to the document of structure design. Now, the two origins are well aligned, and the estimation from LIO can reflect the actual UAV state in the body frame. Moreover, Swarm-LIO supports a feature of gravity alignment, which can align the estimated gravity direction with the z axis of Swarm-LIO's inertia coordinate frame (i.e., the gravity vector is $[0, 0, -g]^T$). This feature ensures that the attitude state is $[0, 0, 1]^T$ during hover without external forces (i.e., the z axis of the body frame is aligned with that of the inertia frame), even if the UAV takes off from a non-horizontal area.

The estimated state includes the UAV’s 3D position, velocity, and attitude (represented by a rotation matrix). The output frequency of posterior states (i.e., updated by LiDAR frame) is 50 Hz. To provide high-frequency state output, Swarm-LIO still outputs the prior state (i.e., propagated by IMU data) at 200 Hz. This high-frequency output is used for control of PULSAR II. Besides states, Swarm-LIO also maintains a 3D point cloud map by continuously registering each new LiDAR frame to the existing map using the estimated state. This map is used for obstacle-avoidance planning for PULSAR II.

5 Attitude delay compensation

5.1 Module structure of delay compensation

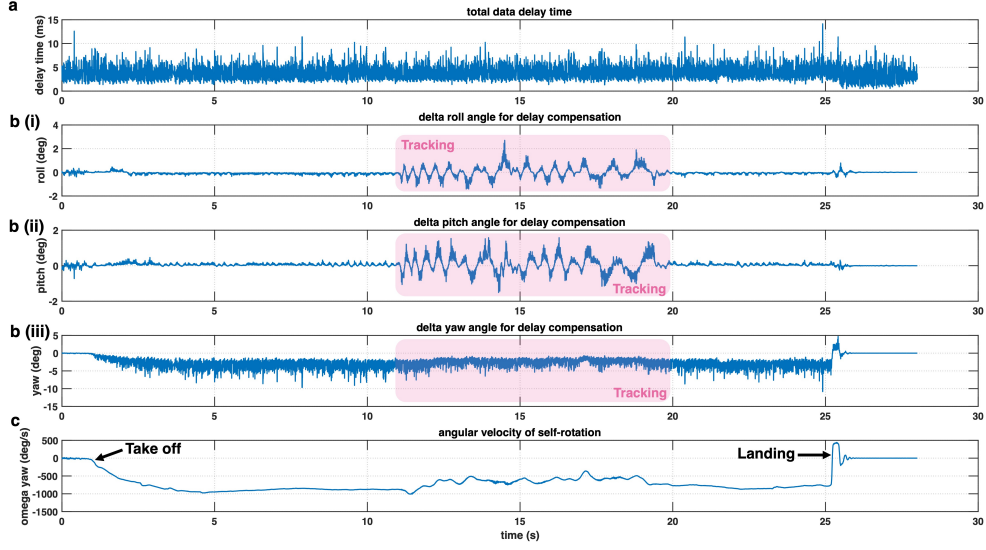
The structure of the module of attitude delay compensation is shown in Supplementary Fig. 6, with the key process of attitude prediction. Implementing the attitude prediction process in the FCU serves two reasons: (i) the 200-Hz attitude output frequency of LiDAR odometry (LIO) does not match the 1000-Hz low-level control frequency; and (ii) to compensate for the delays. The IMU data used in LIO originates from the FCU and the result of LIO, namely the estimated state, is then transmitted back to the FCU. This process introduces three delays: the first one is due to the computation time of LIO, and the remained two are induced by bi-directional USB communication combined with the processing time of MAVROS [55] and MAVLink [70]. These delays necessitate attitude prediction, especially during aggressive motion.



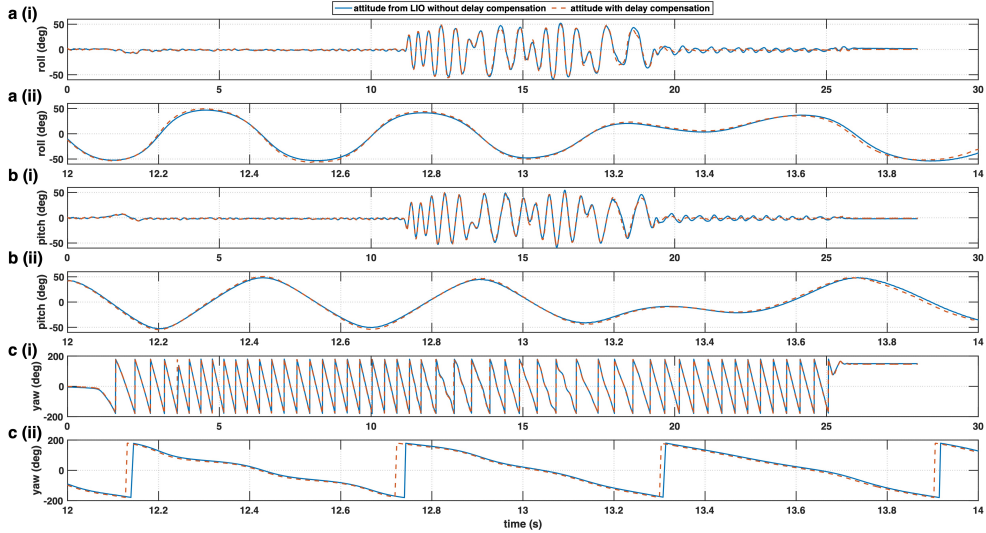
Supplementary Fig. 6: Structure of the module of attitude delay compensation.

Since the timestamp difference of the LIO’s LiDAR point cloud and the IMU data is estimated through LI-init calibration [56] and well compensated, we can deem these data to be synchronized. Since MAVROS and MAVLink provide time synchronization that aligns the onboard computer’s time base with the FCU’s, when the attitude prediction module receives a delayed 200-Hz LIO state (due to USB communication, LIO computation time, etc.), it can predict the state from its timestamp (since the timestamp of the estimated state is assigned by the timestamp of the IMU data) to the current time using 1000-Hz IMU data. This process resolves the time delay issue while simultaneously increasing the attitude feedback frequency to 1000 Hz, meeting the angular velocity control requirement. To implement this, we added an IMU data buffer in the attitude prediction module with a buffer size. This size is determined by

the maximum delay time divided by the IMU sampling frequency (e.g., for a 50-ms maximum delay and 1000-Hz sampling frequency, the buffer size should be 50 samples).



Supplementary Fig. 7: Compensation results during tracking of the figure 8 trajectory (period = 4 s). **a** Total data delay time, which is the sum of delay 1, delay 2, and delay 3. **b** Attitude compensation values. The compensation is conducted by rotation matrix but it is visualized by Euler angles. (i) Roll angle. (ii) Pitch angle. (iii) Yaw angle. **c** Angular velocity of self-rotation during flight.



Supplementary Fig. 8: Attitude comparison with or without compensation during tracking of the figure 8 trajectory (period = 4 s). **a** Roll angle. (i) All data. (ii) Zoom-in view. **b** Pitch angle. (i) All data. (ii) Zoom-in view. **c** Yaw angle. (i) All data. (ii) Zoom-in view.

5.2 Results of delay compensation

During a figure 8 trajectory tracking experiment with a period of 4 s, the total data delay time, attitude compensation values, and self-rotation angular velocity were recorded, as illustrated in Fig. 7. The total data delay time, which varied from 2 to 15 ms, was primarily attributable to communication delays and LIO computation

time. This delay induces attitude error during fast rotational motion. To mitigate this error, attitude compensation was applied. The resulting compensation values are presented in Fig. 7b(i)-(iii). The compensation values for the roll and pitch angles remain near zero during hovering states but become significant during aggressive trajectory tracking. In contrast, due to UAV's continuous self-rotation (about 500-1000 degree/s shown in Fig. 7c), the yaw angle changes rapidly. The attitude compensation values for yaw angle are significant regardless of whether the UAV is in a hover state or in trajectory-tracking flight. Obvious yaw angle errors (e.g., about 10 degrees) may lead to inaccurate control moment directions, further degrading trajectory tracking performance.

In Fig. 8, we plotted the attitude in Euler angle form with and without compensation during the figure 8 trajectory tracking experiment with zoom-in views, for better illustrating the results of delay compensation. As shown, with compensation, the roll, pitch, and yaw angles all exhibit phase advancement compared to the cases without compensation, providing more real-time feedback for control and actuation.

6 $\mathcal{F}$ frame calculation

6.1 Attitude integration for frame $\mathcal{F}$

Although attitude $\mathbf{R}^{\mathcal{IB}}$ is estimated by the LIO module, the attitude $\mathbf{R}^{\mathcal{IF}}$ and $\mathbf{R}^{\mathcal{FB}}$ are still unknown such that the commands represented in $\mathcal{I}$ frame and measurements represented in $\mathcal{B}$ frame cannot be transformed and represented in the fictional frame $\mathcal{F}$ that decouples the self-rotation. Here, we directly use integration of the angular velocity measured by the IMU to calculate $\mathbf{R}^{\mathcal{FB}}$ and further to obtain $\mathbf{R}^{\mathcal{IF}}$. According to $\dot{\mathbf{R}}^{\mathcal{FB}} = \mathbf{R}^{\mathcal{FB}}[\omega_{\mathcal{B},z}^{\mathcal{B}}]$, given the initial condition $\mathbf{R}_0^{\mathcal{FB}}$, we can calculate $\mathbf{R}_k^{\mathcal{FB}}$ at k -th step by the discrete integration with interval Δt as

$$\mathbf{R}_k^{\mathcal{FB}} = \mathbf{R}_0^{\mathcal{FB}} \prod_{i=1}^k \text{Exp}(\omega_{\mathcal{B},z,i}^{\mathcal{B}} \Delta t) \quad (\text{S32})$$

The integration is prone to drift, however, we proved that ultimate effect caused by the accumulated integration error is negligible.

6.2 Integration error

For conciseness, we omit the subscript $\mathcal{B}$ for all $\omega_{\mathcal{B},z}^{\mathcal{B}}$ as $\omega_z^{\mathcal{B}}$. In actual situations, we can only obtain the measurement value of the z -axis angular velocity $\omega_{z,m}^{\mathcal{B}} = \omega_z^{\mathcal{B}} + \omega_{z,e}^{\mathcal{B}}$ where $\omega_{z,e}^{\mathcal{B}} = [0, 0, \omega_{z,e}^{\mathcal{B}}]^T$ is the error between the measurement and the ground truth. We can define another fictional frame $\tilde{\mathcal{F}}$ whose transformation $\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}$ is driven by $\omega_{z,m}^{\mathcal{B}}$, with derivative of $\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}$ as

$$\dot{\mathbf{R}}^{\tilde{\mathcal{F}}\mathcal{B}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}[\omega_{z,m}^{\mathcal{B}}] \quad (\text{S33})$$

Combining with the $\mathcal{F}$ frame definition $\dot{\mathbf{R}}^{\mathcal{FB}} = \mathbf{R}^{\mathcal{FB}}[\omega_z^{\mathcal{B}}]$ and the relation of coordinate transformation $\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}\mathbf{R}^{\mathcal{FB}}$, we can calculate the kinematics between $\mathcal{F}$ and $\tilde{\mathcal{F}}$ as

$$\begin{aligned} \dot{\mathbf{R}}^{\tilde{\mathcal{F}}\mathcal{F}} &= \dot{\mathbf{R}}^{\tilde{\mathcal{F}}\mathcal{B}}(\mathbf{R}^{\mathcal{FB}})^T + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}(\dot{\mathbf{R}}^{\mathcal{FB}})^T \\ &= \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}}\mathbf{R}^{\mathcal{FB}}[\omega_{z,m}^{\mathcal{B}}](\mathbf{R}^{\mathcal{FB}})^T + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}}\mathbf{R}^{\mathcal{FB}}([\omega_z^{\mathcal{B}}])^T(\mathbf{R}^{\mathcal{FB}})^T \\ &= \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}}\mathbf{R}^{\mathcal{FB}}[\omega_{z,e}^{\mathcal{B}}](\mathbf{R}^{\mathcal{FB}})^T \\ &= \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}}[\omega_{z,e}^{\mathcal{F}}] \end{aligned} \quad (\text{S34})$$

We define $\mathbf{R}_0^{\tilde{\mathcal{F}}\mathcal{F}} = \mathbf{I}$ where $\mathbf{I}$ is an identity matrix. Although $\omega_{z,e}^{\mathcal{F}}$ is very small, its integration

$$\mathbf{R}_k^{\tilde{\mathcal{F}}\mathcal{F}} = \mathbf{R}_0^{\tilde{\mathcal{F}}\mathcal{F}} \prod_{i=1}^k \text{Exp}(\omega_{z,e,i}^{\mathcal{F}} \Delta t) \quad (\text{S35})$$

still leads to an accumulated integration error.

6.3 Effect of the integration error

Since $\omega_{z,e}^{\mathcal{F}}$ is very small, we neglect it and only consider the effect of integration error $\mathbf{R}^{\tilde{\mathcal{F}}}$. Similar to the modeling process of the rotational dynamic model shown in Equation (11), we can also derive the model in the $\tilde{\mathcal{F}}$ frame and the result is

$$\mathbf{M}_{t,xy}^{\tilde{\mathcal{F}}} = [\omega_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}}] (\mathbf{I}_B^{\tilde{\mathcal{F}}} \Omega_B^{\tilde{\mathcal{F}}} + \mathbf{I}_R^{\tilde{\mathcal{F}}} \Omega_R^{\tilde{\mathcal{F}}}) + \mathbf{K}_a \omega_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}} + \mathbf{I}_O^{\tilde{\mathcal{F}}} \dot{\omega}_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}} \quad (\text{S36})$$

where $\omega_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}} \omega_z^{\mathcal{B}}$.

According to the coordinate transformation, we can transform both matrices as $\mathbf{I}_B^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_B^{\mathcal{F}} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top}$, $\mathbf{I}_R^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_R^{\mathcal{F}} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top}$, $\mathbf{I}_O^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_O^{\mathcal{F}} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top}$ and vectors as $\mathbf{M}_{t,xy}^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{M}_{t,xy}^{\mathcal{F}}$, $\Omega_B^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \Omega_B^{\mathcal{F}}$, $\Omega_R^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \Omega_R^{\mathcal{F}}$, $\omega_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}} = \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \omega_{\mathcal{F}}^{\mathcal{F}}$, and the derivative of $\omega_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}}$ as

$$\begin{aligned} \dot{\omega}_{\tilde{\mathcal{F}}}^{\tilde{\mathcal{F}}} &= \dot{\mathbf{R}}^{\tilde{\mathcal{F}}\mathcal{F}} \omega_{\mathcal{F}}^{\mathcal{F}} + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \dot{\omega}_{\mathcal{F}}^{\mathcal{F}} \\ &= \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} [\omega_{z,e}^{\mathcal{F}}] \omega_{\mathcal{F}}^{\mathcal{F}} + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \dot{\omega}_{\mathcal{F}}^{\mathcal{F}}. \end{aligned} \quad (\text{S37})$$

Then, we can substitute the above equations into Equation (S36) leading to

$$\begin{aligned} \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{M}_{t,xy}^{\mathcal{F}} &= \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} [\omega_{\mathcal{F}}^{\mathcal{F}}] (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_B^{\mathcal{F}} \Omega_B^{\mathcal{F}} + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_R^{\mathcal{F}} \Omega_R^{\mathcal{F}}) + \mathbf{K}_a \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \omega_{\mathcal{F}}^{\mathcal{F}} \\ &\quad + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \mathbf{I}_O^{\mathcal{F}} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top} (\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} [\omega_{z,e}^{\mathcal{F}}] \omega_{\mathcal{F}}^{\mathcal{F}} + \mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}} \dot{\omega}_{\mathcal{F}}^{\mathcal{F}}). \end{aligned} \quad (\text{S38})$$

Multiplying both sides by $(\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}})^{\top}$ and simplifying, we have

$$\mathbf{M}_{t,xy}^{\mathcal{F}} = [\omega_{\mathcal{F}}^{\mathcal{F}}] (\mathbf{I}_B^{\mathcal{F}} \Omega_B^{\mathcal{F}} + \mathbf{I}_R^{\mathcal{F}} \Omega_R^{\mathcal{F}}) + \mathbf{K}_a \omega_{\mathcal{F}}^{\mathcal{F}} + \mathbf{I}_O^{\mathcal{F}} \dot{\omega}_{\mathcal{F}}^{\mathcal{F}} + \mathbf{I}_O^{\mathcal{F}} [\omega_{z,e}^{\mathcal{F}}] \omega_{\mathcal{F}}^{\mathcal{F}}. \quad (\text{S39})$$

When comparing equations (11) and (S39), the only difference is the last term $\mathbf{I}_O^{\mathcal{F}} [\omega_{z,e}^{\mathcal{F}}] \omega_{\mathcal{F}}^{\mathcal{F}}$ which can be neglected due to the very small value of $\omega_{z,e}^{\mathcal{F}}$. In conclusion, although using the integration of $\omega_{z,m}^{\mathcal{F}}$ to calculate $\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{B}}$ (and deem it as $\mathbf{R}^{\mathcal{F}\mathcal{B}}$) leads to accumulated integration error of $\mathbf{R}^{\tilde{\mathcal{F}}\mathcal{F}}$, it do not affect the final dynamic model in $\mathcal{F}$ frame.

7 Disturbance observation

7.1 Work principle of disturbance observer

According to the system structure shown in Fig. 10c, we can write the output $y(s)$ as

$$\begin{aligned} y(s) &= \frac{G_n G C}{G_n (1 + GCL) + LQ(G - G_n)} r(s) \\ &\quad + \frac{G_n G (1 - LQ)}{G_n (1 + GCL) + LQ(G - G_n)} d(s) \\ &\quad - \frac{G(LQ + G_n C)}{G_n (1 + GCL) + LQ(G - G_n)} n(s) \end{aligned} \quad (\text{S40})$$

where $L(s)$ is a known low-pass filter and $Q(s)$ is a low-pass filter to be determined. For a certain low frequency range $[0, \omega_{c1}]$ satisfying $L(s) \approx Q(s) \approx 1$, the output $y(s)$ shown in Equation (S40) can be simplified as

$$y(s) = \frac{G_n C}{1 + G_n C} r(s) - n(s). \quad (\text{S41})$$

This result indicates that the output $y(s)$ is mainly determined by the nominal model $G_n(s)$ and the controller $C(s)$ without the effect of model uncertainties of $G(s)$. In this perspective, we expect that the equivalence of Equation (S41) can be valid in a frequency range as wide as possible (i.e., ω_{c1} should be large). However, the result also shows that the system is sensitive to noise since all noise $n(s)$ is added directly

to the output. Since the frequency component of noise $n(s)$ typically distributes in a certain high-frequency range, we can design $Q(s)$ to adjust ω_{c_1} such that the noise $n(s)$ included in the range $[0, \omega_{c_1}]$ is as little as possible (i.e., ω_{c_1} cannot be too large). This ensures that the $y(s)$ contains less noise component of $n(s)$.

When $L(s) \approx Q(s) \approx 0$ within a high frequency range $[\omega_{c_2}, +\infty)$ where $\omega_{c_2} > \omega_{c_1}$, the output $y(s)$ shown in Equation (S40) can be simplified as

$$y(s) = GC r(s) + Gd(s) - GCn(s). \quad (\text{S42})$$

In this high frequency range, we have $r(s) \approx 0$ and $d(s) \approx 0$ since their main components are in low-frequency range, leading to

$$y(s) = -GCn(s). \quad (\text{S43})$$

This result means that the output $y(n)$ fully depends on noise $n(s)$ and the frequency response of $G(s)C(s)$. Since $G(s)C(s)$ typically has very low gain in a certain high-frequency range $[\omega_{c_3}, +\infty)$, we can design $Q(s)$ such that $\omega_{c_2} > \omega_{c_3}$. This ensures that the noise $n(s)$ can be well attenuated by $G(s)C(s)$. In summary, the design of $Q(s)$ should ensure that the output $y(s)$ is insensitive to noise and disturbances in the low-frequency range $[0, \omega_{c_1}]$ and can well attenuate noise in the high-frequency range $[\omega_{c_2}, +\infty)$.

However, in the middle frequency range $[\omega_{c_1}, \omega_{c_2}]$, the disturbance rejection and noise attenuation cannot have the best performances at the same time. For the last two terms in Equation (S40), the gain of $d(s)$ is

$$\left\| \frac{G_n G(1 - LQ)}{G_n(1 + GCL) + LQ(G - G_n)} \right\| \quad (\text{S44})$$

which has a negative correlation with the gain of $L(s)Q(s)$. In contrast, the $Q(s)$ -related gain of $n(s)$ is

$$\left\| \frac{GLQ}{G_n(1 + GCL) + LQ(G - G_n)} \right\| \quad (\text{S45})$$

which has a positive correlation with the gain of $L(s)Q(s)$. Hence, we should ensure a good trade-off between the disturbance rejection and the noise attenuation, since their performances are competitive.

7.2 Nominal plant model and its uncertainties

The DOB design depends on the model in the frequency domain, and therefore we need to establish it according to the rotational dynamics and the identified transfer function of the actuator. For the $x - y$ rotational dynamic model Equation (11), we can rewrite it as

$$\mathbf{M}_{xy}^{\mathcal{F}} = [\omega_{\mathcal{F}}^{\mathcal{F}}] \mathbf{L}_z^{\mathcal{F}} + \mathbf{K}_a \omega_{\mathcal{F}}^{\mathcal{F}} + \mathbf{I}_O^{\mathcal{F}} \dot{\omega}_{\mathcal{F}}^{\mathcal{F}} \quad (\text{S46})$$

where the z -axis angular momentum $\mathbf{L}_z^{\mathcal{F}} = \mathbf{I}_B^{\mathcal{F}} \boldsymbol{\Omega}_B^{\mathcal{F}} + \mathbf{I}_R^{\mathcal{F}} \boldsymbol{\Omega}_R^{\mathcal{F}}$. Since $\mathbf{L}_z^{\mathcal{F}} = [0, 0, L_z^{\mathcal{F}}]^{\top}$ and $\boldsymbol{\Omega}_B^{\mathcal{F}}$ and $\boldsymbol{\Omega}_R^{\mathcal{F}}$ only have z -axis component, we can only calculate the z -axis scalar as

$$L_z^{\mathcal{F}} = I_{B,z}^{\mathcal{F}} \Omega_{B,z}^{\mathcal{F}} + I_{R,z}^{\mathcal{F}} \Omega_{R,z}^{\mathcal{F}}. \quad (\text{S47})$$

Further, the component-wise representation of Equation (S46) is

$$\begin{bmatrix} M_x^{\mathcal{F}} \\ M_y^{\mathcal{F}} \end{bmatrix} = \begin{bmatrix} I_{O,x}^{\mathcal{F}} & 0 \\ 0 & I_{O,y}^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{\mathcal{F},x}^{\mathcal{F}} \\ \dot{\omega}_{\mathcal{F},y}^{\mathcal{F}} \end{bmatrix} + \begin{bmatrix} k_a & L_z^{\mathcal{F}} \\ -L_z^{\mathcal{F}} & k_a \end{bmatrix} \begin{bmatrix} \omega_{\mathcal{F},x}^{\mathcal{F}} \\ \omega_{\mathcal{F},y}^{\mathcal{F}} \end{bmatrix} \quad (\text{S48})$$

Since $L_z^{\mathcal{F}}$ varies slowly within a certain range, we approximated it as a constant. Applying the Laplace transform, we can represent the rotational model in the frequency domain as

$$\omega_{\mathcal{F}}^{\mathcal{F}}(s) = \mathbf{G}_r(s) \mathbf{M}_{xy}^{\mathcal{F}}(s) \quad (\text{S49})$$

Supplementary Table 1: Nominal parameters and uncertain ranges of the uncertain plant model $\mathbf{G}(s)$.

Parameter name	Symbol	Unit	Nominal value	Uncertain range
Self-rotation speed (z component)	$\Omega_{B,z}^{\mathcal{F}}$	deg/s	-720	[-1200, -500]
Propeller rotation speed (z component)	$\Omega_{R,z}^{\mathcal{F}}$	rpm	5500	[3500, 8000]
Inertia of body part (z component)	$I_{B,z}^{\mathcal{F}}$	kg · m ²	6.2978×10^{-3}	$[6.0 \times 10^{-3}, 6.6 \times 10^{-3}]$
Inertia of rotor part (z component)	$I_{R,z}^{\mathcal{F}}$	kg · m ²	8.179×10^{-5}	$[6.0 \times 10^{-5}, 12.0 \times 10^{-5}]$
Angular momentum (z component)	$L_z^{\mathcal{F}} = I_{B,z}^{\mathcal{F}}\Omega_{B,z}^{\mathcal{F}} + I_{R,z}^{\mathcal{F}}\Omega_{R,z}^{\mathcal{F}}$	kg · m ² /s	-0.0472	[-0.1277, 0.0705]
Inertia of whole UAV (x and y component)	$J = I_{O,x}^{\mathcal{F}} = I_{O,y}^{\mathcal{F}}$	kg · m ²	6.6×10^{-3}	$[6.3 \times 10^{-3}, 6.9 \times 10^{-3}]$
Coefficient of air damping moment	k_a	N · m/s	2.6×10^{-3}	$[1.6 \times 10^{-3}, 3.6 \times 10^{-3}]$

with the transfer function matrix $\mathbf{G}_r(s)$ as

$$\mathbf{G}_r(s) = \begin{bmatrix} \frac{Js+k_a}{J^2s^2+2k_aJs+k_a^2+(L_z^{\mathcal{F}})^2} & \frac{-L_z^{\mathcal{F}}}{J^2s^2+2k_aJs+k_a^2+(L_z^{\mathcal{F}})^2} \\ \frac{L_z^{\mathcal{F}}}{J^2s^2+2k_aJs+k_a^2+(L_z^{\mathcal{F}})^2} & \frac{Js+k_a}{J^2s^2+2k_aJs+k_a^2+(L_z^{\mathcal{F}})^2} \end{bmatrix} \quad (\text{S50})$$

where the rotational inertia $J = I_{O,x}^{\mathcal{F}} = I_{O,y}^{\mathcal{F}}$.

Combining with the identified transfer function matrix of the actuator $\mathbf{G}_a(s)$ in Equation (S11), we can define the total transfer function matrix (i.e., the nominal plant model $\mathbf{G}_n(s)$) from the desired moment $\mathbf{M}_{xy,d}^{\mathcal{F}}(s)$ to angular velocity $\boldsymbol{\omega}_{\mathcal{F}}^{\mathcal{F}}(s)$, which is

$$\begin{aligned} \boldsymbol{\omega}_{\mathcal{F}}^{\mathcal{F}}(s) &= \mathbf{G}_r(s)\mathbf{M}_{xy}^{\mathcal{F}}(s) \\ &= \mathbf{G}_r(s)\mathbf{G}_a(s)\mathbf{M}_{xy,d}^{\mathcal{F}}(s) \\ &= \mathbf{G}_n(s)\mathbf{M}_{xy,d}^{\mathcal{F}}(s) \end{aligned} \quad (\text{S51})$$

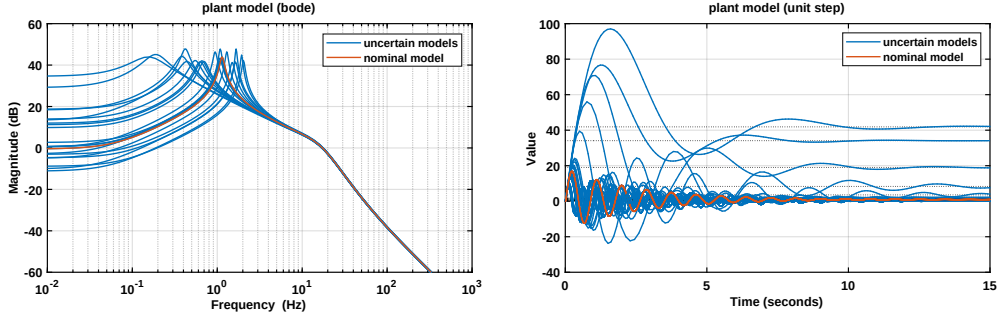
where $\mathbf{G}_n(s) = \mathbf{G}_r(s)\mathbf{G}_a(s)$.

Note that $\mathbf{G}_n(s)$ is the nominal plant model without uncertainty. In the DOB design, we used the uncertain plant model $\mathbf{G}(s)$, namely $\mathbf{G}_n(s)$ with parameter uncertainties. The nominal parameters of $\mathbf{G}_n(s)$ and the uncertainty ranges of $\mathbf{G}(s)$ are listed in the Supplementary Table 1. For all inertial parameters and the coefficient of air damping moment, their nominal values are identified from experimental data, and their identification errors are considered within the uncertain ranges. For the rotational speeds of the propeller and UAV body, the nominal values are measured in hover flight, and the uncertain ranges reflect their value distribution during other flight states. It should be noted that the uncertainties in $\mathbf{G}(s)$ originate entirely from the rotational model $\mathbf{G}_r(s)$. This is because the actuator model $\mathbf{G}_a$ is identified from experimental data obtained under typical conditions in the test stand, which do not account for uncertainties induced by airflow and aerodynamic disturbances—factors that are difficult to replicate in a test-stand environment. Such uncertainties are typically concentrated in the low-frequency range and, as analyzed in Supplementary Note 7.1, are expected to be effectively handled by the DOB.

The Bode diagram and the unit step responses of both the nominal model and the uncertain model are shown in Supplementary Fig. 9. There are multiple curves of the uncertain model due to parameter uncertainties. Each curve represents a group of parameters sampled within their uncertain ranges. As we can see, the uncertain models are very different from the nominal model, indicating that the uncertainties obviously affect the frequency characteristics as well as the unit step responses in the time domain.

7.3 Inverse of the nominal plant model

As shown in Fig. 10c, the DOB structure comprises the inverse of the nominal plant model $\mathbf{G}_n^{-1}(s) = \mathbf{G}_a^{-1}(s)\mathbf{G}_r^{-1}(s)$ which is used to calculate the actual input $\mathbf{u}_a$ based on the measured output $\mathbf{y}_{nl}$. The inversion of $\mathbf{G}_r(s)$ can be directly obtained by



Supplementary Fig. 9: The Bode diagrams and unit step responses of the nominal plant model and the uncertain plant model.

inverting the transfer function matrix $\mathbf{G}_r(s)$ as

$$\mathbf{G}_r^{-1}(s) = \begin{bmatrix} \frac{J^2 s^2 + 2k_a J s + k_a^2 + (L_z^F)^2}{J s + k_a} & \frac{L_z^F}{J s + k_a} \\ \frac{-L_z^F}{J s + k_a} & \frac{J^2 s^2 + 2k_a J s + k_a^2 + (L_z^F)^2}{J s + k_a} \end{bmatrix} \quad (\text{S52})$$

Whereas the inversion of $\mathbf{G}_a$ is

$$\mathbf{G}_a^{-1}(s) = \frac{s^2 + 121.9s + 11060}{-27.97s + 9028} \quad (\text{S53})$$

which is unstable since its denominator has an RHP pole (details can refer to Supplementary Note 2).

The low-pass filter $\mathbf{Q}(s)$ determines the overall performance of the DOB. Here, for simplicity, we directly specify $\mathbf{Q}(s)$ as a diagonal matrix

$$\mathbf{Q}(s) = \begin{bmatrix} Q(s) & 0 \\ 0 & Q(s) \end{bmatrix}. \quad (\text{S54})$$

Since each block inside the DOB structure must be stable and causal, we need to ensure the entirety of $\mathbf{Q}(s)\mathbf{G}_n^{-1}(s)$ satisfying: (i) it has no RHP pole (for stability) and (ii) the order of the numerator is less than or equal to the order of the denominator (for causality), otherwise it is not realizable and cannot be optimized by μ synthesis in the next section. To do this, we fixed the numerator of each element of $\mathbf{Q}(s)$ with the same RHP zero [71] to cancel out the RHP pole such that the entirety of $\mathbf{Q}(s)\mathbf{G}_n^{-1}(s)$ is causal. As a result, $Q(s)$ has the following form

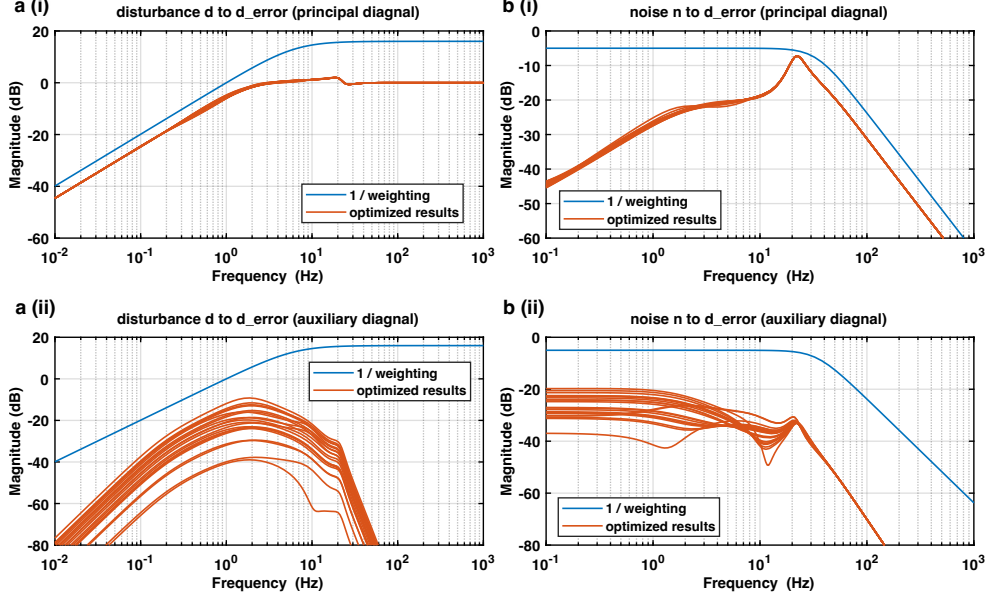
$$Q(s) = \frac{-27.97s + 9028}{(\sum_{i=1}^q c_i s^i) + 9028} \quad (\text{S55})$$

where q is the order of the denominator of $Q(s)$ to ensure that $\mathbf{Q}(s)\mathbf{G}_n^{-1}(s)$ is causal, c_i is the coefficients to be optimized, and the DC gain of $Q(s)$ is fixed to 0 dB to avoid undesired attenuation of low-frequency components.

8 Optimization results and performance comparison of μ synthesis

8.1 Optimization result

The optimization result of μ synthesis shows that the robust performance is $\mu_{\text{perf}} = 0.957$, which means that the closed-loop gain remains below 0.957 even when the uncertainty is up to $\frac{1}{0.957}$ times the value specified in the plant model. Typically, in actual situations, the worst-case closed-loop gain for the specified uncertainty should be smaller, and hence the optimized result can well ensure the robustness of the system.



Supplementary Fig. 10: Bode plots of the reciprocal weighting functions (blue curves) and the optimized results of $\mathbf{T}_d(s)$ and $\mathbf{T}_n(s)$ via μ synthesis (orange curves). **a** Transfer functions from disturbance d to disturbance estimation error d_{error} . (i) First element of the principal diagonal. (ii) First element of the auxiliary diagonal. **b** Transfer functions from noise n to disturbance estimation error d_{error} . (i) First element of the principal diagonal. (ii) First element of the auxiliary diagonal.

The optimal $Q_{\text{opt}}(s)$ obtained via μ synthesis is

$$Q_{\text{opt}}(s) = \frac{-27.97s + 9028}{0.03937s^3 + 2.437s^2 + 766.3s + 9028}. \quad (\text{S56})$$

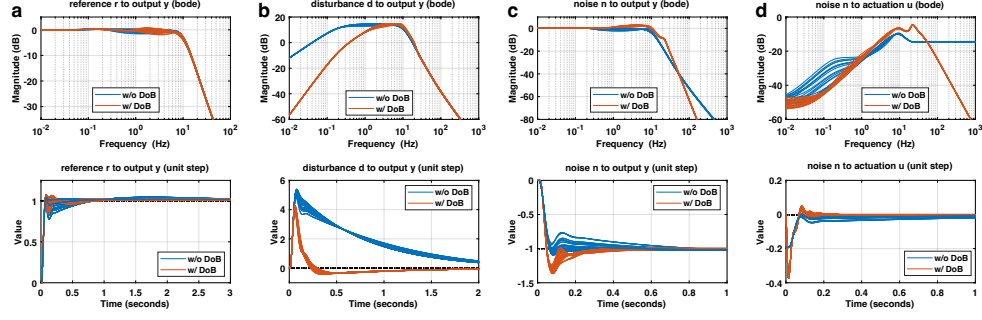
With $Q_{\text{opt}}(s)$, we can evaluate the results of $\mathbf{T}_d(s)$ and $\mathbf{T}_n(s)$. The Bode diagrams of the first element of the principal diagonal and the auxiliary diagonal are shown in Supplementary Fig. 10, respectively. There are many curves for the optimized result since the Bode diagram is plotted with different sampled parameters among the uncertainty ranges of the plant model. As we can see, all the curves of the optimized result are below the reciprocal weighting function, which means that the performance of disturbance rejection and noise attenuation is well satisfied for all given uncertainties.

8.2 Performance comparison

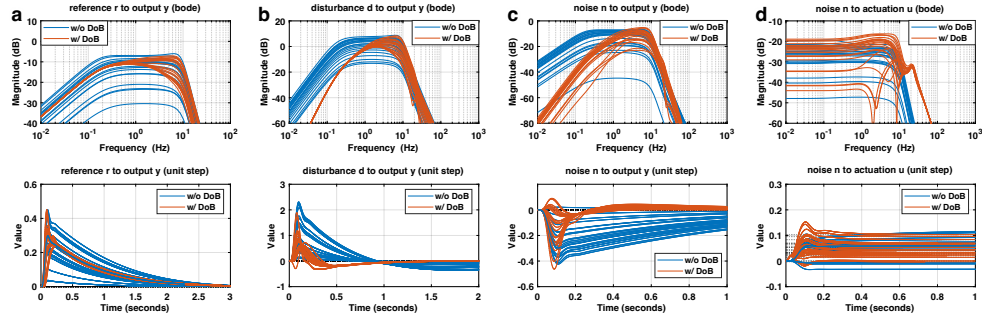
The performance comparisons of the system with and without DOB are shown in Supplementary Fig. 11 and Supplementary Fig. 12.

Supplementary Fig. 11 shows the Bode diagram and unit step response of the first element of the principal diagonal of the transfer function matrix, which dominates the system's performance. In Supplementary Fig. 11a, with the same uncertainty, the system with DOB shows more consistent Bode diagrams and shorter settling times of step responses, indicating that the DOB can recover the performance of the nominal system. For the disturbance rejection performance shown in Supplementary Fig. 11b, the system with DOB shows a much lower gain in the low-frequency range and much faster suppression of the unit step disturbance. In Supplementary Fig. 11(c-d), we evaluated noise attenuation performance from n to y (to avoid vibration of angular velocity) and from n to u (to avoid affecting the SLM modulation). Due to the trade-off of disturbance performance, the gain of the system with DOB is not always smaller than that of the system without DOB. However, for the frequency range where the noise distributes (i.e., 58–133 Hz), the DOB system has a smaller gain and a larger descent slope, which means the propeller noise can be better attenuated.

Supplementary Fig. 12 shows the Bode diagram and unit step response of the first element of the auxiliary diagonal of the transfer function matrix. Since it is a cross



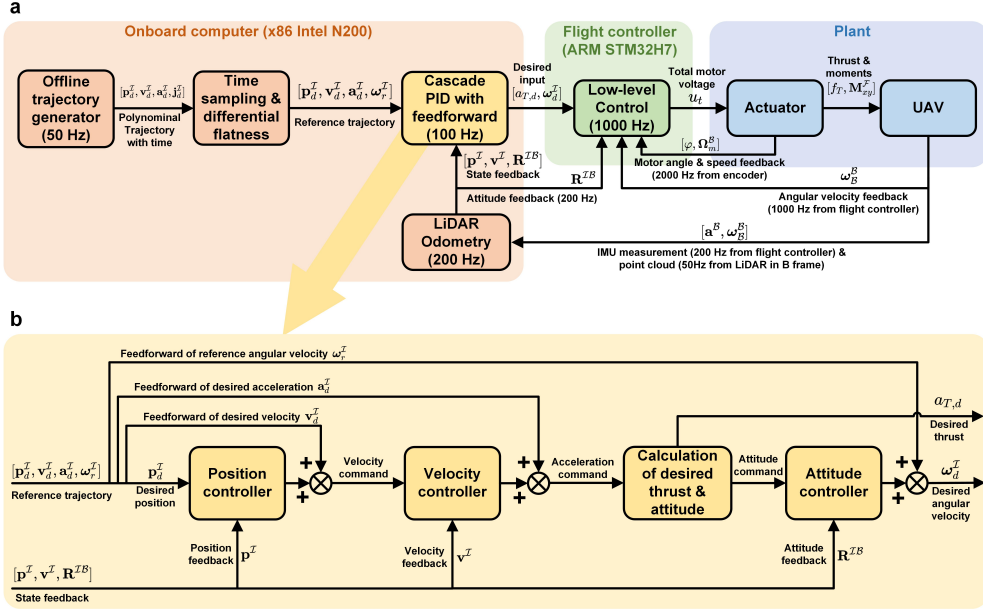
Supplementary Fig. 11: Bode diagrams and their corresponding unit step responses in time domain (the first element of principal diagonal of the transfer function matrix). **a** From reference r to output y . **b** From disturbance d to output y . **c** From noise n to output y . **d** From noise n to actuation u .



Supplementary Fig. 12: Bode diagrams and their corresponding unit step responses in time domain (the first element of auxiliary diagonal of the transfer function matrix). **a** From reference r to output y . **b** From disturbance d to output y . **c** From noise n to output y . **d** From noise n to actuation u .

term, smaller gain is better for decoupling. In Supplementary Fig. 12(a-c), the system with DOB achieved a smaller gain in the Bode diagram as well as more consistent curves and a shorter converging time in the unit step response. In Supplementary Fig. 12d, although the systems with DOB have larger gains when the model parameters are sampled from some values within the uncertainty ranges, the highest gain is still lower than -16 dB, which is sufficient for noise attenuation.

Supplementary Figures



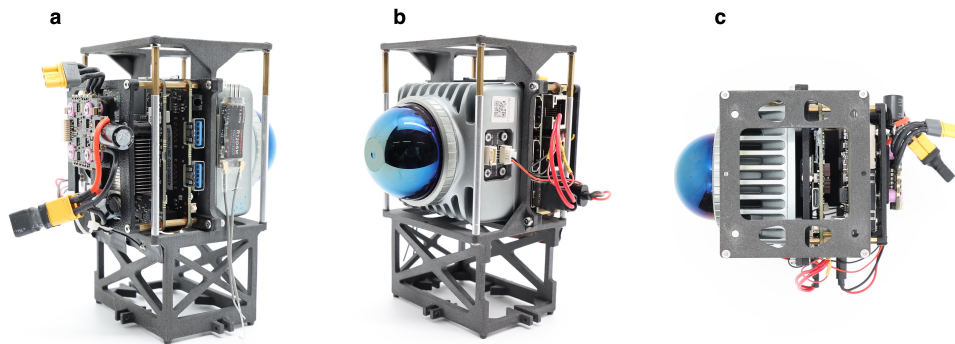
Supplementary Fig. 13: System structure of the cascade PID with feedforward. **a** Entire system for trajectory tracking control. **b** Internal structure of the cascade PID with feedforward.



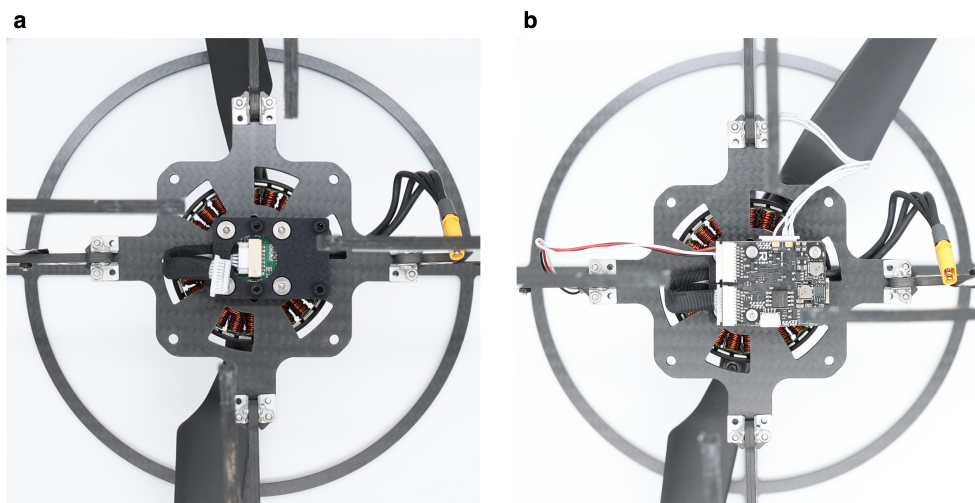
Supplementary Fig. 14: Photo of entire structure of UAV.



Supplementary Fig. 15: Photo of the separate parts of UAV.



Supplementary Fig. 16: Different views of the main structure of UAV. **a** Side view I. **b** Side view II. **c** Top view.



Supplementary Fig. 17: Installation details of the magnetic encoder and the FCU. **a** Magnetic encoder is installed on the carbon fiber board, which is very close to the bottom of motor. **b** FCU is installed close to the magnetic encoder to shorten the SPI connection line.

Nomenclature

$\mathcal{B}$	Body coordinate frame with origin at the UAV's center of mass.
$\mathcal{F}$	Fictional coordinate frame without self-rotation and with origin at the UAV's center of mass.
$\mathcal{I}$	Inertial (ground-fixed) coordinate frame.
$[\cdot]$	Generates a skew-symmetric matrix by a vector.
$(\cdot)^{\mathcal{X}}$	Superscript $\mathcal{X}$ indicates in which coordinate frame the variable is represents. $\mathcal{X}$ could be $\mathcal{B}$, $\mathcal{F}$, or $\mathcal{I}$.
$(\cdot)_i$	A vector with non-zero element in i axis ($i = x, y, z$) and other axes are zero.
$(\cdot)_{ij}$	A vector with non-zero element in i and j axes ($ij = xy, yz, xz$) and the remained axis is zero.
$\mathbf{R}^{\mathcal{Y}\mathcal{X}}$	A rotation matrix that can conduct a coordinate transform from frame $\mathcal{X}$ to frame $\mathcal{Y}$.
$\mathbf{I}_O^{\mathcal{F}}$	Inertia matrices of the whole UAV represented in $\mathcal{F}$ frame.
$\mathbf{I}_B^{\mathcal{F}}$	Inertia matrices of the UAV's body (body part) represented in $\mathcal{F}$ frame.
$\mathbf{I}_R^{\mathcal{F}}$	Inertia matrices of the UAV's propeller and motor rotor (rotor part) represented in $\mathcal{F}$ frame.
$I_{O,i}^{\mathcal{F}}$	Three diagonal elements of $\mathbf{I}_O^{\mathcal{F}}$ and $i = x, y, z$.
$(\cdot)_d$	The subscript s indicates it is a desired value.
$(\cdot)_m$	The subscript m indicates it is a measured value.
$(\cdot)_k$	The subscript k indicates it is at k -th step.
$\mathbf{x}$	States of the state space model.
$\mathbf{u}$	Inputs of the state space model.
$\mathbf{e}_p^{\mathcal{X}}$	Integration of position error vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\mathbf{p}^{\mathcal{X}}$	Position vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\mathbf{v}^{\mathcal{X}}$	Velocity vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\mathbf{a}^{\mathcal{X}}$	Acceleration vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\mathbf{j}^{\mathcal{X}}$	Jerk vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\mathbf{n}^{\mathcal{X}}$	Direction of z axis (unit vector) $\in S^2$ of $\mathcal{F}$ frame represented in $\mathcal{X}$ frame.
$\boldsymbol{\omega}_{\mathcal{X}}^{\mathcal{X}}$	Angular velocity $\in \mathbb{R}^3$ with subscript $\mathcal{X}$ indicating which coordinate frame it belongs to and superscript indicating which coordinate frame represented in.
a_T	Acceleration $\in \mathbb{R}^1$ generated by the motors' thrust.
$\mathbf{g}^{\mathcal{X}}$	Constant gravity vector $\in \mathbb{R}^3$ represented in $\mathcal{X}$ frame.
$\boldsymbol{\Omega}_R^{\mathcal{X}}$	Angular velocity of rotor part represented in $\mathcal{X}$ frame.
$\boldsymbol{\Omega}_B^{\mathcal{X}}$	Angular velocity of body part represented in $\mathcal{X}$ frame.
$\boldsymbol{\Omega}_m^{\mathcal{X}}$	Motor Angular velocity measured by encoder and represented in $\mathcal{X}$ frame.
$\mathbf{L}^{\mathcal{X}}$	Angular momentum vector represented in $\mathcal{X}$ frame.
$\mathbf{M}_t^{\mathcal{X}}$	Total moment vector represented in $\mathcal{X}$ frame.
$\mathbf{M}_a^{\mathcal{X}}$	Aerodynamic moment vector induced by air drag represented in $\mathcal{X}$ frame.
$\mathbf{M}_{xy,d}^{\mathcal{X}}$	Desired moment vector in xy plane represented in $\mathcal{X}$ frame.
$\mathbf{M}_{xy,c}^{\mathcal{X}}$	Moment vector in xy plane calculated by controller and represented in $\mathcal{X}$ frame.
$\mathbf{M}_{xy}^{\mathcal{X}}$	Moment vector in xy plane generated by actuator and represented in $\mathcal{X}$ frame.
$\mathcal{D}_x(\cdot)$	Function mapping from flatness outputs to system states.
$\mathcal{D}_u(\cdot)$	Function mapping from flatness outputs to system inputs.
Δt	Time interval of discretization.
u_t	Total voltage of motor.
u_n	Nominal voltage of motor to generate thrust.
u_a	Amplitude of the sinusoidal voltage.
φ	Motor angle measured by encoder.
γ	Rotor angle where the acceleration begins (i.e., rotor acceleration position).
β	Direction of the generated moment of the swashplateless mechanism.
λ	Lag angle of the moment direction caused by the inertia of swashplateless mechanism and the encoder measurement delay.
f_T	Thrust of the swashplateless mechanism.
$\delta \mathbf{x}_k$	Error states in k step.
$\delta \mathbf{u}_k$	Error inputs in k step.
$\mathbf{Q}_k$	Weighting matrices for error states MPC optimization at .

$\mathbf{R}_k$	Weighting matrices for error inputs MPC optimization.
$\mathbf{F}_{\mathbf{x}_k}$	System matrix of the linearized error-state system.
$\mathbf{F}_{\mathbf{u}_k}$	Input matrix of the linearized error-state system.
N	MPC prediction horizon.
$\mathbf{Q}_w$	A sequence of waypoints.
$\mathbf{W}_u$	Positive diagonal matrix penalizing the control effort for trajectory optimization.
T_f	Duration time of trajectory.
ρ	Penalty of duration time of trajectory.
$\mathbf{G}(s)$	Transfer function matrix of the plant model with parameter uncertainties.
$\mathbf{G}_n(s)$	Transfer function matrix of the nominal model without uncertainty.
$\mathbf{G}_a(s)$	Transfer function matrix of the actuator model.
$\mathbf{G}_r(s)$	Transfer function matrix of rotational model of the whole UAV.
$\mathbf{C}(s)$	Transfer function matrix of the typical proportional-integral (PI) controller with tuned parameter.
$\mathbf{Q}(s)$	Transfer function matrix of a low-pass filter.
$\mathbf{P}(s)$	Generalized plant model for μ synthesis.
$\mathbf{W}(s)$	Generalized disturbance input for μ synthesis.
$\mathbf{U}(s)$	Generalized control input for μ synthesis.
$\mathbf{Z}(s)$	Error output for μ synthesis.
$\mathbf{Y}(s)$	Measurement output for μ synthesis.
$\mathbf{K}(s)$	Equivalent structure of disturbance observer for μ synthesis.
$\mathbf{W}_d(s)$	Frequency-dependent weighting functions for disturbance performance in μ synthesis.
$\mathbf{W}_n(s)$	Frequency-dependent weighting functions for noise performance in μ synthesis.
$\mathbf{T}_d(s)$	Transfer function matrix from disturbance input to estimated disturbance.
$\mathbf{T}_n(s)$	Transfer function matrix from noise input to estimated disturbance.
$\bar{\omega}$	Represent the angular velocity $\omega_{\mathcal{X}}^{\mathcal{X}} \in \mathbb{R}^3$ in the $\mathbb{R}^2$ tangent space of a point $\mathbf{n}^{\mathcal{X}} \in S^2$.
$\mathbf{B}(\cdot)$	Projection matrix generated by $\mathbf{n}^{\mathcal{X}}$, which transforms a vector $\in \mathbb{R}^3$ to the tangent space of $\mathbf{n}^{\mathcal{X}}$ as a vector $\in \mathbb{R}^2$.
$\boxplus$	$S^2 \times \mathbb{R}^2 \mapsto S^2$, adding the vector $\in \mathbb{R}^2$ in the tangent space to a point on $\in S^2$ manifold to generate a new point on $\in S^2$ manifold.
$\boxminus$	$S^2 \times S^2 \mapsto \mathbb{R}^2$, calculating the error between two points on the S^2 manifold and generating the result as a vector $\in \mathbb{R}^2$ in the tangent space.
$\text{Exp}(\cdot)$	A mapping defined as $\exp([\cdot])$, transforming a vector $\in \mathbb{R}^3$ to a point $\in SO^3$.

References

- [68] He, D., Xu, W., Zhang, F.: Symbolic representation and toolkit development of iterated error-state extended kalman filters on manifolds. *IEEE Transactions on Industrial Electronics* **70**(12), 12533–12544 (2023)
- [69] He, D., Xu, W., Zhang, F.: Kalman filters on differentiable manifolds. *arXiv preprint arXiv:2102.03804* (2021)
- [70] Marhalling / communication library for drones (2024). <https://github.com/mavlink/mavlink> Accessed 2025-04-27
- [71] Wang, L., Su, J.: Disturbance rejection control for non-minimum phase systems with optimal disturbance observer. *ISA transactions* **57**, 1–9 (2015)