International Journal of Micro Air Vehicles

ORIGINAL RESEARCH ARTICLE

Open Access

https://doi.org/10.48130/mav-0026-0001

International Journal of Micro Air Vehicles **2026**, 18: e001

An adaptive hierarchical control and sensor fusion strategy for an intelligent multirotor UAV landing system

Mingyang Huang[1,2,3*] and Weize Zhang[1,2,3]

[1] *Key Laboratory of Intelligent Bionic Unmanned Systems, Ministry of Education, University of Science and Technology Beijing, Beijing 100083, China*
[2] *School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, China*
[3] *Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China*
* Correspondence: huang@ustb.edu.cn (Huang M)

Abstract

This paper introduces an adaptive hierarchical control framework with multimodal sensing for reliable multirotor unmanned aerial vehicle (UAV) landings in unstructured environments. A bio-inspired rocker-based landing mechanism is designed to minimize control coupling and provide terrain adaptation redundancy. We develop a Lagrangian dynamic model with real-time attitude perception and an adaptive sliding mode control (ASMC) strategy, supported by Lyapunov stability analysis for estimating the disturbance and a bounded tracking error. For terrain perception, a light detection and ranging (LiDAR)-based dynamic terrain flatness scanning and fusion (DTFSF) algorithm achieves over 96% recognition accuracy on basic terrains with a 1-ms processing time across various scenarios. Simulations show that the ASMC reduces the root mean square error by 77.0% compared with proportional–integral–derivative (PID) control, with a 2.2% increase in energy consumption, while maintaining robustness to faults in the actuator and sensor. Field tests confirm its real-time operation at 1 kHz on Cortex-M7 embedded platforms, with power consumption below 20 W.

Citation: Huang M, Zhang W. 2026. An adaptive hierarchical control and sensor fusion strategy for an intelligent multirotor UAV landing system. *International Journal of Micro Air Vehicles* 18: e001 https://doi.org/10.48130/mav-0026-0001

Introduction

The development of adaptive landing gear for rotorcraft has gained increasing interest, motivated by the demand for safer operations in dynamic environments. The initial designs concentrated on passive systems such as spring dampers, which are simple and low-cost yet cannot adjust to varying terrains. Subsequent semiactive systems, including magnetorheological dampers, improved energy dissipation but still offered limited real-time adaptability. Fully active systems using hydraulic or pneumatic actuation enabled in-flight adjustments, though they often required high energy and complex control architectures.

Internationally, research has drawn inspiration from the leg structures of birds and insects to design landing systems for unmanned aerial vehicles (UAVs) operating in complex terrain. For example, the Defense Advanced Research Projects Agency (DARPA) in the US developed a multijoint robotic leg that adjusts the leg angles in real time using tactile sensors, allowing the body to remain level on a 20° slope[1]. However, such designs remain vulnerable to lateral impacts. Similarly, bio-inspired leg and claw mechanisms have been developed at Stanford University[2], Skolkovo Institute[3], the University of Tokyo[4], Eidgenössische Technische Hochschule (ETH) Zurich[5], and the University of Edinburgh's Napier University[6]. A recurring limitation across these systems is their structural vulnerability under lateral forces. Integrating buffering components could help distribute the impact loads, but current research has not sufficiently addressed the dynamic interaction between the landing mechanisms and complex terrains. Most systems have been validated only on flat surfaces, limiting their applicability in unstructured environments.

Multisensor fusion has become a key enabler for improving reliability and environmental adaptability[7]. The related research approaches fall into three main categories: Deep learning-based feature fusion, adaptive fault-tolerant control[8], and real-time processing techniques.

In deep learning-driven fusion, efforts focus on integrating multimodal data and advanced feature extraction. Yang et al.[9] introduced a local–global fusion architecture using sparse transformers to handle limited training samples. Xu et al.[10] combined Principal Component Analysis (PCA)-based sensor fusion with spiking neural networks. Parsons *et al.*[11] designed a dual-channel convolutional neural network (CNN) for light detection and ranging (LiDAR), and red–green–blue (RGB) data, improving object detection robustness. Lv et al.[12] proposed a hierarchical residual long short-term memory (LSTM) with dynamic attention. Though such neural network methods capture intricate data patterns, their computational demands often hinder real-time performance in UAV landing applications.

In fault-tolerant control, Sun et al.[13] developed an adaptive tracking control framework using historical data-driven weighting to counteract sensor faults and disturbances. Wu et al.[14] integrated adaptive control with sensor fusion in a variable-weight Model Predictive Control (MPC) structure. However, many methods treat sensor errors as disturbances and may fail if the controller becomes unstable, highlighting the need for modular designs.

For real-time processing, researchers have emphasized computational efficiency and noise suppression. Hussain et al.[15] compared fusion methods for space target tracking, showing a balance between accuracy and speed. Wang et al.[16] proposed a distributed multirate fusion algorithm using sequential Weighted Minimum Mean Square Error (WMMSE) estimation to handle unknown noise correlations. Xu et al.[17] incorporated state saturation and the sensors' resolution into a covariance intersection framework. These optimization methods perform well with limited data and improve systems' efficiency.

Recent perception-driven landing strategies combine vision, LiDAR, and Inertial Measurement Unit (IMU) data to improve

landing precision. Ban et al.[18] fused camera, LiDAR, and IMU data with green feature mapping and point cloud fusion. Sestras et al.[19] integrated LiDAR and photogrammetric data for robust terrain modeling. Liu et al.[20] developed a semantic Simultaneous Localization and Mapping (SLAM) framework with object-level clustering. Saldiran et al.[21] created a LiDAR-based grid-processing system for emergency landings under critical failures. Chen et al.[22] implemented multisensor SLAM for parafoil UAVs in complex terrains. Jiao et al.[23] achieved real-time centimeter-accurate semantic mapping via LiDAR–visual–inertial fusion. Despite these advances, such methods lack the adaptive hierarchical control needed to balance the computational load and precision across dynamic landing scenarios.

For rotorcraft landing systems requiring high tracking accuracy and speed, linear control methods[24] may be inadequate. Sliding mode control (SMC) has gained attention, as it combines the simplicity of proportional–integral–derivative (PID) control with bounded error and exponential convergence. SMC drives a system's states to a predefined sliding surface in finite time and offers robustness against disturbances and parameter variations.

Although SMC for linear systems has been widelystudied[25,26], its application to nonlinear systems like rotorcrafts' landing gear remains challenging, mainly because of chattering—high-frequency oscillations caused by discontinuous control actions. Recent studies have proposed techniques for mitigating chattering. Labbadi et al.[27] applied backstepping to derive error dynamics and designed switching laws to suppress chattering[28,29]. Irfan et al.[30] developed chattering-free SMC for a twin-rotor UAV. Robust SMC schemes can handle complex nonlinear dynamics[31], and saturation-constrained SMC is suitable for mechanical systems[32]. A fast integral terminal SMC has also been proposed for robotic systems[33]. Though innovative, such methods still lack the speed and reliability needed for landing systems' decision-making.

In summary, a significant research gap persists in unifying robust mechanical designs, real-time perception, and adaptive control into a cohesive system capable of handling uncertainties in complex landing scenarios. To address this, the present study introduces the following:

(1) A bio-inspired linkage-based landing mechanism with a tri-leg configuration that simplifies control, ensures redundancy, and favors axial loading for improved terrain adaptation;

(2) A sliding-window terrain perception algorithm, known as the dynamic terrain flatness scanning and fusion (DTFSF) model, using LiDAR/infrared sensors and IMU for real-time, noise-resistant terrain identification on embedded platforms;

(3) A Lyapunov-stabilized adaptive sliding mode controller (ASMC) robust to model the uncertainties, partial actuator failures, and external disturbances;

(4) An adaptive hierarchical control framework (AHCF) that integrates mechanical, perception, and control layers to enable autonomous landing in complex environments.

The integrated system's architecture

Morphological design of the adaptive landing mechanism

In large aircraft, the landing gear is categorized into three types depending on the shock absorbers' position and the load-bearing mode: Strut-type, semilevered, and levered. In levered landing gear, the shock absorber withstands axial forces only. This configuration enhances the resistance to forward impact loads, improves ground taxiing stability, reduces bushing friction, decreases wear, and improves the sealing performance. However, it is relatively heavy and requires more space for retraction.

Considering the trade-off between weight and flexibility, we developed a rocker-actuated adaptive landing gear for an octocopter UAV (Fig. 1). Inspired by levered landing gear, a servo-driven lead screw actuator adjusts the landing legs' angle. The drive mechanism withstands axial forces only and forms a stable triangular configuration with the airframe and landing legs. A single landing leg can be simplified into a planar two-bar constrained structure, with only one constraint equation (Eq. [1]), thus resulting in one degree of freedom (DoF).

The entire landing system consists of two rocker-actuated landing leg and one fixed landing leg (see Fig. 2), arranged in a triangular formation when viewed from the vertical direction. This configuration ensures that the system contacts the ground at three different heights, allowing the rotorcraft to achieve a stable landing on certain complex terrains. Under these conditions, the number of control variables in the system is reduced to two, thereby

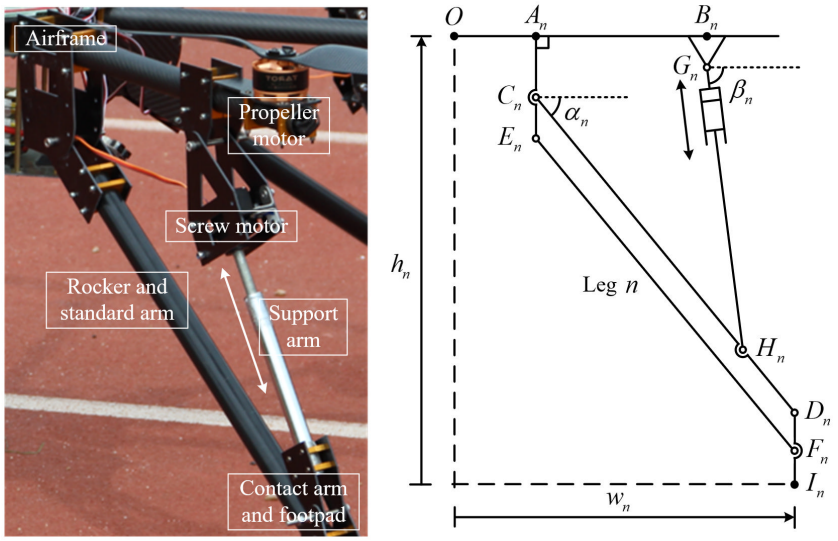

Fig. 1 Single rocker-actuated landing leg.

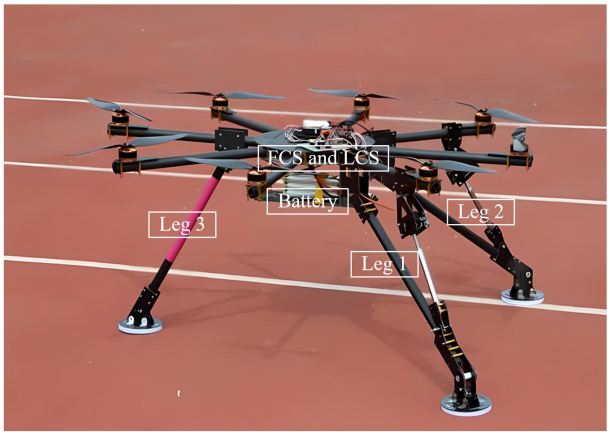

Fig. 2 Movable legs (Nos. 1–2, black) and fixed leg (No. 3, red).

minimizing the weight of the motors and increasing the payload capacity of the rotorcraft. The system exhibits high flexibility, as the fixed landing leg can be removed, leaving three rocker-actuated landing legs. This ensures that the system can still land on complex terrains even if one leg fails, providing robustness.

Based on the mechanism schematic shown in Fig. 1, the kinematic equations (Eqs. [1]–[4]) relating the motor's rotation to the terminal horizontal/vertical motion of the rocker-actuated landing leg can be derived. The angles α_n, β_n represent the angles between the rocker arm C_nD_n and the support arm G_nH_n, respectively, and the drone's horizontal plane. The distances w_n, h_n denote the width and height of the landing leg tip, measured with respect to the drone's center axis as the zero point. L with a subscript represents the length of the arm.

$$\alpha_n = \cos^{-1}\frac{L_{C_nH_n}^2 + L_{A_nB_n}^2 + (L_{A_nC_n} - L_{B_nG_n})^2 - L_{G_nH_n}^2}{2L_{C_nH_n}\sqrt{L_{A_nB_n}^2 + (L_{A_nC_n} - L_{B_nG_n})^2}} - \tan^{-1}\frac{L_{A_nC_n} - L_{B_nG_n}}{L_{A_nB_n}} \tag{1}$$

$$\beta_n = \cos^{-1}\frac{L_{C_nH_n}^2 - L_{A_nB_n}^2 - (L_{A_nC_n} - L_{B_nG_n})^2 - L_{G_nH_n}^2}{2L_{G_nH_n}\sqrt{L_{A_nB_n}^2 + (L_{A_nC_n} - L_{B_nG_n})^2}} - \tan^{-1}\frac{L_{A_nC_n} - L_{B_nG_n}}{L_{A_nB_n}} \tag{2}$$

$$w_n = L_{OA_n} + L_{C_nD_n}\cos\alpha_n \tag{3}$$

$$h_n = -L_{A_nC_n} - L_{C_nD_n}\sin\alpha_n - L_{D_nI_n} \tag{4}$$

Here, n = 1, 2 denotes the index of the movable landing leg. The support arm is designed with a right-handed thread. In this configuration, the elongation of the support arm is determined by the rotation angle of the motor $r(t)$ (in radians) and the pitch p, satisfying Eq. (5). When viewed from above the drone, the rotation is defined as positive when the motor turns counterclockwise.

$$L_{G_nH_n}(t) = L_{G_nH_n}(0) + \frac{pr(t)}{2\pi} \tag{5}$$

When the mechanical structure changes over time, the velocity kinematics equation for a landing leg system is given by Eqs. (6) and (7). The single-motor actuation scheme inherently induces a coupling relationship between the terminal horizontal dw_n and vertical velocities dh_n.

$$J_n = -\begin{bmatrix} L_{C_nD_n}\sin\alpha_n & L_{C_nD_n}\cos\alpha_n \end{bmatrix}^{\mathrm{T}} \tag{6}$$

$$v_n(t) = [\mathrm{d}w_n, \mathrm{d}h_n]^{\mathrm{T}} = J_n\frac{p\dot{r}}{2\pi}\frac{\mathrm{d}\alpha_n}{\mathrm{d}L_{G_nH_n}} \tag{7}$$

Dynamic modeling

To facilitate the derivation of the dynamic model while preserving the fundamental characteristics of the system, the following standard assumptions are adopted:

(1) Rigid body assumption: All components are rigid; elastic deformations are neglected.

(2) Lumped parameter model: The system's properties (mass, damping, stiffness) are concentrated at discrete points.

(3) Ideal joints: The joints are frictionless, without backlash or hysteresis.

(4) Idealized actuators: The actuators are perfect sources of torque or force sources; the internal dynamics are neglected.

(5) Uniform gravity: A constant gravitational field is assumed for consistent potential energy reference.

These assumptions yield a deterministic, time-invariant, nonlinear model, forming the basis for the controller's design and stability analysis. Though it is simplified, it captures the dominant dynamics that are essential for evaluating this work's core contributions.

The landing leg's dynamic model is derived using Lagrangian mechanics, a framework that is ideal for constrained systems with multiple degrees of freedom. This energy-based approach enables a systematic derivation of the equations of motion, simplifying the analysis of complex systems like the landing leg by avoiding direct force calculations.

The length of the arm G_nH_n, as the variable s_n, is selected as the independent coordinate to compute the kinetic and potential energy of the entire landing leg. The dynamics of a landing leg are primarily constituted by the following four active linkages: the rocker arm C_nD_n, the standard arm E_nF_n, the support arm G_nH_n, and the contact rod D_nI_n. The kinetic energy K_n, the potential energy P_n, the generalized forces, and the Lagrange equation of a single landing leg are described in Eqs. (8)–(11), where m with an arm's name as the subscript represents the mass of the arm.

$$\begin{aligned} K_n(s_n, \dot{s}_n) = &\left[\left(\frac{m_{C_nD_n} + m_{E_nF_n}}{6} + \frac{m_{D_nI_n}}{2}\right)L_{C_nD_n}^2 + \frac{m_{G_nH_n}}{2}L_{C_nH_n}^2\right]\dot{\alpha}_n^2 \\ &+ \frac{m_{G_nH_n}}{6}L_{G_nH_n,\max}^2\dot{\beta}_n^2 - \frac{\mathrm{m}_{G_nH_n}}{2}L_{C_nH_n}L_{G_nH_n,\max}\dot{\alpha}_n\dot{\beta}_n\cos(\alpha_n - \beta_n) \end{aligned} \tag{8}$$

$$\begin{aligned} P_n(s_n) = &3P_{C_n} + P_{E_n} - m_{G_nH_n}\left(\vec{a}_n\cdot\overrightarrow{C_nH_n}\right) + \frac{\mathrm{m}_{G_nH_n}L_{G_nH_n,\max}}{2s_n}\left(\vec{a}_n\cdot\overrightarrow{G_nH_n}\right) \\ &-(m_{C_nD_n} + m_{D_nI_n})\left(\vec{a}_n\cdot\overrightarrow{C_nD_n}\right) - \frac{m_{D_iI_i}}{2}\left(\vec{a}_n\cdot\overrightarrow{D_nI_n}\right) \end{aligned} \tag{9}$$

$$L_n = K_n - P_n \tag{10}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_n}{\partial\dot{s}_n}\right) - \frac{\partial L_n}{\partial s_n} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial K_n}{\partial\dot{s}_n}\right) - \frac{\partial K_n}{\partial s_n} + \frac{\partial P_n}{\partial s_n} = \frac{\partial r}{\partial s_n}(\tau_{Mn} + \tau_{Dn}) \tag{11}$$

The dynamic parameter with s_n, $\dot{s}_n$ as the variables is denoted M_n, C_n, G_n. The Lagrangian equation for the n-th landing leg simplifies to:

$$M_n(s_n, \dot{s}_n)\ddot{s}_n + C_n(\vec{a}_i, s_n, \dot{s}_n)\dot{s}_n + G_n(\vec{a}_n, s_n) = p^{-1}(\tau_{Mn} + \tau_{Dn}) \tag{12}$$

By substituting the independent variable s_n with h_n and updating the dynamic parameters to M'_n, C'_n, G'_n, the dynamic equation of height (Eq. [13]) for a single landing leg is derived. Subsequently, by synthesizing the dynamics of both landing legs, the multibody dynamic equation (Eq. [14]) for the landing system is obtained.

$$M'_n(h_n, \dot{h}_n)\ddot{h}_n + C'_n(\vec{a}_n, h_n, \dot{h}_n)\dot{h}_n + G'_n(\vec{a}_n, h_n) = p^{-1}(\tau_{Mn} + \tau_{Dn}) \tag{13}$$

$$M_L\ddot{\mathbf{H}} + C_L\dot{\mathbf{H}} + G_L = p^{-1}(\tau_M + \tau_D) \tag{14}$$

where, $M_L \in \mathbb{R}^{n\times n}$ is the mass matrix, $C_L \in \mathbb{R}^{n\times n}$ contains the Coriolis and centrifugal terms, and $G_L \in \mathbb{R}^n$ is the gravity vector, which

incorporates the inertia and gravitational effects of the entire landing system. Under low-speed operation of the landing legs, the influence of higher-order terms (unmodeled dynamics) of State $\mathbf{H}$ on the system is negligible. The conservative and nonconservative forces within the system originate from the motor's actuation, ground contact, and joint–motor friction.

According to the static analysis, the support arm bears the primary compressive force. Because of the unique characteristics of the lead screw transmission, the angle of the compressive force acting on the screw thread exceeds the friction angle of the contact surface, resulting in a self-locking phenomenon. This indicates that the energy cannot propagate into the system, and thus its impact is disregarded until full landing is achieved. The combined friction is aggregated into a single term and operates in the same channel as the motor control input.

A model for the dynamic perception of landing

To enhance the model's accuracy, the landing system's potential energy is dynamically adjusted on the basis of the aircraft's six-axis attitude (Figs. 3, 4). The method involves the following.

Acquisition of the gravity vector: The equivalent gravity in the inertial frame is obtained from accelerometers or an attitude and heading reference system (AHRS) by extracting the low-frequency gravitational component.

Transformation and projection of the coordinates: Gravity is transformed to the airframe and projected onto two orthogonal planes (Ox_1z, Ox_3z) to get the components $\vec{a}_1$ and $\vec{a}_2$. A leg extension angle is acquired via an encoder.

Calculation of potential energy: Taking the UAV's center O as the zero point, the leg's potential energy is computed by summing the inner products of the gravity components and linkage vectors in their respective planes.

According to Eq. (13), the system's matrices M_L, C_L, and G_L and are nonlinear functions of the leg's height or velocity. Real-time parameter updates make the dynamic model converge toward the actual physics. This refined model drives the ASMC controller and initializes the full-order state observer (FOSO). The observer runs in parallel, providing states for verifying the ASMC's stability.

This model reveals that the two legs, though driven independently, exhibit low coupling because of the numerical correlations between the gravity components. Thus, they can be treated as decoupled, independent systems. Distributed parallel computing via Microprocessor Unit (MPU) communication halves the computation time while maintaining precision.

Sensor multimodal perception and data fusion

The DTFSF algorithm

The DTFSF algorithm uses data from at least four ranging sensors and one attitude sensor to perform terrain detection, classification,

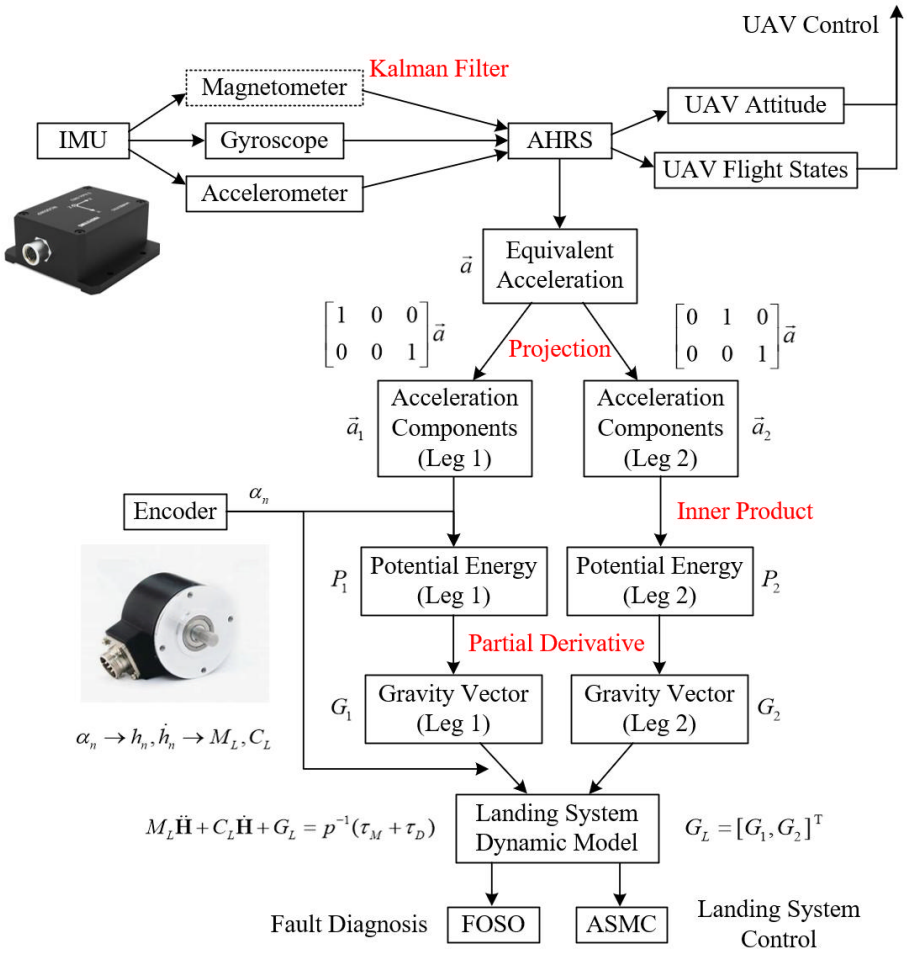

Fig. 3 Dynamic model solution for the landing system.

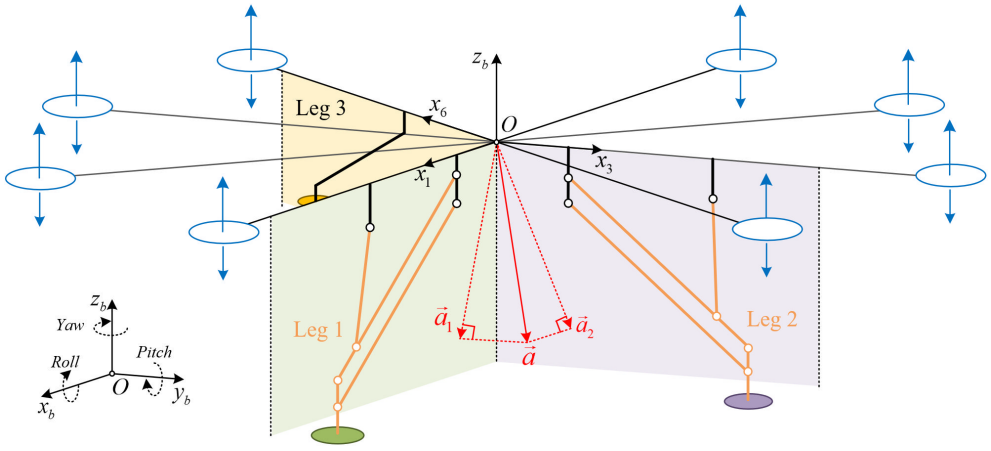

Fig. 4 Dynamic perception strategy for the landing model.

and error correction during low-altitude UAV flight. It incorporates a sliding window mechanism to ensure the data's timeliness and efficient Microcontroller Unit (MCU) memory use, enabling intelligent landing on compact embedded devices (the process is shown in Fig. 5).

Although the current ranging sensors operate at 300–1,000 Hz, traditional point cloud processing methods are too computationally expensive for small embedded systems. DTFSF overcomes this by replacing complex classifiers with optimized geometric primitives, achieving real-time terrain assessment in just 1 ms on microcontrollers like the STM32H7 (50× faster than PCA). It maintains high recognition accuracy for typical terrains with minimal data. Crucially, for landing, UAVs only need to detect local flatness within the landing zone, not map the entire area, enabling a rapid touchdown.

Step 1. The system's setup

Three ranging sensors are mounted on the three legs' endpoints and on one LiDAR system at the airframe center. Let the positions of sensors in the airframe be $P_1, P_2, P_3 \in \mathbb{R}^3$ (mounted on the three legs) and $P_0 \in \mathbb{R}^3$ (the center of the airframe). P_i is determined from the installation position of the landing leg and the kinematic equation, that is:

$$P_i = \begin{cases} [0,0,-h_0]^{\mathrm{T}}, & i=0 \\ [w_1(\alpha_1),0,-h_1(\alpha_1)]^{\mathrm{T}}, & i=1 \\ [0,w_2(\alpha_2),-h_2(\alpha_2)]^{\mathrm{T}}, & i=2 \\ \left[-\frac{w_3}{\sqrt{2}},-\frac{w_3}{\sqrt{2}},-h_3\right]^{\mathrm{T}}, & i=3 \end{cases} \tag{15}$$

To ensure full coverage of all landing legs by the center LiDAR's point cloud prior to touchdown, the vertical field of view (FOV) θ_v must satisfy the following:

$$\theta_v \geq 2\arctan\frac{w(s_{\min})}{h(s_{\min})-h_0} \tag{16}$$

where, $s_{\min}$ denotes the minimum permissible length of the support strut.

Step 2. IMU-based ground point acquisition

At each movement step $t < k$, the UAV moves by Δs_t along a direction v_t. The ground contact points are directly computed in the airframe as follows:

$$Q_i^{(t+1)} = R_t(P_i - d_i^{(t+1)}\hat{z}_b) + \sum_{N=1}^{t}\Delta s_N v_N, \quad i=\{0,1,2,3\},\ t<k \tag{17}$$

where, $\hat{z}_b = (0,0,1)^{\mathrm{T}}$ is the vertical unit vector of the airframe, $d_i^{(t)}$ represents the filtered ranging result of the *i*-th infrared (IR) sensor of

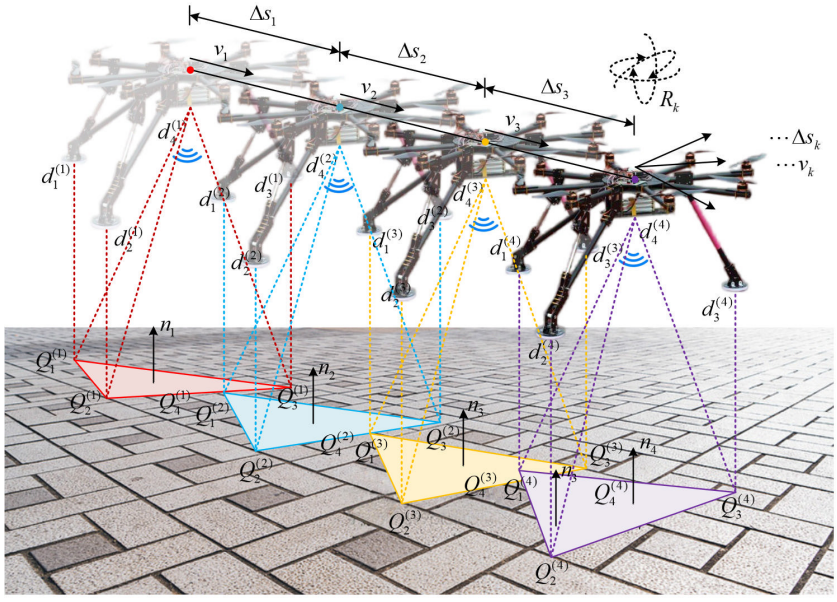

Fig. 5 Schematic diagram of the DTFSF method.

step t, $R_t \in \mathbb{R}^{3\times3}$ is the roll–pitch–yaw (RPY) rotation matrix of the rotor UAV's airframe at the current position relative to the initial point ($t = 0$).

Note: Certain LiDAR models directly output relative coordinates, enabling one to bypass the coordinate transformation steps.

Step 3. Sliding window initialization

Define a sliding window covering the latest m movement steps as follows:

$$W_k = \{t | t = k-m+1, k-m+2, \cdots, k\} \tag{18}$$

Step 4. LiDAR–ranging sensor fusion

Fault detection leverages hardware redundancy and a statistical consistency check (a two-sample t-test) between the measurements of LiDAR and the ranging sensor over the sliding window. If significant inconsistency is detected for a single leg, that ranging sensor's confidence is downweighted. If inconsistencies occur on two or more legs, the LiDAR's confidence is downweighted instead.

Step 5. Local and global normal vector

For each $t \in W_k$, compute the plane normal vector using three noncollinear points as follows:

$$n_t = \frac{(Q_2^{(t)} - Q_1^{(t)}) \times (Q_3^{(t)} - Q_1^{(t)})}{\left\| (Q_2^{(t)} - Q_1^{(t)}) \times (Q_3^{(t)} - Q_1^{(t)}) \right\|} \tag{19}$$

Construct the normal vector matrix $\mathbf{N}_W$ for the window as follows:

$$\mathbf{N}_W = \begin{bmatrix} n_{k-m+1}^{\mathrm{T}} \\ \vdots \\ n_k^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{m\times3} \tag{20}$$

Compute the following cosine similarity matrix (CSM) $\mathbf{S}_W \in \mathbb{R}^{m\times m}$:

$$\mathbf{S}_W = \mathbf{N}_W \mathbf{N}_W^{\mathrm{T}} \tag{21}$$

Step 6. Dynamic flatness decision

Define the thresholds λ (angular tolerance) and κ_1, κ_2 ($0 < \kappa_2 < \kappa_1 < 1$) (consistency ratio). Here, $\mathbf{S}_{u,v}$ is the element in the u-th row and v-th column of the matrix $\mathbf{S}_W$, where $1 \le u, v \le m$. The landing mode will be changed to "strict" if

$$\mathrm{CR}_\lambda = \frac{2}{m(m-1)} \sum_{u=1}^{m-1} \sum_{v=u+1}^{m} \mathbb{I}_{\{\mathbf{S}_{u,v} \ge \lambda\}} < \kappa_1 \tag{22}$$

where, $\mathbb{I}_{\{*\}}$ is a logical operator that outputs 1 when the statement * in the brace is true, and 0 otherwise. The intelligent landing control system (LCS) will reject the landing command if

$$\mathrm{CR}_\lambda < \kappa_2 \tag{23}$$

This indicates that CR_λ serves as a critical indicator reflecting the terrain's flatness.

Figure 6 presents a flowchart summarizing Steps 1 to 6. To ensure that each displacement vector generates useful ground information, the direction v_t and distance Δs_t are determined by the rotorcraft's path planning algorithm (boustrophedon, genetic algorithm [GA] or particle swarm optimization [PSO]), such that all sampling points are uniformly distributed within the prelanding area. The size of this area is typically determined by the horizontal dimensions of the UAV, and it is generally set to 3–5 times the UAV's size.

Multiterrain simulation

To validate the effectiveness of the algorithm, we utilized real terrain data to compute the $\mathrm{CR}_{0.95}$. The terrains included basic terrains (flat and slope), obstacle terrains (rocks, trenches, and staircases), and extreme terrains (urban debris or planetary analogs) to ensure diversity. For the aforementioned terrains, a dedicated terrain model was implemented in for simulations with the DTFSF algorithm. Paths were generated with complete randomness at step sizes below the maximum leg span (0.5–0.8 m in this study) to ensure full coverage of all triangular facets. The sliding window width was empirically set to 20. Gaussian functions ($\mu = 0$, $\sigma = 0.02$ m = measurement accuracy) modeled potential ranging sensor errors. Each terrain underwent 10 randomized path simulations, with representative paths visualized in Fig. 7. The black lines depict the trajectories of the UAV's center, the red points indicate ground projections of the ranging sensors (approximating the measurement locations), and the blue-shaded regions represent convex hulls encompassing all measurement points.

The visualization of the landing terrain's CSM is shown in Fig. 8. The performance metrics are quantified in Table 1.

As one can see in Table 1, the experimental results demonstrate a statistically significant correlation between the terrain's complexity and the $\mathrm{CR}_{0.95}$ metric, validating the efficacy of the DTFSF algorithm in dynamic terrain assessment. Flat terrain exhibited near-optimal $\mathrm{CR}_{0.95}$ values (mean: 98.7%, range: 95%–100%), indicating minimal terrain-induced risks and suitability for autonomous landing. In contrast, sloped terrain showed moderate degradation (mean: 90.2%, range: 82%–96%), necessitating adaptive control strategies to mitigate slope-induced uncertainties. Obstacle terrains, particularly continuous obstacles (mean: 58.8%, range: 51%–72%) and extreme terrains, including urban debris (mean: 8.3%, range: 4%–17%) and planetary analog terrains (mean: 11.8%, range: 9%–14%), exhibited near-zero $\mathrm{CR}_{0.95}$ values, aligning with their inherent structural irregularities and reinforcing the necessity of preemptive landing cancellation. Our simulation results across diverse terrains (Table 1) demonstrate that $\mathrm{CR}_{0.95}$ effectively distinguishes optimally flat terrain, slopes, and obstructed terrains, providing a robust safety margin while maintaining practical applicability. The hierarchical $\mathrm{CR}_{0.95}$ thresholds (> 80%: safe; 50%–80%: strict; < 50%: reject; i.e. $\kappa_1 = 0.8$, $\kappa_2 = 0.5$), effectively operationalize terrain-driven decision-making, underscoring the algorithm's robustness in diverse real-world scenarios.

The selection of specific λ thresholds for high-confidence safe landing was determined through an extensive empirical analysis of the trade-off between the ability to discriminate the terrain and operational safety. A higher threshold would excessively restrict landing opportunities on undulating yet safe terrains, leading to unnecessarily aborted landings and reducin the system's utility. Conversely, a lower threshold would compromise the acquisition of terrain information within the sliding window and increase the risk of accepting areas with potential localized hazards, thereby affecting the landing's stability.

Analysis and comparison of the DTFSF algorithm

In the DTFSF algorithm, both the step size and sliding window width critically influence the terrain assessment metrics. Intuitively, smaller step sizes and wider windows increase the measurement density within local regions, theoretically improving the accuracy of the analyses. We sampled 120 datasets (20 per terrain type across six terrains). For window sizes {1, 5, 10, 20, 30, 50}, each dataset's $\mathrm{CR}_{0.95}$ output was classified into three terrain types: Extreme (0%–50%), obstacle (50%–80%), and basic (80%–100%).

Four sampling strategies were implemented:

(1) Variable step: inversely proportional to the gradient;

(2) Fixed step: Below the sensors' spacing, approximately equal to the sensors' spacing, or exceeding the sensors' spacing.

The results of classification accuracy are shown in Fig. 9.

As the sliding windows' width m increases and the step size Δs decreases, it effectively captures localized and short-term terrain information around the intended landing site, demonstrating enhanced data reliability.

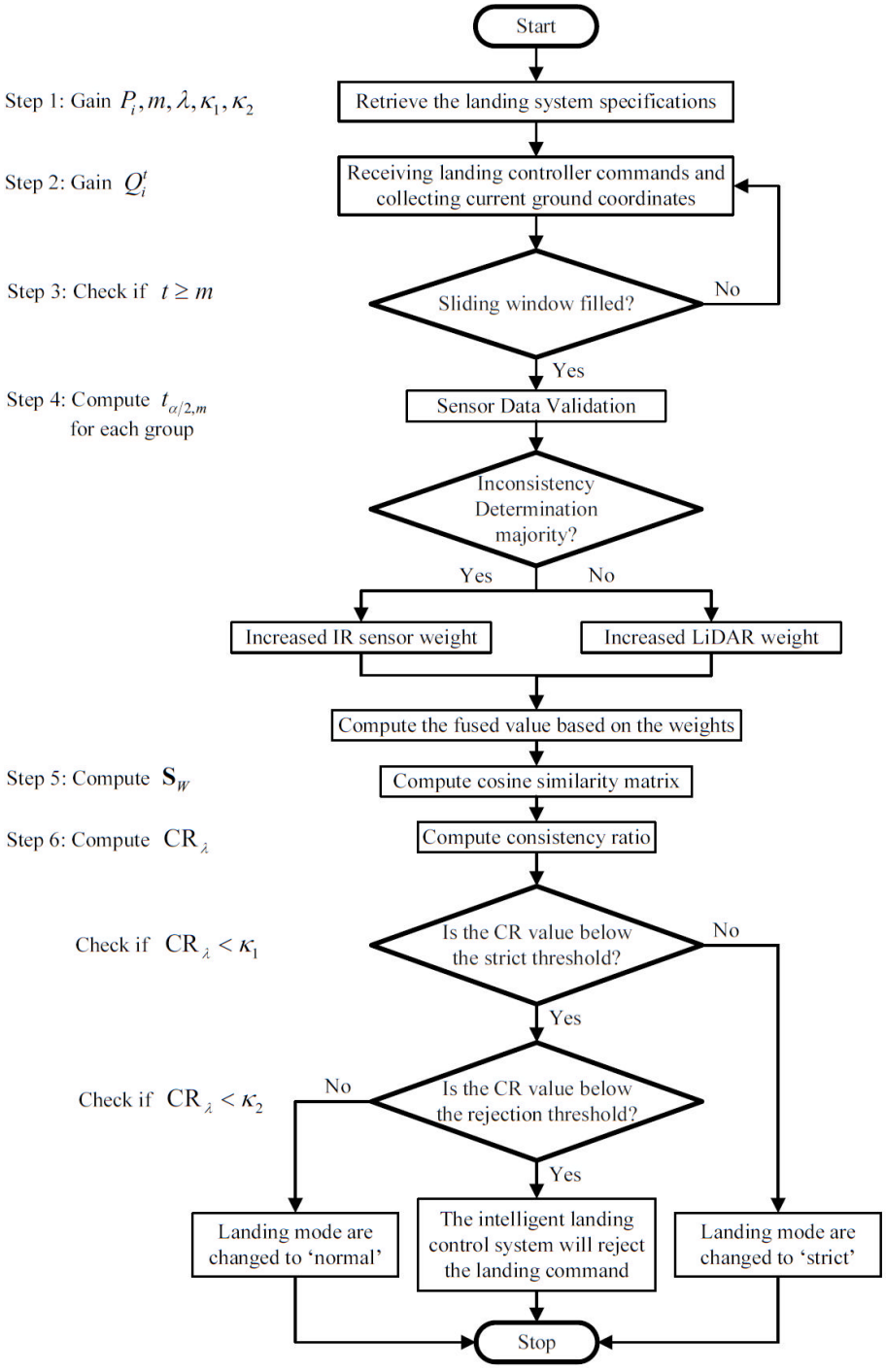

Fig. 6 Flowchart of the DTFSF method.

The results of the simulation indicate that terrain classification achieves optimal efficiency (accuracy > 96%) with a step size of 0.125 m and a sliding window width of 20. Increasing the width to 50 incurs approximately sixfold computational overhead while improving the accuracy by less than 2%. Consequently, the experimental configuration uses a step size of 0.1 m and a window width of 20 to balance real-time performance with classification accuracy.

Compared with the full-batch analysis that requires one to acquire and analyze the complete set of $4k$ data points (full batch), the DTFSF method exhibits the advantages summarized in Table 2. For resource-constrained UAVs navigating quasistatic yet evolving environments, this method strikes an optimal balance between adaptability and efficiency, a feat that cannot be achieved with traditional full-batch techniques.

The ASMC design

Description

Based on the dynamic response estimation and structural characteristics of the controlled plant, the sliding surface equation $\mathbf{\Omega}$ and control law u are proposed:

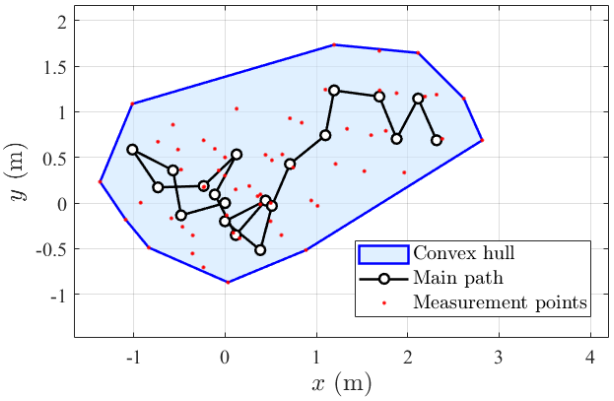

Fig. 7 Illustration of a randomized path ($m = 20$).

$$\mathbf{\Omega} = \dot{e} + \mathbf{\Lambda} e = (\dot{\mathbf{H}} - \dot{\mathbf{H}}_d) + \mathbf{\Lambda}(\mathbf{H} - \mathbf{H}_d) \tag{24}$$

$$u = u_{\text{eq}} + u_{\text{sw}} = \underbrace{M_L\left(\ddot{\mathbf{H}}_d - \mathbf{\Lambda}\dot{e}\right) + C_L\left(\dot{\mathbf{H}} - \mathbf{\Omega}\right) + G_L - \frac{\hat{\tau}_D}{p}}_{u_{\text{eq}}} \underbrace{- \mathbf{K}\,\text{sat}\left(\frac{\mathbf{\Omega}}{\delta}\right) - \mathbf{\Phi}\mathbf{\Omega}}_{u_{\text{sw}}} = \frac{\tau_M}{p} \tag{25}$$

where, $\mathbf{\Lambda}$ is the coefficient of the linear sliding surface, and $\mathbf{K}$ and $\mathbf{\Phi}$ are the gain matrix of the control law. The terms u_{eq}, u_{sw} denote the equivalent control input and switching control input, respectively.

According to Eq. (25), the motor control torque is obtained as follows:

$$\tau_M = p\left[-M_L\mathbf{\Lambda}\dot{\mathbf{H}} + C_L\left(\dot{\mathbf{H}} - \mathbf{\Omega}\right) + G_L - \mathbf{K}\,\text{sat}\left(\frac{\mathbf{\Omega}}{\delta}\right) - \mathbf{\Phi}\mathbf{\Omega}\right] - \hat{\tau}_D \tag{26}$$

where, δ is the boundary layer's thickness. Let $S \subseteq \{1, 2, \ldots, n\}$ denote the set of indices where $|\sigma_i| > \delta$, and the saturation function is defined as follows:

$$\text{sat}\left(\frac{\sigma_i}{\delta}\right) = \begin{cases} \text{sgn}(\sigma_i) & i \in S \\ \dfrac{\sigma_i}{\delta} & i \notin S \end{cases} \tag{27}$$

The adaptive disturbance estimation $\hat{\tau}_D$ complies with the following update law:

$$\dot{\hat{\tau}}_D = \mathbf{\Gamma}\mathbf{\Omega} \tag{28}$$

Here, $\mathbf{\Gamma}$ is the gain matrix of the adaptive law, and $\hat{\tau}_D$ is used to estimate the real constant disturbance τ_D, as the disturbance magnitude can be considered to be bounded and constant during the short landing period, i.e.

$$\|\tau_D\|_\infty \le D \tag{29}$$

where, D is the disturbance bound.

Main results

Construct a quadratic Lyapunov function incorporating the sliding surface σ and the adaptive parameter estimation error $\tilde{\tau}_D = \hat{\tau}_D - \tau_D$ as follows:

$$V = \frac{1}{2}\mathbf{\Omega}^{\text{T}} M \mathbf{\Omega} + \frac{1}{2p}\tilde{\tau}_D{}^{\text{T}}\mathbf{\Gamma}^{-1}\tilde{\tau}_D \tag{30}$$

The derivative of the Lyapunov function is obtained as follows:

$$\dot{V} = \mathbf{\Omega}^{\text{T}} M \dot{\mathbf{\Omega}} + \frac{1}{2}\mathbf{\Omega}^{\text{T}}\dot{M}\mathbf{\Omega} - \frac{\mathbf{\Omega}^{\text{T}}\tilde{\tau}_D}{p} \tag{31}$$

From Eqs (14), (24), and (26), it follows that

$$\dot{\mathbf{\Omega}} = M_L^{-1}\left[-\mathbf{K}\,\text{sat}\left(\frac{\mathbf{\Omega}}{\delta}\right) - \mathbf{\Phi}\mathbf{\Omega} - C_L\mathbf{\Omega} + \frac{\tilde{\tau}_D}{p}\right] \tag{32}$$

The skew-symmetric property $x^{\text{T}}(\dot{M}_L - 2C_L)x = 0,\ \forall x \in \mathbb{R}^n$, is a fundamental characteristic of Lagrangian mechanical systems. This property arises from the energy conservation structure of the Euler–Lagrange equations, where $C_L(\dot{\mathbf{H}}, \ddot{\mathbf{H}})$ is defined such that $\dot{M}_L - 2C_L$ is skew-symmetric. Physically, it implies that the internal forces do not network on the system, thereby preserving the

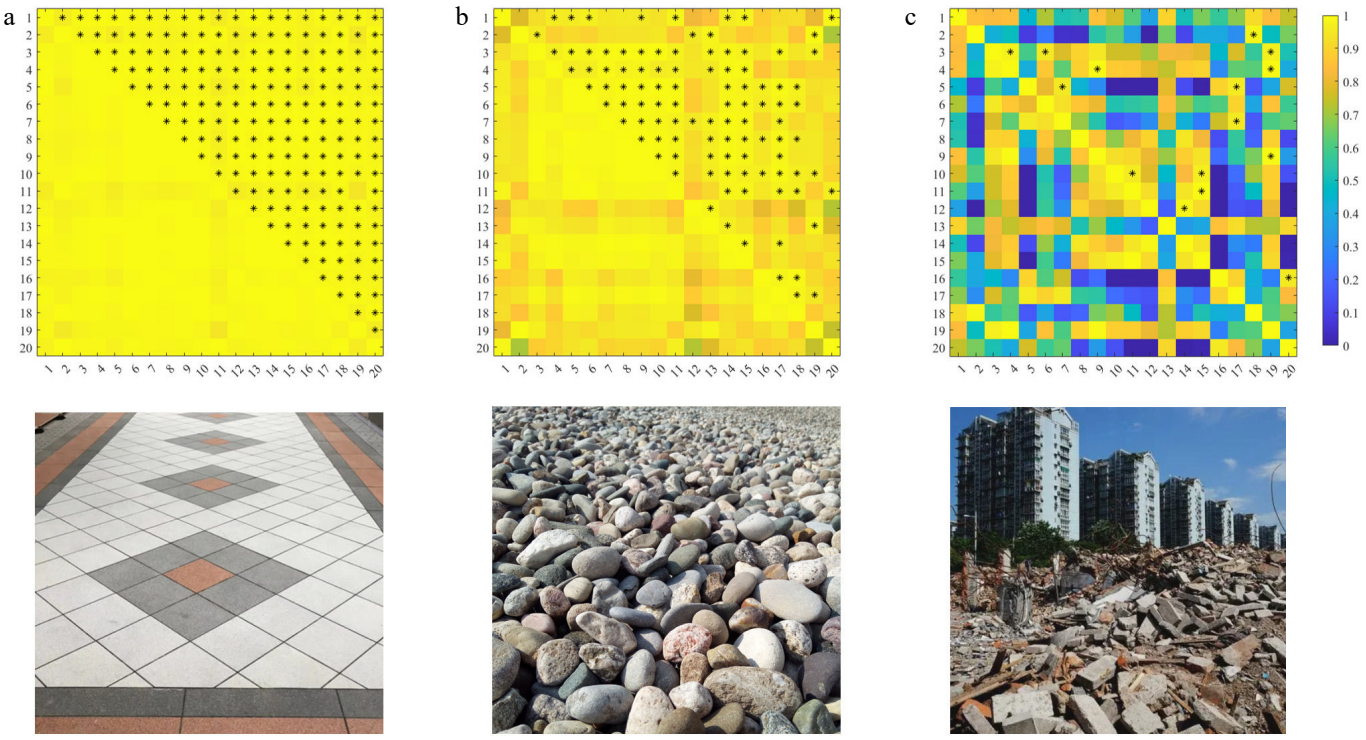

Fig. 8 Landing terrain types and the corresponding CSM: (a) Basic, (b) obstacle, and (c) extreme.

Table 1. Quantitative evaluation of the terrains' CR.

Terrains		Landing feasibility	10 Random $CR_{0.95}$ (%)
Basic	Flat	○	100.00, 97.89, 96.84, 94.74, 100.00, 98.95, 100.00, 100.00, 100.00, 99.47
	Slope	○	89.47, 91.05, 96.32, 82.63, 93.16, 82.11, 95.26, 94.74, 82.63, 94.74
Obstacle	Discrete	○	66.84, 77.89, 55.79, 76.84, 73.68, 64.74, 60.00, 57.89, 60.53, 77.89
	Continuous	○	61.58, 57.37, 71.58, 55.79, 50.53, 68.95, 55.26, 50.53, 52.11, 64.74
Extreme	Urban debris	×	11.05, 7.37, 4.74, 5.79, 4.74, 16.84, 5.79, 10.53, 7.37, 8.42
	Planetary analog	×	12.11, 12.68, 10.53, 12.63, 10.00, 12.11, 12.11, 9.47, 12.63, 14.21

system's passivity. Mathematically, this property is derived by differentiating $M_L(\dot{\mathbf{H}},\ddot{\mathbf{H}})$ and constructing $C_L(\dot{\mathbf{H}},\ddot{\mathbf{H}})$ via the Christoffel symbols of the first kind[34].

By substituting Eq. (32) into Eq. (31) and simplifying, we obtain

$$\begin{aligned}\dot{V} &= \mathbf{\Omega}^{\mathrm{T}}[-\mathbf{K}\mathrm{sat}(\frac{\mathbf{\Omega}}{\delta}) - \mathbf{\Phi}\mathbf{\Omega} + \frac{\tilde{\tau}_D}{p}] + \frac{1}{2}\mathbf{\Omega}^{\mathrm{T}}(\dot{M}_L - 2C_L)\mathbf{\Omega} - \frac{\mathbf{\Omega}^{\mathrm{T}}\tilde{\tau}_D}{p} \\ &= -\mathbf{\Omega}^{\mathrm{T}}\mathbf{K}\mathrm{sat}(\frac{\mathbf{\Omega}}{\delta}) - \mathbf{\Omega}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Omega} \end{aligned} \tag{33}$$

Theorem 1.

Consider the following Lyapunov function derivative:

$$\dot{V} = -x^{\mathrm{T}}A\mathrm{sat}(\frac{x}{\varepsilon}) - x^{\mathrm{T}}Bx$$

where, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, and $\varepsilon > 0$. The derivative $\dot{V}$ is seminegative definite if the matrix A is a diagonal semipositive definite matrix and B is a semipositive definite matrix.

Proof

Let $\xi = \mathrm{sat}(x/\varepsilon)$. The saturation function preserves the sign of ξ_i, i.e., $\xi_i x_i \geq 0$ and $0 \leq |\xi_i| \leq 1$. Define two regions as follows:

Linear region: $|x_i| \leq \varepsilon \Rightarrow \xi_i = x_i/\varepsilon$;

Saturated region: $|x_i| > \varepsilon \Rightarrow \xi_i = \mathrm{sgn}(x_i)$.

Case 1: All components in the linear region

If $|x|_\infty \leq \varepsilon$, then $\xi = x/\varepsilon$. Substituting into $\dot{V}$, we have

$$\dot{V} = -x^{\mathrm{T}}(\frac{A}{\varepsilon} + B)x \tag{34}$$

For $\dot{V} \leq 0$, $x^{\mathrm{T}}(A/\varepsilon + B)x \geq 0$ must hold for all $x \neq 0$. This requires $A/\varepsilon + B$ to be semipositive definite. The satisfaction of the condition is a direct consequence of the known information.

Case 2: Partial or full saturation

Assume that some components of x are saturated. In this case

$$x^{\mathrm{T}}A\mathrm{sat}(\frac{x}{\varepsilon}) = \sum_{i\in S} x_i(\frac{x_i}{\varepsilon})A_{i,i} + \sum_{i\notin S} x_i\mathrm{sgn}(x_i)A_{i,i} \tag{35}$$

Since $x_i\mathrm{sgn}(x_i) = |x_i| \geq 0$, the diagonal term satisfies $x_iA_{ii}\xi_i \geq 0$ if $A_{ii} \geq 0$.

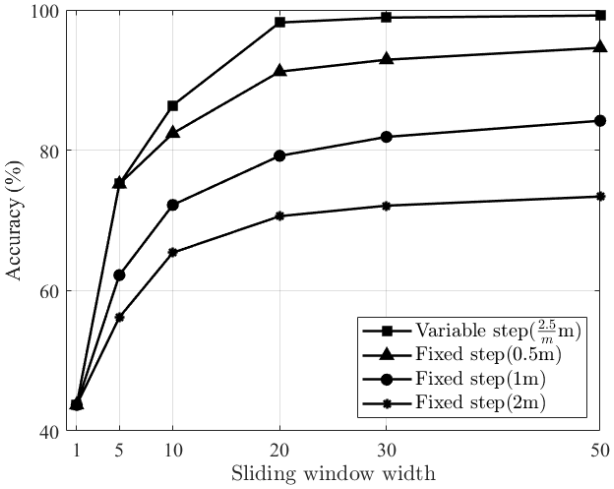

Fig. 9 Recognition accuracy at different step sizes.

$$\begin{aligned}\dot{V} &= -\frac{1}{\varepsilon}\sum_{i\in S} x_i^2A_{i,i} - \sum_{i\notin S}|x_i|A_{i,i} - \sum_{i\neq j} x_iA_{i,j}\xi_j - x^{\mathrm{T}}Bx \\ &\leq -\frac{1}{\varepsilon}\sum_{i\in S} x_i^2A_{i,i} - \sum_{i\notin S}|x_i|A_{i,i} \\ &\leq 0 \end{aligned} \tag{36}$$

Combining both cases above, the proof is completed.

This indicates that the system's sliding mode kinetic energy decays to zero, causing tracking errors to vanish near-exponentially. Similar to adaptive PID-C, this ASMC strategy offers robustness against disturbances. Although its boundary layer reduces chattering, it slightly increases the tracking error. Timely controller switching or parameter adjustment can boost the response speed and minimize residuals.

The developed control law provides adaptive compensation against unknown disturbances, achieving full robustness against in-channel disturbances and partial robustness to unmodeled dynamics. Unlike existing SMCs, our design incorporates multiple dedicated tuning parameters for the tracking velocity $\mathbf{\Lambda}$, transient dynamics $\mathbf{K}$, steady-state error δ, system stability $\mathbf{\Phi}$, and robustness $\mathbf{\Gamma}$. Crucially, it operates without predefined trajectories, enabling seamless real-time path integration with fast convergence. As a model-specific robust controller, it can be combined with observers for enhanced stability. Future work may explore making its fixed parameters adaptive.

A drawback is potential control signal saturation from large initial errors, requiring clamping of the motor voltage. For the decoupled 2-DoF landing system, the controller is computationally lightweight, sustaining a 1,000-Hz frequency on embedded systems for high-precision real-time landings.

Table 2. Comparative analysis: Full batch vs. DTFSF.

Metric	Full batch	DTFSF
Computational complexity	$\mathrm{O}(k^2)$	$\mathrm{O}(m^2)$
Real-time performance	Latency: $T_{\mathrm{delay}} = k\Delta t$	Bounded latency: $T_{\mathrm{delay}} = m\Delta t$
Memory footprint	Requires storage of an $\mathrm{O}(k^2)$ similarity matrix and $\mathrm{O}(k)$ data points.	Fixed storage: $\mathrm{O}(m^2)$ matrix and $\mathrm{O}(m)$ data points.
Robustness to noise	Susceptible to historical noise caused by global averaging.	Enhanced robustness via dynamic window updates and localized outlier rejection.
Hardware compatibility	Requires high-performance processors (e.g., Graphics Processing Unit [GPU]) for large k	Executable on microembedded systems (e.g., ARM Cortex-M4) with constrained resources.

Numerical simulation

Although existing multirotor landing systems for 1–12 articles predominantly rely on hierarchical PID-C architectures, this work implements an ASMC scheme grounded in Lagrangian mechanics for the underactuated landing gear dynamics, with comparative simulations designed to rigorously quantify its nonlinear compensation capabilities against conventional linear control paradigms. This system is simulated with four scenarios:

(1) Case 1 (without disturbance): $\tau_D = 0$;

(2) Case 2 (static disturbance): $\tau_D = \begin{cases} 0 & t < 5\ \mathrm{s} \\ 0.002 & t \geq 5\ \mathrm{s} \end{cases}$;

(3) Case 3 (multiplicative fault): $\tau_M = 0.8u$;

(4) Case 4 (dynamic disturbance): τ_D is Gaussian white noise with 0 mean and 0.001 variance.

In order to apply the proposed method, the parameters of the ASMC and PID-C are selected as follows:

$$\mathbf{\Omega} = \dot{e} + Ie, \mathbf{K} = \mathrm{diag}(1,1), \mathbf{\Phi} = \mathrm{diag}(0.1,0.1), \mathbf{\Gamma} = \mathrm{diag}(2,2), \delta = 0.01$$

$$K_P = 0.5, K_I = 0.2, K_D = 0.5$$

THe initial height, initial velocity, and target height are

$$H(0) = 0.273\ \mathrm{m},\ \ \dot{H}(0) = 0\ \mathrm{m/s},\ \ H_d = -0.4\ \mathrm{m}\ .$$

The initial disturbance estimation is

$$\hat{\tau}_D(0) = 0\ \mathrm{N \cdot m}$$

The PID-C parameters are tuned according to the ASMC framework. When the matrix $\mathbf{\Lambda}$ is the identity matrix, K_P/K_D are set to 1. The system's stability is achieved by adjusting K_P to 0.5. To eliminate the steady-state error, K_I is set to 0.2, which achieves a good balance between response speed and stability.

Four scenarios with their corresponding results shown in Figs. 10–12 and Table 3.

Under ideal disturbance-free conditions, the curves in Fig. 10a demonstrate exponential convergence behavior in trajectory tracking for all three controllers. The ASMC achieves the fastest convergence rate, whereas PID-C exhibits slower convergence to ensure stability. Figure 10b confirms that both ASMC and SMC ensure convergence to the sliding manifold under ideal conditions, driving the system's states along the manifold with mitigated chattering.

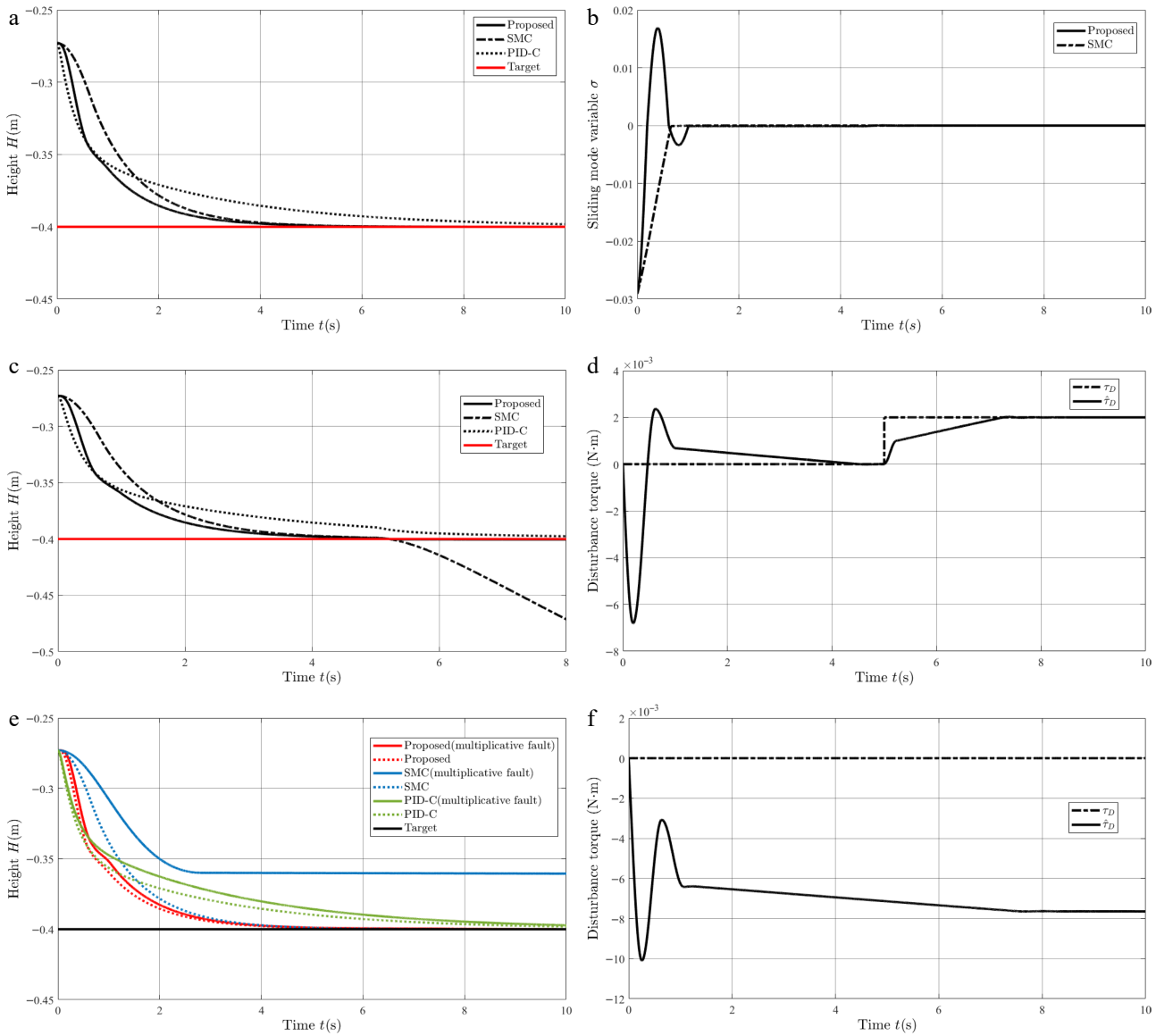

Fig. 10 Simulation results: (a) H for Case 1; (b) σ for Case 1; (c) H for Case 2; (d) $\hat{\tau}_D$ for Case 2; (e) H for Case 3; (f) $\hat{\tau}_D$ for Case 3.

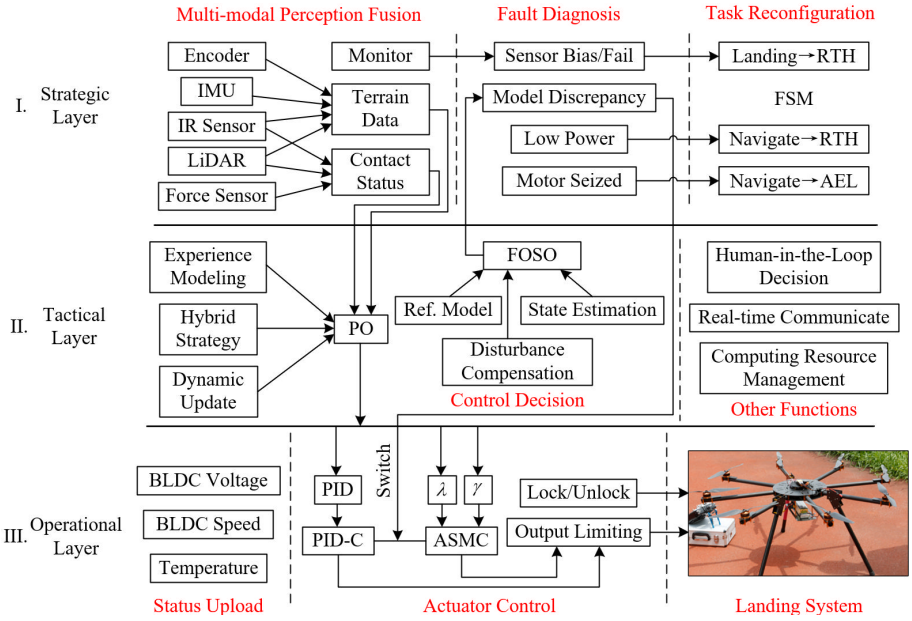

Fig. 11 Diagram of the adaptive hierarchical control framework.

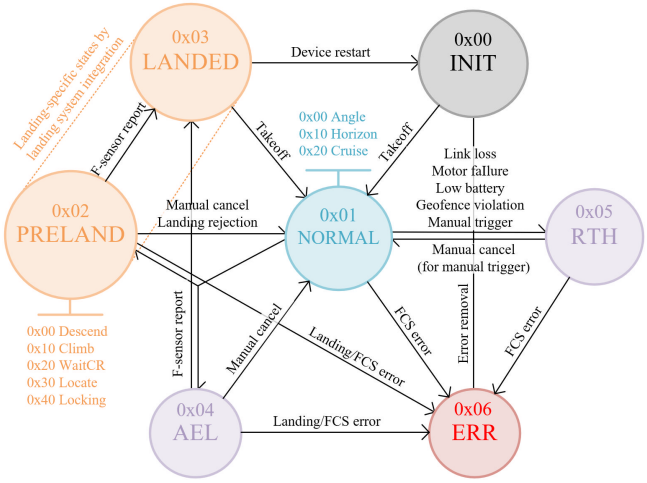

Fig. 12 Basic finite state machine framework for an intelligent multirotor UAV.

When subjected to instantaneous constant load disturbances (Fig. 10c), SMC fails to reject the disturbance because of its limited robustness and lack of adaptability, resulting in trajectory divergence. Other controllers maintain their performance. Figure 10d validates the adaptive estimation $\hat{\tau}_D$ and convergence of the disturbance torque τ_D.

Under multiplicative actuator faults (Fig. 10e), SMC exhibits significant steady-state errors. Both ASMC (adaptive compensation) and PID-C (integral compensation) resolve this issue, with ASMC demonstrating superior fault tolerance.

The proposed method represents an evolved form of PID-C, combining its rapid response with enhanced precision. Conventional SMC proves to be unsuitable for physical implementations because of its limited robustness. ASMC excels in noncontact applications (e.g., positioning in mechanical systems). PID-C remains effective in contact-mode operations involving complex disturbances (e.g., shock absorption control of a helicopter's landing gear during touchdown).

As shown in Table 3, we used six metrics—root mean square error (RMSE), integrated absolute error (IAE), integrated squared error (ISE), integrated time-weighted absolute error (ITAE), integrated time-weighted squared error (ITSE), and control energy (CE)—to comprehensively evaluate the controller's steady-state error, convergence speed, energy consumption, and robustness. For each metric, the first value represents the group mean, and the second value represents the group's standard deviation.

The proposed method achieved significant improvements in the first five metrics. Across all 50 simulation tests, the controller

Table 3. Statistical comparison of landing performance metrics (Case 4).

Metric	Units	Proposed (1)	SMC (2)	PID-C (3)	p_{12} value†	p_{13} value†
RMSE‡	10^{-3} m	6.79, 0.162	8.31, 0.244	29.5, 1.52	<0.001*	<0.001*
IAE‡	10^{-2} m·s	2.91, 0.138	3.90, 0.213	18.6, 1.94	<0.001*	<0.001*
ISE‡	10^{-4} m^2·s	4.61, 0.218	6.95, 0.380	87.2, 9.10	<0.001*	<0.001*
ITAE‡	10^{-2} m·s^2	2.89, 0.098	4.13, 0.253	42.7, 1.94	<0.001*	<0.001*
ITSE‡	10^{-4} m^2·s^2	2.03, 0.0859	4.09, 0.311	76.5, 4.36	<0.001*	<0.001*
CE‡	10^{-2} N·m·s	36.5, 1.10	35.7, 1.52	35.7, 0.917	0.998*	>0.999*

* Significant group difference (proposed vs. SMC and PID-C). † A battery of multiple univariate one-tailed *t*-tests (with Holm–Bonferroni correction) was conducted to compare the proposed controller with the two baseline controllers on all six performance indices (n = 50 trials per group, α = 0.01 with the total number of tests set to 10). ‡ RMSE is defined as $\sqrt{\frac{1}{T}\int_0^T e^2 dt}$; IAE is defined as $\int_0^T |e| dt$; ISE is defined as $\int_0^T e^2 dt$; ITAE is defined as $\int_0^T t|e| dt$; ITSE is defined as $\int_0^T te^2 dt$; CE is defined as $\int_0^T |\tau_M| dt$.

maintained consistently lower errors, outperforming SMC by 18.3%, 25.4%, 33.7%, 30.0%, and 50.4%, and surpassing PID-C by 77.0%, 84.4%, 94.7%, 93.2%, and 97.3%, respectively, for RMSE, IAE, ISE, ITAE, ITSE, and CE. Statistical analysis confirmed the superiority of the proposed method across all valid metrics (all $p < 0.001$ after Holm–Bonferroni correction). The low standard deviations indicate strong robustness against disturbance uncertainties, whereas the significantly lower p-values verify statistically meaningful improvements over both SMC and PID-C in terms of steady-state error and convergence speed.

Notably, these performance gains were achieved with only a 2.2% increase in energy output according to the CE metric. This implies that ASMC improved all key metrics by over 18% with nearly unchanged energy consumption—an outcome of considerable value for the design of controllers in onboard energy-limited systems.

This simulation required MATLAB/Simulink (Version R2024a). The computational platform was a desktop computer equipped with an AMD Ryzen 5 5600X central processing unit. The ode45 (Dormand–Prince) solver was used, with a relative tolerance of 1×10^{-3} to ensure the solution's accuracy.

The adaptive hierarchical control framework

Strategic–tactical–operational management system

Conventional control frameworks often used fixed hierarchies and parameters, which can lead to performance degradation or failure in dynamic environments where the terrain's geometry, disturbances, or task objectives change unpredictably. To address this limitation, we propose an adaptive hierarchical control framework (AHCF) that enables cross-layer adaptation through real-time sensor fusion and context-aware policy switching. The AHCF integrates a three-tier strategic–tactical–operational (STO) management system, as illustrated in Fig. 11.

Strategic layer

This layer orchestrates multimodal perception fusion, fault diagnosis, and mission reconfiguration. It synthesizes the terrain data and contact states from redundant sensor networks to maintain robust environmental awareness. A unified monitoring interface streams real-time state data to support fault diagnosis for both the flight control system (FCS), such as IMU drift and the landing system, including sensor failures, model discrepancies, and the actuator jamming. Critical system warnings include communication link loss (> 5 s), low battery, and abnormal motor voltage warnings.

As shown in Fig. 12, the mission reconfiguration subsystem governs the UAV's finite state machine (FSM), managing transitions between key states: initiation (INIT), normal, pre-landing (PRELAND), landed, autonomous emergency landing (AEL), return to home (RTH), and faulty (ERR). Critical faults trigger a transition from NORMAL to ERR, halting the UAV until resolution, whereas unresolved warnings initiate a NORMAL→RTH transition.

Operating at 10 Hz, this layer integrates peripheral data processing, calculating the terrain's flatness (DTFSF), and low-level hardware ISRs, updating a high-priority 1-byte status register.

Tactical layer

This layer implements a dual-module control architecture. A parameter optimizer (PO) used data-driven strategies to dynamically adjust the control parameters during contingencies. The PO module monitors the terrain's features and the system's response in real time, dynamically adjusting the ASMC's parameters and the DTFSF thresholds. For instance, upon detecting complex terrain, the PO can appropriately relax κ_1 to allow landing with a larger tilt angle while increasing the robustness gain of the ASMC to handle greater disturbances. A full-order state observer (FOSO), based on an approximate linearization of the landing leg's dynamics, provides real-time state estimates via a Kalman filter (KF) to evaluate the ASMC's performance. If the deviation between the ASMC output and the FOSO estimate exceeds a predefined robustness margin—indicating excessive model perturbation—the system immediately switches to the more conservative PID-C controller to prioritize stability. The switching logic is based on a threshold judgment of the FOSO estimation error, and a hysteresis comparator is adopted to avoid frequent switching. At the moment of switching, the system retains the output from the previous control cycle as the initial value, ensuring state continuity and a smooth transition while preventing any instantaneous jitter caused by the switching process.

Operational layer

Directly interfacing with the hardware, this layer monitors the brushless direct current (BLDC) motor's parameters (e.g., winding temperature, rotor position) and ambient conditions via auxiliary sensors. The control logic combines tunable PID controllers with ASMC, reinforced by output constraint enforcement. Mechanical safety is ensured through centralized locking and unlocking of the landing gear mechanism.

The STO architecture implements software-level scheduling via priority-based and timer-driven interrupts. To guarantee the control system's stability, the ASMC is assigned the highest priority, maintaining a stable 1-kHz execution frequency. Lower-priority processes (e.g., multimodal perception) are segmented and executed during the controller's idle cycles. Crucially, UAV fault detection is granted the highest interruption priority, enabling it to preempt all landing control system (LCS) operations in the event of an FCS failure. The AHCF's time synchronization structure is depicted in Fig. 13. Let the time proportions of the STO programs be denoted as P_S, P_T, P_O, respectively. We then have

$$P_S + \frac{P_T}{N_1} + \frac{P_O}{N_1 N_2} < 1 \tag{37}$$

Prelanding phase

This section details the sensor requirements and programming logic for the PRELAND phase. IMU, barometer, and global positioning system (GPS) data are fused by the flight controller for 6-DoF state estimation (three-dimensional position + three-dimensional orientation), determining the timing of the landing system's activation. Current landing methods include vision-only, beacon-based optical, and hybrid approaches (the sensors' trade-offs are shown in Table 4). Pure optical landing (LiDAR + IR) offers the highest energy efficiency.

As per the control strategy, landing Legs 1 and 2 use a controllable rocker mechanism for optimal impact absorption, whereas Leg 3 has a fixed buffer (the working logic is shown in Fig. 14).

In operation, the landing subprogram first uses LiDAR's IR ranging to measure footpad-to-ground distances. When all distances are below 5 m, the DTFSF activates to classify the terrain (enabling image-free autonomy) into three types on the basis of Control Register (CR) thresholds: Basic terrain (default strategy), obstacle

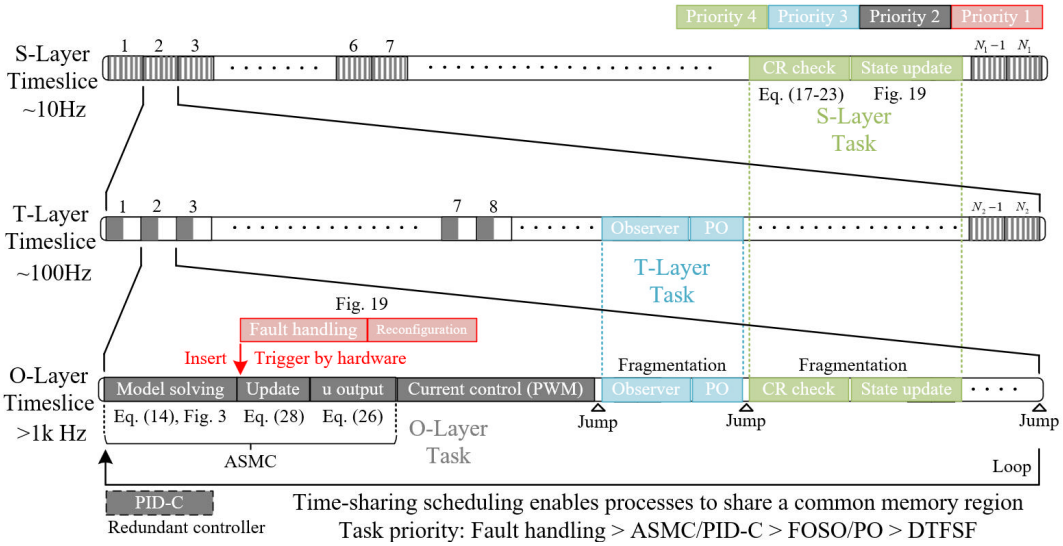

Fig. 13 Diagram of the AHCF's time synchronization with task priority.

Table 4. Sensors potentially required for the landing phase

Sensor	Primary role	Advantages	Limitations
IMU	Attitude estimation	High update rate, low latency	Attitude drift, vibration sensitivity
GPS	Global positioning	High accuracy, no drift, wide coverage	Low update rate, signal occlusion
Barometer	Relative altitude	Low cost, power-efficient	Thermal drift, airflow disturbance
IR/ultrasonic	Low-altitude sensing	A mm-level resolution without light	Short range, rain/fog attenuation
Stereo vision	Terrain mapping	SLAM capability, rich data	High computation, lighting-dependent
LiDAR	Precision ranging	Electromagnetic Interference (EMI)-resistant, cm-level accuracy, long range	High power, rain/fog attenuation
Force	Detecting collisions	Fast response, power efficient	Affected by vibration

terrain (strict strategy, adjusted precision/error limits), and extreme terrain (landing rejected).

The controller then sync-adjusts the legs until the vertical distance difference between the sensors is under 5 mm. Finally, the actuators lock for simultaneous touchdown, engaging all strut buffers upon contact.

During the adjustment of the rocker-actuated mechanism, severe deviations in the computed system model that exceed the robustness range of the ASMC may lead to significant chattering and instability. To mitigate this, an actuator saturation link and a controller switching strategy are implemented to maximize the controller's stability. If the UAV requires rapid landing, the landing subprogram can be switched to fast mode before activation at the terminal. This mode enables a rapid descent from an altitude of approximately 5 m within 5–10 s, but it is only suitable for safe terrains (flat surfaces or gentle slopes). To ensure terrain awareness, this mode requires the integration of low-latency remote video transmission capabilities in both the terminal and the UAV.

Field experiments

This section outlines the UAV platform's specifications, the computational load characteristics of the safe landing algorithm, and the results of the flight test. The platform's architecture and integrated sensors are first described, followed by a description of the system's performance parameters and configuration schemes for evaluating and identifying the landing zone typologies. Subsequently, the real-time processing capabilities of multiple modules within the AHCF are demonstrated through comprehensive test scenarios. Finally, the landing test data acquired in nominal environments are presented.

Configurations and real-time computation

The flight controller uses a Radiolink F722 octocopter system with an STM32F722RE MCU (216 MHz), an ICM42688 IMU, and an SPL06-001 barometer (altitude accuracy = 5 cm). Attitude angles from sensor fusion achieve ~0.1° accuracy. Critical flight data are output via Universal Asynchronous Receiver/Transmitter (UART)/USB for the multisubsystem LCS design.

We present a low-power, real-time embedded platform for UAV vertical takeoff and landing (VTOL) systems. The core is an STM32H743VI MCU (480 MHz Arm® Cortex®-M7 with an Floating Point Unit [FPU], 2424 CoreMark), balancing high computation and energy efficiency. It meets the demands of complex algorithms like ASMC and DTFSF. Key low-power features include:

(1) 275 μA/MHz in Run mode (sustained power < 1 W);

(2) 2.43 μA in standby mode.

Its 1 MB of Static Random-Access Memory (SRAM) and over 35 peripherals (e.g., high-speed Serial Peripheral Interface [SPI], Inter-Integrated Circuit [IIC], 16-bit Analog-to-Digital Converters [ADCs]) enable seamless sensor integration, forming a complete perception–control loop.

During standard high-altitude flight, the landing system is locked, and the controller is downclocked to 48 MHz to save energy. LiDAR runs at a reduced 5 Hz for avoidance, and the nonessential landing sensors are off. A 3-m altitude threshold triggers this power-saving mode, with the variable-frequency control logic detailed in Table 5.

A detailed analysis of the algorithm's execution time using the STM32H743VI (480 MHz Cortex-M7, FPU enabled) is presented in Table 6.

The real-time performance analysis reveals that the ASMC controller dominates the computational load (14% CPU usage at 1 kHz), constrained by double-precision FPU operations. In contrast, the PID controller achieves exceptional efficiency (1.2% usage)

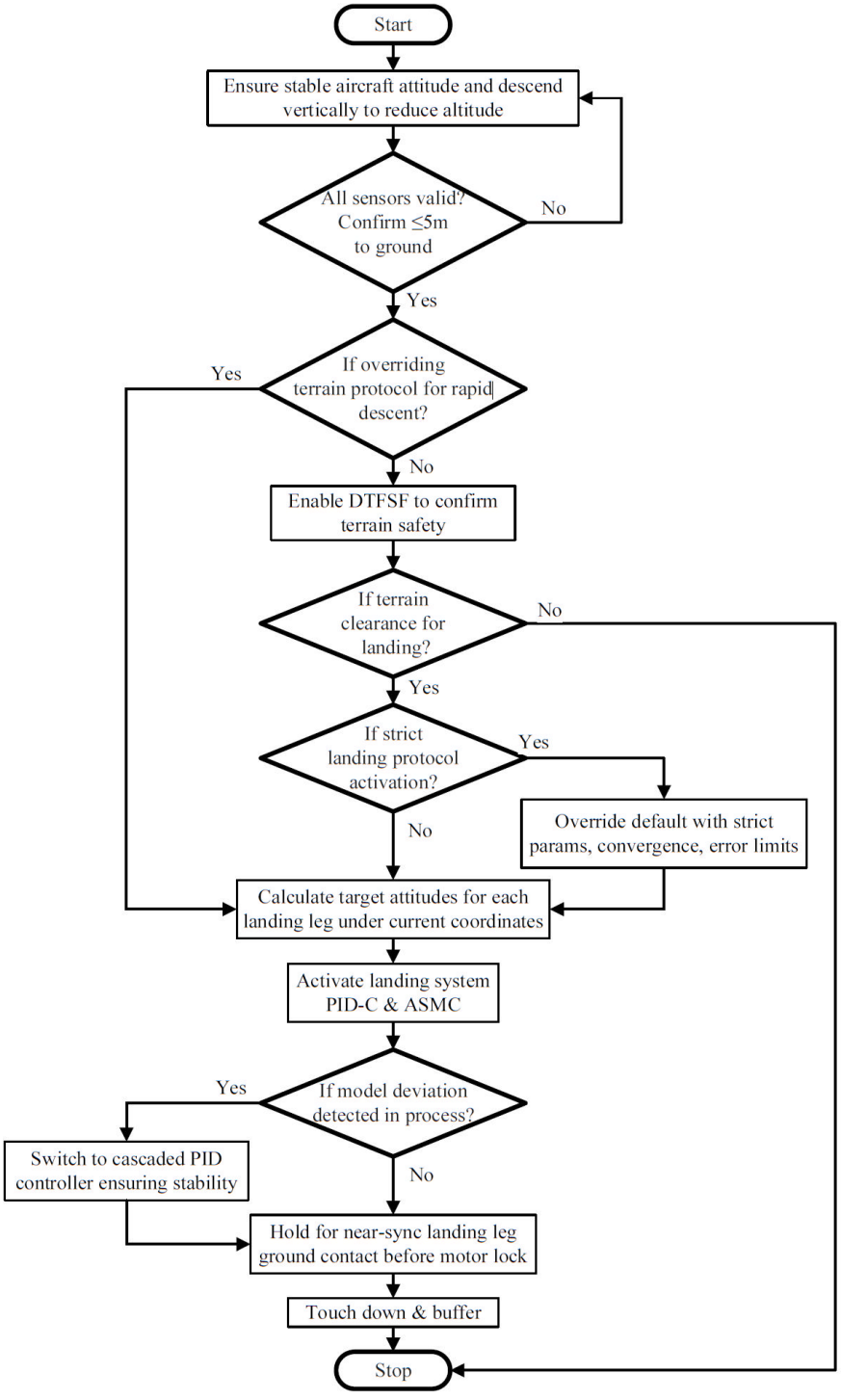

Fig. 14 Working principles and flowchart of the landing process.

Table 5. MCU and sensor duty cycling.

Stages	Flight MCU	DTFSF	Landing MCU	LiDAR	IR	Force
Cruise (>3 m)	100 MHz	10 kHz	48 MHz	5 Hz	OFF	OFF
Near ground (<3 m)	100 MHz	10 kHz	480 MHz	10 Hz	1 kHz	1 kHz

through lean implementation. Acceleration of the hardware proves critical: FOSO leverages Common Microcontroller Software Interface Standard - Digital Signal Processor (CMSIS-DSP) with FPU/DSP to maintain 3.2% occupancy at 100 Hz, whereas DTFSF exploits the Art Accelerator (ART)/I-cache to manage cycles of 2.4 ± 0.2 ms efficiently despite its memory-intensive workload. Crucially, the aggregate 20.8% total occupancy leaves substantial headroom (79.2%) for auxiliary tasks, validating the architecture's real-time capability. The ASMC's execution variance (± 15 μs) emerges as the primary optimization target for deterministic control.

Test results

As illustrated in Fig. 15, the prototype UAV in this study was flown along a predefined route A→B→C→D→E, with a total

Table 6. Analysis of the core algorithm's execution time

Modules	Frequency	Cycle	CPU usage	Optimization	Hardware dependency
DTFSF	10 Hz	2.4 ms	2.4%	CMSIS-DSP	ART accelerator + I-cache
FOSO	100 Hz	320 µs	3.2%	CMSIS-DSP	FPU + DSP
ASMC	1 kHz	140 µs	14%	FPU-accelerated	Double-precision FPU
PID-C	1 kHz	12 µs	1.2%	–	–

Covering the key modules and the system's components. Time data were measured using Keil Microcontroller Development Kit (MDK) optimization compilation (-O3) and the on-board Discrete Wavelet Transformation (DWT) counter.

ground-projected trajectory length of 127.8 m. To evaluate the terrain recognition capability, Segments BC and CD traversed an undulating grass field, followed by landing at the relatively flat Location E. The primary landing safety metric $CR_{0.95}$ was used, with values exceeding 80% indicating secure terrain. The experimental results encompass the full-process data from the system's initiation to landing, including:

(1) the UAV's relative altitude (sourced from the flight controller's serial output);

(2) the $CR_{0.95}$ curve (initialized 10 s after takeoff at 10 Hz, generated by the LCS);

(3) The landing system's total power consumption (covering the LCS, sensors, and two BLDC motors in the landing legs);

(4) The system's status (1 byte of data, output by the LCS).

Figure 16a–c presents both the raw data and filtered data processed with a moving average filter (window size = 10).

As depicted in Fig. 16a, Points A–E on the map correspond to the labeled positions in the trajectory plot. During the mission, the UAV executed two manual ascents, one manual descent and one autonomous descent. The first ascent (0.2–2 m) validated the system's stress response under the 3-m threshold (see Row 1, Table 5). The second ascent (2–3.5 m) assessed low-power operation above the threshold. The manual descent to 0.7 m minimized the sensors' measurement errors near the ground. Subsequent autonomous descent triggered landing procedures with rapid leg actuation for a low-impact touchdown.

In Fig. 16b, the CR index computed at 10 Hz by the DTFSF reflects undulation within a 2-m radius—approximately triple the UAV's size—satisfying the measurement area requirement in Section 2.1. The horizontal velocity averaged 1 m/s. Notable drops in CR (80%–50%) occurred over segments BC/CD, whereas the values exceeded 80% on flat terrain, confirming the metric's sensitivity to localized terrain complexity.

Figure 16c demonstrates the power consumption, aligning with Table 5's low-energy profile.

(1) Ground standby < 5 W. With LiDAR deactivated before takeoff, the controller and peripherals maintained their power consumption below 5 W.

(2) Low-altitude flight: 15–20 W. At altitudes of < 3 m, all sensors were activated.

(3) High-altitude flight: 10–15 W. Above 3 m, the leg-mounted ranging/force sensors were disabled, whereas LiDAR operated at a scaled-down frequency, reducing the power draw by 5 W.

(4) Landing phase: 20–40 W. The dual 12-V landing leg BLDC motor exhibited a peak power draw of 20 W.

As indicated in Fig. 16c, the LCS's average power consumption of 15 W enables the miniaturization of UAVs below 0.2 m, subject to a dual-series (> 5 V) power supply and a > 300 g payload capacity (encompassing the compact landing leg, low-power LiDAR, sensors, and airframe).

Figure 16d documents state transitions: INIT → normal (horizon mode) → PRELAND (descent, leg positioning, locking) → LANDED. Horizon mode maintained a stable attitude during rapid manual maneuvers (ascent, descent, lateral motion), proving essential for A→ E navigation.

These experimental results demonstrate that the octocopter with adaptive multilevel control achieved the following:

(1) Low-power terrain scanning;

(2) 10-Hz mode switching;

(3) Precision leg control;

(4) Real-time embedded operation (a 1-kHz control loop, 20% CPU load).

Fig. 15 Landing procedure in a real environment.

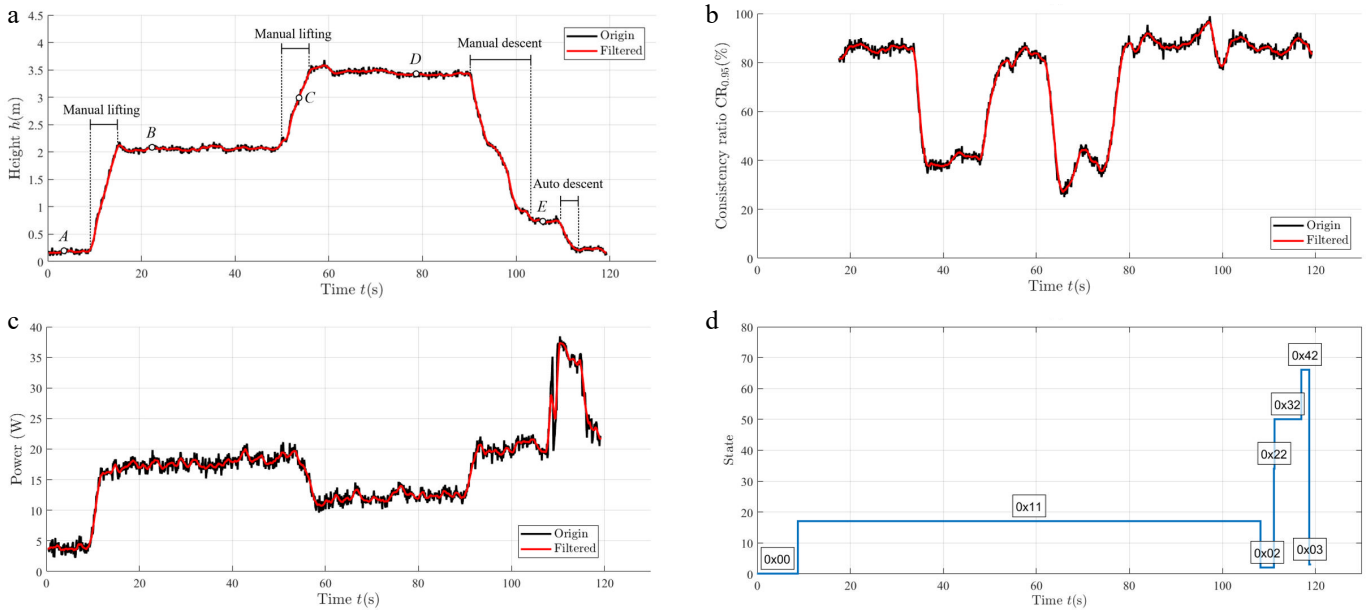

Fig. 16 Data from the entire process from INIT to LANDED. (a) Flight; (b) perception; (c) power; (d) status.

This validates the simulations in the previous sections.

Conclusions

We propose an AHCF with integrated sensor fusion for intelligent multirotor UAV landing systems.

(1) A bio-inspired rocker bogie landing mechanism reduces the complexity of control while ensuring kinematic redundancy for robust terrain adaptation. The Lagrangian-based dynamic model, enhanced by real-time attitude perception, enables precise compensation for gravity and low-coupling motion control.

(2) The DTFSF algorithm fuses IMU, LiDAR, range finder, and force sensor data to ensure computational efficiency and resilience to noise. Validated on diverse terrains, the algorithm distinguishes basic (flat/ramp), obstacle (continuous/discrete), and extreme terrain types with > 95% accuracy at a window width of 20.

(3) The ASMC controller with saturation-function-based disturbance adaptation guarantees Lyapunov stability. Numerical simulations demonstrated ASMC's superiority over conventional and nonadaptive methods in terms of rapid convergence (< 4 s), suppressed chattering, and robustness to disturbances and actuator faults (20% loss).

(4) The STO framework integrates multimodal sensor fusion, fault diagnosis, and ASMC, achieving synchronized data interaction across the strategic, tactical, and executive layers at < 1-ms latency.

Unifying mechanical design, environmental perception, and adaptive control with high-efficiency implementations, the system achieves real-time operation at 1 kHz on a Cortex-M7 MCU. This provides a unified solution for safe landing in unstructured environments, enabling miniaturization and concealment of landers.

Author contributions

The authors confirm their contributions to the paper as follows: study conception and design, draft manuscript preparation: Huang M; data collection, analysis and interpretation of results: Zhang W. Both authors reviewed the results and approved the final version of the manuscript.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgments

This work was supported by the project supported by the Open Fund of the National Key Laboratory of Air Traffic Management System in 2025, Grant No.: SKLATM202504; in part by the Natural Science Foundation of China, 62503043, Mingyang HUANG; in part by the China Postdoctoral Science Foundation, No.18 Special Funding under Grant No.: 2025T180469, Mingyang HUANG; in part by the Talented Person Research Start Funds from University of Science and Technology Beijing (Grant No. 00007796), Mingyang HUANG; in part by High-end Foreign Experts Recruitment Plan of China in 2025, S20250215, Mingyang HUANG; in part by High-end Foreign Experts Recruitment Plan of China in 2025, H20251101, Mingyang HUANG; in part by High-end Foreign Experts Recruitment Plan of China, Y20240265, Mingyang HUANG; in part by the University of Science and Technology Beijing, Nanjing University of Aeronautics and Astronautics, 2024BK7119, Mingyang HUANG; and in part by the Fundamental Research Funds for the Central Universities (FRFCU) Future Exploration Project for Young Teachers (FRF-TP-25-027), Mingyang HUANG. This paper was supported by Research Fund of the State Key Laboratory of Mechanics and Control for Aerospace Structures (Nanjing University of Aeronautics and Astronautics), No. MCAS-E-0226Y02, Mingyang HUANG.

Dates

Received 31 July 2025; Revised 15 January 2026; Accepted 27 January 2026; Published online 16 March 2026

References

[1] Langley J, Costello M. 2016. Ground resonance mitigation of rotorcraft with articulated robotic landing gear: a preliminary investigation.

Proceedings of the Vertical Flight Society 72nd Annual Forum. May 17–19, 2016. West Palm Beach, Florida. US: The Vertical Flight Society. pp. 1–15

[2] Roderick WRT, Cutkosky MR, Lentink D. 2021. Bird-inspired dynamic grasping and perching in arboreal environments. *Science Robotics* 6:eabj7562

[3] Sarkisov YS, Yashin GA, Tsykunov EV, Tsetserukou D. 2018. DroneGear: a novel robotic landing gear with embedded optical torque sensors for safe multicopter landing on an uneven surface. *IEEE Robotics and Automation Letters* 3:1917–1924

[4] Ikura M, Miyashita L, Ishikawa M. 2021. Stabilization system for UAV landing on rough ground by adaptive 3D sensing and high-speed landing gear adjustment. *Journal of Robotics and Mechatronics* 33:108–118

[5] Zufferey R, Tormo-Barbero J, Feliu-Talegón D, Nekoo SR, Acosta JÁ, et al. 2022. How ornithopters can perch autonomously on a branch. *Nature Communications* 13:1–11

[6] Boix DM, Goh K, McWhinnie J. 2017. Modelling and control of helicopter robotic landing gear for uneven ground conditions. *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Linköping, Sweden, 3–5 October 2017*. USA: IEEE. pp. 60–65 doi: 10.1109/red-uas.2017.8101644

[7] Li X, Dunkin F, Dezert J. 2024. Multi-source information fusion: progress and future. *Chinese Journal of Aeronautics* 37:24–58 (in Chinese)

[8] Kibrete F, Engida Woldemichael D, Shimels Gebremedhen H. 2024. Multi-Sensor data fusion in intelligent fault diagnosis of rotating machines: a comprehensive review. *Measurement* 232:114658

[9] Yang Z, Li G, Xue G, He B, Song Y, et al. 2025. A novel multi-sensor local and global feature fusion architecture based on multi-sensor sparse Transformer for intelligent fault diagnosis. *Mechanical Systems and Signal Processing* 224:112188

[10] Xu Z, Chen X, Xu J. 2025. Multi-modal multi-sensor feature fusion spiking neural network algorithm for early bearing weak fault diagnosis. *Engineering Applications of Artificial Intelligence* 141:109845

[11] Parsons T, Hanafi Sheikhha F, Seo J, Lee H. 2024. RGB-LiDAR sensor fusion for dust de-filtering in autonomous excavation applications. *Automation in Construction* 168:105850

[12] Lv J, Kim BG, Parameshachari BD, Slowik A, Li K. 2025. Large model-driven hyperscale healthcare data fusion analysis in complex multi-sensors. *Information Fusion* 115:102780

[13] Sun C, Lin Y, Meng Q, Li L. 2025. Adaptive output feedback fault-tolerant control for a class of nonlinear systems based on a sensor fusion mechanism. *ISA Transactions* 156:457–467

[14] Wu D, Qiao B, Du C, Zhu Y, Yan F, et al. 2024. Multi-objective dynamic coordinated Adaptive Cruise Control for intelligent electric vehicle with sensors fusion. *Mechanical Systems and Signal Processing* 209:111125

[15] Hussain KF, El-Din Safwat N, Thangavel K, Sabatini R. 2025. Space-based debris trajectory estimation using vision sensors and track-based data fusion techniques. *Acta Astronautica* 229:814–830

[16] Wang P, Ji H, Zhang Y, Zhu Z. 2025. Consistent fusion for distributed multi-rate multi-sensor linear systems with unknown correlated measurement noises. *Chinese Journal of Aeronautics*

[17] Xu L, Shi C, Yu H, Hou L. 2024. Fusion estimation for state-saturated multi-rate systems with integral measurement and sensor resolution. *Journal of the Franklin Institute* 361:107281

[18] Ban C, Wang L, Chi R, Su T, Ma Y. 2024. A Camera-LiDAR-IMU fusion method for real-time extraction of navigation line between maize field rows. *Computers and Electronics in Agriculture* 223:109114

[19] Sestras P, Badea G, Badea AC, Salagean T, Oniga VE, et al. 2025. A novel method for landslide deformation monitoring by fusing UAV photogrammetry and LiDAR data based on each sensor's mapping advantage in regards to terrain feature. *Engineering Geology* 346:107890

[20] Liu H, Xu G, Liu B, Li Y, Yang S, et al. 2025. A real time LiDAR-Visual-Inertial object level semantic SLAM for forest environments. *ISPRS Journal of Photogrammetry and Remote Sensing* 219:71–90

[21] Saldiran E, Hasanzade M, Cetin A, Inalhan G. 2025. Ensuring operation time safety of VTOL UAV: autonomous emergency landings in unknown terrain. *IEEE Transactions on Aerospace and Electronic Systems*

[22] Chen K, Sun Q, Sun H, Liu Q, Chen Z. 2025. Tightly coupled lidar-inertial-GPS environment detection and landing area selection based on powered parafoil UAV. *IEEE Transactions on Instrumentation and Measurement* 74:2500116

[23] Jiao J, Geng R, Li Y, Xin R, Yang B, et al. 2025. Real-time metric-semantic mapping for autonomous navigation in outdoor environments. *IEEE Transactions on Automation Science and Engineering* 22:5729–5740

[24] Saleem O, Kazim M, Iqbal J. 2025. Robust position control of VTOL UAVs using a linear quadratic rate-varying integral tracker: design and validation. *Drones* 9:73

[25] Miranda-Villatoro FA, Brogliato B, Castaños F. 2017. Set-valued discrete-time sliding-mode control of uncertain linear systems. *IFAC-PapersOnLine* 50:9607–9612

[26] Fan X, Wang Z. 2020. Event-triggered integral sliding mode control for linear systems with disturbance. *Systems & Control Letters* 138:104669

[27] Labbadi M, Muñoz-Vázquez AJ, Djemai M, Boukal Y, Zerrougui M, et al. 2022. Fractional-order nonsingular terminal sliding mode controller for a quadrotor with disturbances. *Applied Mathematical Modelling* 111:753–776

[28] Qu Y, Zhang B, Chu H, Shen H, Zhang J, et al. 2023. Sliding-mode anti-disturbance speed control of permanent magnet synchronous motor based on an advanced reaching law. *ISA Transactions* 139:436–447

[29] Cao G, Jia Z, Wu D, Li Z, Zhang W. 2024. Trajectory tracking control for marine vessels with error constraints: a barrier function sliding mode approach. *Ocean Engineering* 297:116879

[30] Irfan S, Zhao L, Ullah S, Javaid U, Iqbal J. 2024. Differentiator- and observer-based feedback linearized advanced nonlinear control strategies for an unmanned aerial vehicle system. *Drones* 8:527

[31] Rakhtala SM, Ahmadi M. 2017. Twisting control algorithm for the yaw and pitch tracking of a twin rotor UAV. *International Journal of Automation and Control* 11:143–163

[32] Ebrahimpour M, Lungu M, Kakavand M. 2024. Antisaturation fixed-time backstepping fuzzy integral sliding mode control for automatic landing of fixed-wing unmanned aerial vehicles. *Journal of the Franklin Institute* 361:107185

[33] Ali K, Ullah S, Mehmood A, Mostafa H, Marey M, et al. 2022. Adaptive FIT-SMC approach for an anthropomorphic manipulator with robust exact differentiator and neural network-based friction compensation. *IEEE Access* 10:3378–3389

[34] Ott C, Albu-Schaffer A, Kugi A, Hirzinger G. 2008. On the Passivity-Based Impedance Control of Flexible Joint Robots. *IEEE Transactions on Robotics* 24:416–429