RflySimSaT: A Safety Assessment Platform for UAVs Based on Hardware-in-the-Loop Simulation

Xunhua Dai, Jinhu Tu, Yong Chen, and Quan Quan, *Senior Member, IEEE*

***Abstract*—As unmanned aerial vehicles (UAVs) lead the way in the development of future digital smart cities, they continue to face scrutiny due to safety concerns. While robotics simulators offer UAVs efficient and cost-effective testing environments, they often lack comprehensive safety design considerations and user testing requirements. In this study, we introduce RflySimSaT, a dedicated safety testing platform for UAVs that addresses safety factors throughout the entire lifecycle. This platform incorporates diverse fault testing scenarios, high-fidelity dynamic models, an integrated safety assessment framework, standardized testing procedures, and customizable interfaces. The modular architecture and deployment of RflySimSaT facilitate plug-and-play cross-platform closed-loop safety testing. Users simply need to supply their aircrafts and autopilots, enabling them to efficiently navigate the phases of development, deployment, testing, and assessment using the customizable modules and standardized processes offered by the platform. To validate RflySimSaT's credibility and versatility, we designed various test cases that demonstrate its practicality and scalability. Additionally, we provide a comprehensive user manual, detailed case studies, and a rich fault dataset. The platform is open-source and available at: https://github.com/RflySim/RFlySimSafe/tree/RflySimSaT**

***Note to Practitioners*—As the use of UAVs grows in smart cities, ensuring their safety has become a critical concern. While existing simulation platforms are efficient for testing, they often lack comprehensive safety design and user testing capabilities. To address this gap, we introduce RflySimSaT, a UAV safety assessment platform based on HIL simulation. The platform combines a variety of fault-testing scenarios, precise dynamic models, an integrated safety assessment framework, standardized testing protocols, and customizable interfaces, offering a holistic solution for UAV safety assessment. Its modular architecture supports cross-platform, closed-loop safety testing with plug-and-play functionality. Users can easily conduct development, deployment, testing, and safety assessments by simply providing their own UAVs and autopilots. The platform also includes a user manual, case studies, and a rich fault dataset, enabling users to quickly get started and perform comprehensive safety testing.**

***Index Terms*—Safety assessment, hardware-in-the-loop (HIL) simulation, unmanned aerial vehicle (UAV).**

Received 21 February 2025; revised 29 July 2025 and 27 October 2025; accepted 3 January 2026. Date of publication 6 January 2026; date of current version 22 January 2026. This article was recommended for publication by Associate Editor A. Freddi and Editor C. Yang upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62406345, Grant 62573021, and Grant 62003374; in part by Hunan Provincial Natural Science Foundation of China under Grant 2025JJ50341; and in part by the National Key Research and Development Program of China under Grant 2025YFE0216900. *(Corresponding author: Jinhu Tu.)*

Xunhua Dai is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: dai.xh@csu.edu.cn).

Jinhu Tu and Yong Chen are with the School of Automation, Central South University, Changsha 410083, China (e-mail: tjhcsu@csu.edu.cn; chenyong@csu.edu.cn).

Quan Quan is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: qq_buaa@buaa.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TASE.2026.3651543, provided by the authors.

Digital Object Identifier 10.1109/TASE.2026.3651543

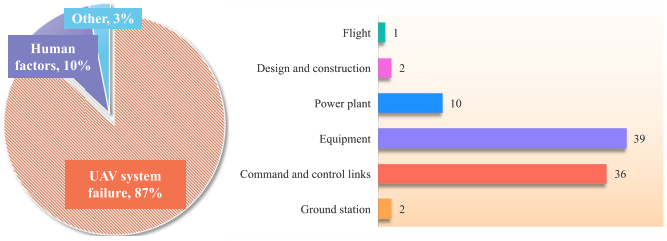
List of Abbreviations

Abbreviation	Expansion
AP	Action priority.
ATI	Automatic testing interface.
DLL	Dynamic link library.
ELOS	Equivalent level of safety.
FII	Fault injection interface.
FMEA	Failure mode and effects analysis.
FPGA	Field programmable gate array.
GPS	Global positioning system.
HIL	Hardware in the loop.
IMU	Inertial measurement unit.
MTBF	Mean time between failure.
PWM	Pulse width modulation.
RL	Risk level.
RPN	Risk priority number.
ROS	Robot operating system.
SAI	Safety assessment interface.
SIL	Software in the loop.
UE5	Unreal engine 5.
USB	Universal serial bus.
UAV	Unmanned aerial vehicle.
VTOL	Vertical take-off and landing.

I. Introduction

A. Background

UAVs, due to their compact size, versatility, agility, and cost-effectiveness, are increasingly utilized across various sectors including military, agriculture, and logistics [1]. As vital components in the evolution of digital smart cities and aerial transportation, UAVs are essential for future advancements [2]. However, the rapid growth in demand for UAV technology has outpaced government regulations and oversight, leading to significant safety concerns [3]. As illustrated in Fig. 1(a), UAV accidents are predominantly attributed to system-level failures [4], which constitute the



(a) Statistics on causes of UAV accidents

(b) Statistics on the frequency of subcomponent failures that lead to UAV system failures

Fig. 1. Statistics of accidents caused by UAV safety issues [4].

primary root causes of loss of control, crashes, and mission failures. Furthermore, Fig. 1(b) presents the statistical distribution of component-level failures, revealing that equipment malfunctions—including failures in flight control units, actuators, and sensors—account for the majority of safety incidents. Frequent incidents involving UAVs have eroded public trust, highlighting the urgent need for standardized operations and rigorous safety assessments based on established airworthiness standards [5], [6]. Implementing these measures will enhance the reliability and safety of UAV operations while supporting the development of effective hazard mitigation and emergency management strategies [7], [8].

Real-world experiments have long been the most effective way to enhance the safety of UAV critical systems [9]. However, the complexity of UAV components and the multitude of test cases make integrated experimental testing costly, inefficient, and risky, often hindered by airspace restrictions and environmental limitations [10]. Consequently, relying solely on experimental safety assessments is impractical, as these methods are hard to automate and fail to provide comprehensive evaluations. Advancements in simulation technology offer a more efficient and cost-effective approach to validate UAV safety by simulating real-world fault scenarios [11]. Designing a wide range of test scenarios is essential for identifying safety vulnerabilities [12] and developing simulation models that accurately reflect the dynamics of atypical aircraft behavior [13]. This enables the application of experimental techniques for safety assessment and the establishment of hazard-based test scenarios [14]. As high-precision modeling [15], system identification, and simulation credibility assessment [16] continue to advance, safety testing methods based on credible simulations [17], [18] are emerging as a significant trend in UAV testing and safety assessment.

B. Related Work and Motivations

The growing application of simulation technology in robotics has significantly improved the efficiency of algorithm development and validation while reducing operational risks in real deployments [19]. Particularly, the concept "Sim2Real" has gradually made it feasible to quickly migrate simulations to real-world operations [20]. Recent advancements include digital twins [21], high-fidelity simulations [22], virtual reality [23], and HIL simulation [24], which are evolving rapidly. Using digital simulations to replicate both normal and fault scenarios during flight missions allows for real-time system monitoring and evaluation, garnering more attention in the field. Many simulation systems now incorporate cutting-edge real-time rendering engines like UE5 and Unity3D, facilitating immersive virtual reality experiences and semi-physical simulation capabilities for autopilot hardware. Platforms such as Microsoft's AirSim [25] and Google's FlightGoggles [26], along with others like Gazebo [27], Webots [28], MAV3DSim [29], RotorS [30], XTDrone [31], and FlightMare [32], are widely utilized in UAV development, significantly enhancing both development and testing efficiency. However, despite these advances, several critical challenges remain unresolved, particularly in the context of safety assessment.

i) While simulation-based methods offer clear advantages in terms of cost and scalability, they often fall short in supporting comprehensive safety validation across the UAV lifecycle. As highlighted in recent studies, current simulation platforms typically prioritize task validation objectives—such as path planning, visual perception, or control tuning—over system-level safety assurance. Consequently, key components such as fault injection mechanisms, structured safety indicators, and credibility validation pipelines are either missing or underdeveloped, limiting their applicability to real-world safety-critical applications.

ii) Although HIL techniques enhance the realism of control loop testing by incorporating actual hardware, they do not guarantee consistency during the modeling and rendering phases. Inadequate model fidelity, coarse fault representation, and limited environmental variability all contribute to simulation-reality mismatch, raising concerns about the trustworthiness of safety conclusions drawn from such platforms. These limitations underscore the need for systematic, measurable, and verifiable safety assessment methods capable of bridging the gap between simulation and real-world deployment.

iii) Existing research efforts in the domain of UAV safety assessment and simulation testing—while making progress in algorithm evaluation [33], safety modeling [34], and indicator formulation [35]—often remain confined to theoretical constructs or offline analyses. Many approaches focus narrowly on individual components or static indicators, without adequately accounting for the dynamic safety evolution of the system during task execution. Even those that introduce real-time evaluation frameworks [36] typically lack deep integration with system-level modeling, HIL simulation, and empirical validation workflows, thereby hindering their practical utility.

Due to the complexity of UAV safety, existing platforms often lack integration, reusability, and validation for real-world deployment [37]. This motivates our development of a unified safety testing platform integrating modeling, fault injection, and evaluation. To this end, we have conducted extensive foundational preliminary work, focusing on platform design, credibility assessment, and safety theory.

i) Regarding platform design, we developed the RflySim platform [38], which provides a unified modeling frame-

TABLE I
COMPARISON BETWEEN RFLYSIMSAT AND OTHER SIMULATION PLATFORMS

Features	Simulation Platform						
	MAV3DSim	FlightGoggles	Gazebo	RotorS	AirSim	XTDrone	**RflySimSaT**
Physical Model	Coarse-grained	Coarse-grained	Fine-grained	Coarse-grained	Fine-grained	Coarse-grained	Fine-grained
Graphic Fidelity	Low	High	Low	Middle	High	Middle	High
Sim2Real Support	HIL	HIL	SIL, HIL	HIL	HIL	SIL	SIL,HIL,Mixed
Function Support	Low	Middle	Middle	Low	Middle	Low	High
Sensors Support	✓	✓	✓	✓	✓	✓	✓
FII Support	✘	✘	✘	✘	✘	✘	✓
ATI Support	✘	✘	✘	✘	✘	✘	✓
SAI Support	✘	✘	✘	✘	✘	✘	✓

work for various aircraft types. In that stage, HIL integration was introduced using FPGA-based autopilot and sensor interfaces, enhancing the platform's extensibility and execution efficiency. However, this prior platform primarily focused on modeling and simulation capabilities and did not yet incorporate a systematic safety evaluation process or indicator-based assessment framework.

ii) Regarding credibility assessment, we proposed a HIL-based credibility evaluation method [39] that ensures the trustworthiness of simulations at both software and hardware levels. While this approach ensured the fidelity of the HIL simulation environment, it remained centered on simulation validation rather than safety modeling or risk feedback mechanisms.

iii) Regarding safety theory, we introduced a robust safety index and assessment method based on fuzzy reliability theory [40], which accounts for the evolution of system safety performance. This formulation enabled qualitative classification and quantitative estimation of UAV safety states. Nevertheless, the method was not yet aligned with airworthiness requirements and lacked deep integration into a full platform-level implementation or assessment workflow.

C. Contributions

Based on previous research, this work addresses practical UAV safety testing needs by proposing a unified safety assessment framework and workflow spanning development to deployment. We implement this framework in the HIL-integrated platform RflySimSaT, enabling real-time, closed-loop evaluation from modeling to execution. Table I provides a comparative analysis of RflySimSaT against other simulators based on key attributes. Compared to traditional simulation platforms, the proposed RflySimSaT platform offers significant theoretical advancements and practical capabilities for UAV safety assessment. The contribution of this work is embodied at three key levels:

i) Unified integration of safety assessment methodologies: RflySimSaT integrates online profust safety modeling, offline FMEA, and a credibility verification mechanism into a unified framework. This enables automated classification of system-level risks and traceable, dual-dimensional evaluation that considers both safety and credibility under a consistent modeling paradigm.

ii) End-to-end abstraction and plug-and-play usability: RflySimSaT provides a modular workflow from modeling to reporting, supporting plug-and-play UAV models and autopilots for reproducible and portable engineering.

iii) Cross-platform extensibility and real-world applicability: By combining HIL-based simulation and real-world sensor data, RflySimSaT supports diverse data formats and real-time communication via MAVLink. This ensures credible, reproducible assesses across various scenarios, enabling broad applicability in both academic research and industrial deployment.

D. Paper Organization

The organization of this paper is as follows: Section II presents the architecture of the RflySimSaT. Section III introduces the unified safety assessment framework. Section IV describes the unified safety testing and assessment process. In Section V, the safety assessment framework and standardized testing and assessment process are utilized to simulate and validate the flight of a multirotor UAV under fault conditions in both simulated and real flight scenarios. Section VI concludes the paper.

II. UAV SAFETY ASSESSMENT PLATFORM ARCHITECTURE

As illustrated in Fig. 2, RflySimSaT primarily consists of a motion simulator CopterSim (Block ①), a lower controller (Block ②), a 3D environment engine RflySim3D (Block ③), an upper controller (Block ④), and a safety assessment model RflySimSA (Block ⑤). Together, these components form a closed-loop system that accommodates both SIL and HIL simulations for safety assessment.

A. CopterSim

As a core component of RflySimSaT, CopterSim is responsible for the real-time execution of the simulation model and data interaction with the controller, serving as a central data relay. The high-fidelity simulation model is developed in MATLAB/Simulink, utilizing automatic code generation to create a DLL. This DLL is then imported into CopterSim for real-time execution, ensuring the reliability of the simulation software. CopterSim receives motor control signals from the controller via external interfaces, driving the simulation model in real-time and on a periodic basis. It transmits output

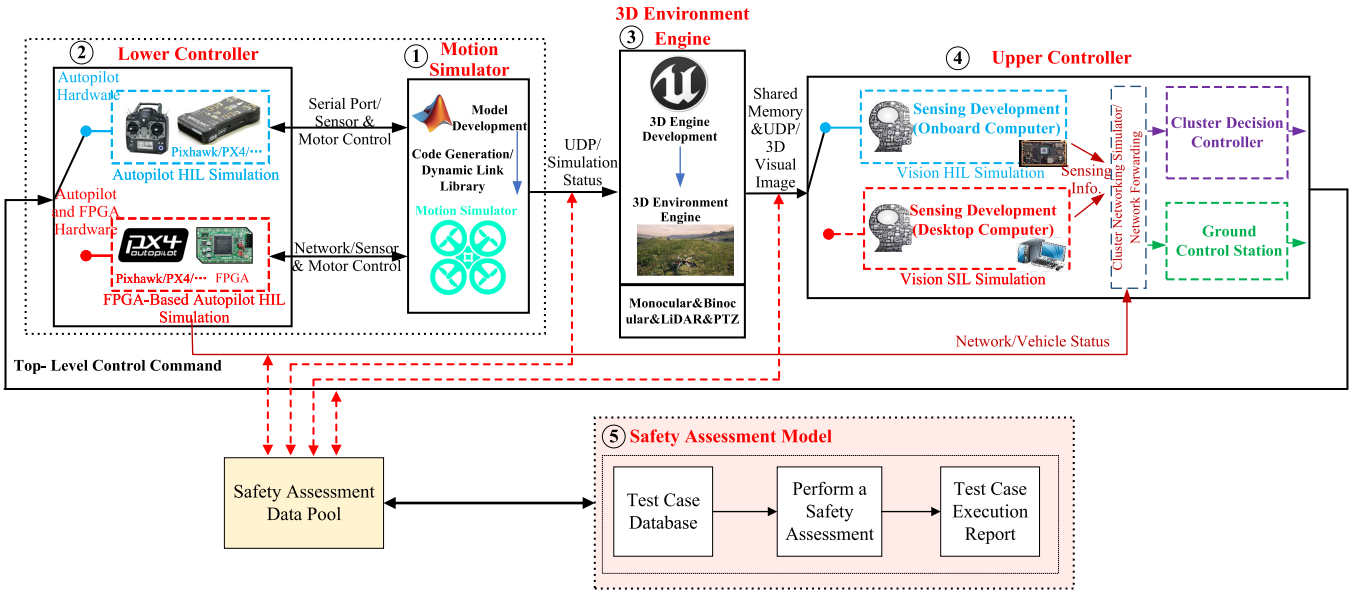


Fig. 2. Theoretical architecture of RflySimSaT platform.

sensor data—such as acceleration, GPS, and magnetometer readings—back to the controller for normal operation. Additionally, CopterSim sends simulation state data, including vehicle attitude and motor information, to RflySim3D, allowing for the visualization of the UAV's motion in a graphical environment.

B. Autopilot

The lower controller (autopilot) consists of both software and hardware components that work together to execute commands for attitude, velocity, and position control, driving UAV movement. HIL simulations are conducted using actual autopilot hardware in conjunction with a real-time simulation computer. We also offer an FPGA-based version of the HIL simulation [39]. This version achieves nanosecond-level real-time updates, enabling high-performance, code-free FPGA-based simulation of vehicle dynamics, sensors, and circuits with flexible, scalable support for diverse autopilot systems.

C. RflySim3D

RflySim3D, developed using the latest UE5, is a powerful tool that delivers real-time rendering and realistic simulation effects. It integrates data from the motion simulation model and provides several key functionalities:

i) Physical Collision Detection: RflySim3D includes mechanisms for detecting collisions between the UAV and ground objects, as well as between multiple vehicles.
ii) Network Communication Performance: The tool assesses communication performance in environments with obstacles such as buildings, trees, and jammers, taking into account various radiation angle ranges.
iii) Integration of Global-Scale Satellite Imagery: RflySim3D supports the loading of satellite imagery for large-scale scenes and enables the visualization of various sensor data, including RGB images, depth maps, and laser point clouds.

D. Upper Controller

The upper controller in our framework is not merely a conventional ground control station, but a modular and extensible high-level decision-making subsystem that integrates perception processing, networked communication, and swarm coordination. Specifically, it comprises: i) a visual perception layer for processing sensor data and generating situational awareness; ii) a networking simulation layer that emulates realistic inter-UAV communication topologies; and iii) a swarm control layer that encompasses both traditional ground station interfaces and distributed swarm decision-making modules. The input data for the upper controller comes from RflySim3D's visual sensors, such as camera images and laser point clouds, as well as from the autopilot's vehicle state data, including attitude, position, speed, and acceleration. The output consists of top-level control commands—such as unlock, take-off, desired speed, desired position, and desired route—that are sent to the autopilot. This controller primarily simulates perceptual information, network status, and cluster planning.

E. RflySimSA

As one of the core components of the RflySimSaT framework (corresponding to Block ⑤ in Fig. 1), is responsible for multi-source heterogeneous data fusion and system-level safety assessment. This module integrates data from autopilot logs, CopterSim simulation results, RflySim3D perception outputs, and task state feedback from the upper controller. Based on a standardized safety evaluation metrics system, it performs comprehensive quantitative assessment of performance, safety,

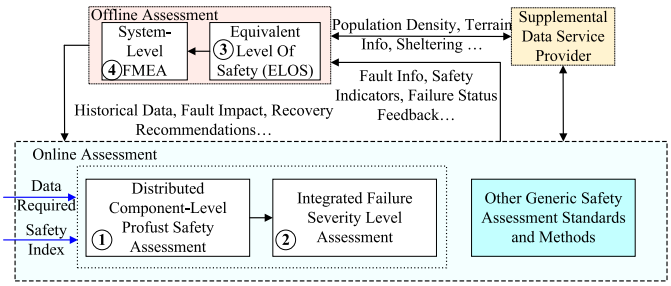
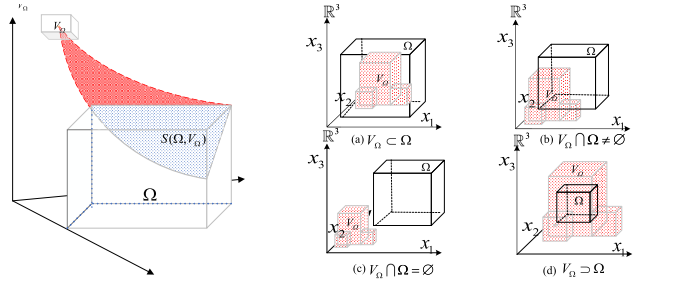


Fig. 3. Unified safety assessment framework of UAV.



(A) Safety analysis of rate model based on fuzzy risk space ($\mathbb{R}^3$ space)

(B) Analysis of the relationship between fuzzy risk space and causative events ($\mathbb{R}^3$ space)

Fig. 4. Safety analysis of profust Safety based on fuzzy logic [40].

and robustness throughout the mission execution process. A detailed description of this module is provided in Section III.

III. Unified Safety Assessment Framework

In this paper, the goals of safety analysis and assessment are twofold: (i) to determine whether the provided autopilot and vehicle meet minimum safety objectives; and (ii) to generate a FMEA table that identifies and assesses potential failure modes, their impacts on performance, safety, and reliability, along with recommendations for improvement and decision-making strategies. To address these objectives, a unified safety assessment framework is proposed to tackle these challenges effectively.

A. Problem Description and Solution Objectives

UAV safety assessment poses a multifaceted systems engineering challenge, as it involves complex interactions among hardware failures, control software anomalies, environmental uncertainty, and mission dynamics. Existing methods often suffer from two key limitations: (i) insufficient coverage of failure modes and environmental disturbances, and (ii) poor consistency between real-time monitoring results and offline safety conclusions. To address these issues, we propose a unified UAV safety assessment framework (see Fig. 3) that integrates both online and offline assessment modules into a closed-loop architecture. This framework simultaneously supports real-time safety monitoring and lifecycle-oriented post-analysis, ensuring consistency and traceability throughout the UAV operational process.

As shown in Fig. 3, the proposed framework consists of an online and an offline safety assessment module, forming a closed-loop architecture. The online module operates during flight simulation or real missions, offering real-time risk monitoring and feedback. It includes: i) component-level profust safety assessment (Block ①), which collects data from key components (e.g., motors, IMUs) and detects potential faults or degradations using a profust safety assessment approach, and ii) failure severity level assessment (Block ②), which quantifies the impact of anomalies as severity levels, providing inputs for safety-aware decision-making. The offline module performs post-mission analysis based on data gathered by the online module, enabling system-level safety evaluation. It includes: i) equivalent level of safety assessment (Block ③), which uses contextual mission factors (e.g., population density) to compute the required reliability (e.g., MTBF), verifying whether the system meets airworthiness risk thresholds, and ii) system-level FMEA (Block ④), which generates structured FMEA reports using simulation data and safety results to identify high-risk failure modes and inform mitigation strategies.

From a systems perspective, the safety assessment problem can be abstractly modeled as a closed-loop decision system

$$\mathcal{S} = \langle \mathcal{U}, \mathcal{E}, \mathcal{F}, \mathcal{O}, \mathcal{R} \rangle, \tag{1}$$

where $\mathcal{U}$ denotes UAV states and internal control variables (e.g., posture, actuator status), $\mathcal{E}$ represents external environmental conditions (e.g., wind, terrain, population density), $\mathcal{F}$ refers to potential fault modes (e.g., actuator degradation, sensor failure), $\mathcal{O}$ includes multi-dimensional safety outputs (ELOS, risk level, FMEA table, etc.), $\mathcal{R}$ captures real-time constraints and safety thresholds.

Based on this formulation, our framework is designed to fulfill two core objectives:

- *Objective 1 (real-time safety quantification)*: Develop a profust-based real-time safety metric to continuously estimate the system's safety degree during flight.
- *Objective 2 (offline-oriented risk assessment)*: Provide consistent offline assessment tools such as ELOS metrics and structured FMEA tables, based on accumulated test data. These tools help quantify long-term risk and prioritize mitigation strategies.

The detailed mathematical modeling and implementation strategies for each objective are elaborated in the following subsections.

B. Online Assessment Decomposition

1) Profust Safety: We proposed the profust safety index in [40] to quantitatively calculate the real-time safety status of UAVs. Fig. 4(a) illustrates the analytical process of profust safety definition taking $\mathbb{R}^3$ space as an example. Specifically, it is determined by the fuzzy risk space Ω and the causative event group V_Ω. Fig. 4(b) illustrates the relationship between Ω and V_Ω in $\mathbb{R}^3$ space. It encompasses four possibilities, namely,

- $V_\Omega \subset \Omega$, show that Ω has the possible conditions for the occurrence of each event in V_Ω;
- $V_\Omega \cap \Omega \neq \varnothing$, show that Ω has the possible conditions for the occurrence of partial event in V_Ω;
- $V_\Omega \cap \Omega = \varnothing$, show that Ω does not have the possible conditions for any event in V_Ω;

- $V_\Omega \supset \Omega$, show that any event in V_Ω happens, Ω will happen. At this point, V_Ω becomes the inevitable event of Ω.

Based on aforementioned definition, given a time interval $[t_0, t]$, if Ω is bounded on V_Ω, then the profust safety is defined as

$$S(\Omega, V_\Omega) = \frac{R_{V_\Omega}(t_0, t)}{I(\Omega)}, \tag{2}$$

where $S(\Omega, V_\Omega) \in [0, 1]$ represents the profust safety degree of the UAV and serves as a quantitative measure of the system's safety state. A value of $S(\Omega, V_\Omega)$ closer to 1 indicates a higher level of safety. $R_{V_\Omega}(t_0, t)$ denotes the probability of causative events occurring in the Ω, taking into account the transitions of safety states. $I(\Omega)$ maps a qualitative risk state to a quantitative fuzzy risk space within the range of $[0, 1]$. Further,

$$R_{V_\Omega}(t_0, t) = \sum_{i=1}^{n} \omega_i \mu_S(S_{x_i,k}) \cdot (1 - \mu_{T_{SF}}(m_{kj})), \tag{3}$$

where $\mu_S(\cdot)$ is the membership function for the fuzzy system state, $\mu_{T_{SF}}(m_{kj})$ denotes the membership function for the state transition, ω_i is the weight coefficient for each state variable, with $\sum \omega_i = 1$. And

$$I(\Omega) = \sum_{i=1}^{n} \omega_i I(\Omega_i). \tag{4}$$

Definitions and details of intermediate parameters such as $S_{x_i,k}$ and m_{kj} can be found in [40]. Furthermore, the accident probability of UAVs can be defined as follows:

$$P_{\text{crash}} = \frac{1}{1 + e^{-\eta \cdot k} + 100 \times \left(\frac{1}{S(\Omega, V_\Omega)}\right)^{\frac{2}{k-1}}}, \tag{5}$$

where $P_{\text{crash}} \in [0, 1)$ is the accident probability, $S(\Omega, V_\Omega)$ is the profust safety degree of the system, and $k \in [0, 1)$ is the penalty factor. The value $\eta \in [0, +\infty)$ is the adjustment coefficient.

Remark 1: It is important to note that the proposed online risk indicator P_{crash}, which quantifies the probability of UAV failure during task execution, is fundamentally different from the traditional reliability metric t_{mtbf} (as pointed in Section III-C). While t_{mtbf} reflects long-term hardware reliability based on historical or manufacturing data, P_{crash} is computed in real-time from the system's evolving safety state $S(\Omega, V_\Omega)$, integrating environmental risk modeling and current operational conditions.

2) Failure Safety Level: After obtaining the profust safety degree, it is possible to preliminarily assess the system's safety performance and its capacity to handle risks. To further assess the system's safety and the extent of casualties, we propose a qualitative fault severity level within and offer a quantitative approach to solve it [40]. This allows for the real-time assessment of the current safety status of the UAV and the harm it may pose to third parties.

C. Offline Assessment Decomposition

1) ELOS: The ELOS [35] metric is commonly used to analyze the Equivalent Level Of Safety indicator, P_{elos}, in the event of a ground impact by a UAV. Generally, $P_{\text{elos}} < 10^{-7}$ is considered acceptable. Specifically,

$$P_{\text{elos}} = \frac{1}{t_{\text{mtbf}}} A_{\text{exp}} \rho_{\text{peo}} P_{\text{pen}} (1 - P_{\text{mit}}), \tag{6}$$

where P_{elos} is the target safety level expressed in terms of ground personnel fatality rate, unit: h^{-1}; t_{mtbf} is the average time between failures, unit: h; A_{exp} is the average area of impact when the UAV collides with the ground, unit: m^2; ρ_{peo} is the average population density at the ground impact location, unit: people/m^2; P_{pen} is the penetration factor, which is associated with the impulse and obstruction factors related to the aircraft crash landing; $P_{\text{mit}} \in [0, 1)$ is used to quantify the capabilities of various mitigation mechanisms in specific risk scenarios. This factor can be regarded as a weight coefficient to reduce the severity of failure consequences. The closer the value is to 1, the more effective the mitigation measures are, and the closer it is to 0, the less mitigation measures are taken.

Remark 2: t_{mtbf} is an important indicator for assessing system reliability and is influenced by factors such as system or product design, manufacturing processes, and quality. UAV manufacturers typically provide reliability indicators, including the average time between failures, for their manufactured components. Based on the reliability indicators provided by the manufacturer, the equivalent safety level can be calculated using Equation (6).

2) FMEA: Generally, FMEA relies on three key parameters to determine the system's risk level: 1) the likelihood of occurrence (O), 2) the severity of failure (S), and 3) the detection (D) [41]. Following the release of the fifth edition FMEA manual by AIAG & VDA [42], it became possible to finely and quantitatively categorize these three key parameters. To align with relevant standards and design requirements for UAVs, we have made the following improvements based on [43]. As illustrated in Table II, O, S, and D are divided into five distinct levels, and different levels correspond to scores ranging from 1 to 5, with higher scores indicating greater levels of risk. The RPN is calculated as RPN = O × S × D. According to [44], a system is considered to have a higher-priority risk when the RPN exceeds 30.

In conjunction with [43], this paper presents a risk matrix in Table III, which is divided based on different parameter combinations of "O" and "S". Green represents safety, yellow represents danger, and red represents a catastrophic level. Additionally, as shown in Table IV, a criterion for determining AP was proposed, aiming to determine different priority responses under various "O", "S", and "D" conditions. Finally, we provide a case study of the RflySimSA module in the experimental section (**Objective** 3 and **Objective** 4 of the experimental section) to demonstrate the implementation process of safety assessment.

Remark 3: The assignment of FMEA metrics in this study adheres to a structured and transparent methodology, in response to the current lack of unified standards and limited access to publicly available failure data for UAV systems operating in complex, low-altitude environments. Specifically, the severity, occurrence, and detection ratings presented in Tables II, III and IV are established based on a multi-faceted

TABLE II

UAV FMEA OCCURRENCE, SEVERITY, AND DETECTION STANDARD CLASSIFICATION

FMEA standard division						
Occurrence (O)		**Severity (S)**			**Detection (D)**	
Rank	**Scale**	**Rank**	**Description**	**Scale**	**Rank**	**Scale**
Never	1	Unaffected	The function is normal and the UAV performs its tasks normally without causing any losses.	1	Almost Certain	1
Low	2	Minor	Function degradation but the impact on a UAV is minor, allowing it to continue performing its original tasks with a decrease in reliability.	2	High	2
Medium	3	Medium	Resulting in the loss of mission objectives and degradation of original functions, but not complete loss of work.	3	Medium	3
High	4	Serious	Completely loses its intended functions but other systems can compensate (trigger failsafe), and the UAV can hover or make an emergency landing without causing harm to the ground.	4	Low	4
Very High	5	Catastrophic	Completely loss of functions and a UAV crash, resulting in damage to property or casualties on the ground.	5	Remote	5

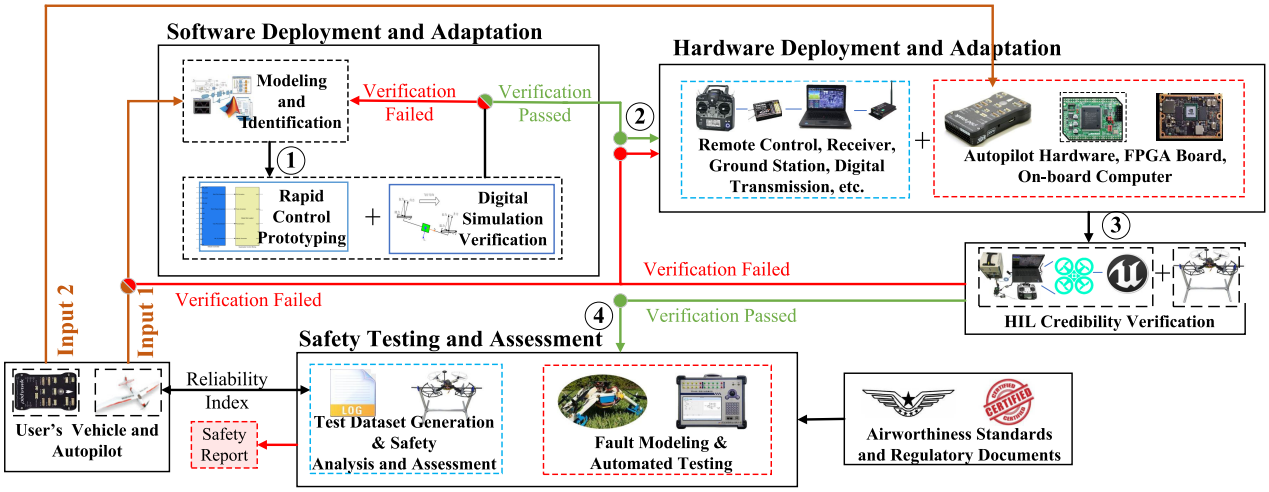


Fig. 5. Unified software adaptation and deployment, hardware adaptation and deployment, and safety testing and assessment flow charts of the RflySimSaT platform.

TABLE III

RISK RANKING MATRIX DIVISION BASED ON OCCURRENCE AND SEVERITY

O \ S	Unaffected	Minor	Medium	Serious	Catastrophic
Very high	Y	Y	R	R	R
High	Y	Y	R	R	R
Medium	G	Y	Y	R	R
Low	G	G	Y	Y	R
Never	G	G	G	Y	Y

TABLE IV

ACTION PRIORITY DIVISION

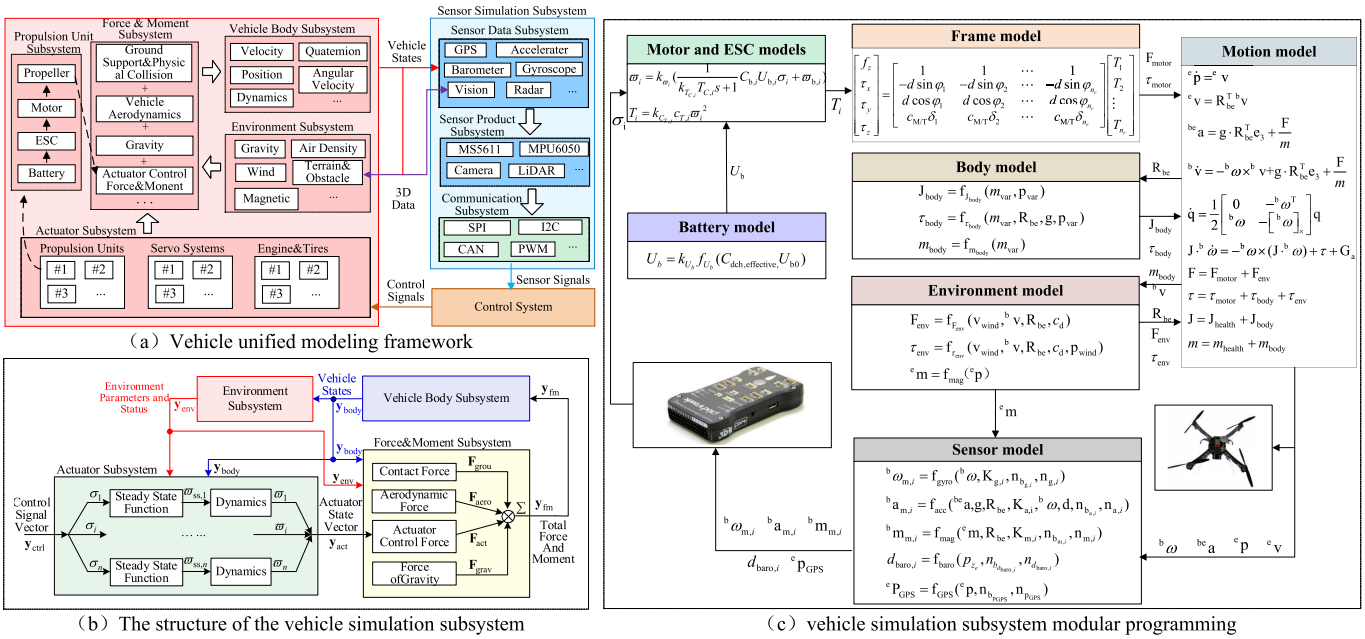
S	O	D	AP	S	O	D	AP
1	1 ∼ 5	1 ∼ 5	**L**	4	4 ∼ 5	1 ∼ 5	**H**
2	5	4 ∼ 5	**M**		3	3 ∼ 5	**H**
	5	1 ∼ 2	**L**		3	1 ∼ 2	**M**
	1 ∼ 4	1 ∼ 5	**L**		2	4 ∼ 5	**M**
3	5	4 ∼ 5	**H**		2	1 ∼ 3	**L**
	5	1 ∼ 3	**M**		1	1 ∼ 5	**L**
	4	1 ∼ 5	**M**	5	3 ∼ 5	1 ∼ 5	**H**
	3	3 ∼ 5	**M**		2	3 ∼ 5	**H**
	3	1 ∼ 2	**L**		2	1 ∼ 2	**M**
	1 ∼ 2	1 ∼ 5	**L**		1	1 ∼ 5	**L**

AP: Action priority **L:** Low **M:** Medium **H:** High

approach: i) reference to international guidelines such as ISO 31010 [45] and SAE ARP5580 [46], as well as prior FMEA practices in aerospace and robotic domains; ii) empirical results from fault injection experiments under representative mission scenarios (e.g., hovering, cruising), with evaluations guided by domain experts; and iii) publicly accessible resources to enhance reproducibility. The simulation platform and representative fault dataset used in this process can be accessed from the link in V-C.

IV. UNIFIED SAFETY TESTING AND ASSESSMENT PROCESS

Fig. 5 presents an overview of the unified process for the development, deployment, testing, and assessment of UAVs. This framework comprises three key modules: software deployment and adaptation, hardware deployment and adaptation, and safety testing and assessment. The purpose of this section is to guide users through an efficient safety testing and assessment process using RflySimSaT, allowing them to navigate seamlessly through the entire lifecycle—from design and development to testing and evaluation—using a specified frame and autopilot hardware setup.



（a）Vehicle unified modeling framework

（b）The structure of the vehicle simulation subsystem

（c）vehicle simulation subsystem modular programming

Fig. 6. Unified modeling framework for UAVs. Figure (c) presents the modular programming architecture of the UAV system simulation subsystem. In this framework, the Motor and ESC module functions as the primary power source, generating thrust T_i and torque τ_{motor}, which are transmitted through the Frame model and subsequently serve as inputs to the Body model. The Body model characterizes the vehicle's dynamic behavior through its mass m_{body} and inertia tensor J_{body}, while simultaneously accounting for external excitations originating from the Environment model—such as aerodynamic disturbances F_{env} and magnetic perturbations ${}^{e}m$. The Sensor model transforms the true body states (e.g., angular velocity ${}^{b}\omega$ and acceleration ${}^{b}a$ into observation signals contaminated by stochastic noise n and deterministic bias b, thereby forming a closed-loop perception subsystem. The Motion model then integrates all dynamic interactions to realize full six-degree-of-freedom state evolution. This process is governed by Euler's equations and quaternion-based differential kinematics, completing a physically consistent simulation loop that encompasses power generation, external disturbance, and feedback sensing. Key model parameters include: (i) structural parameters, such as the center-of-mass position p_{var} and propeller pitch d; (ii) aerodynamic and physical coefficients, including the drag coefficient c_d and thrust coefficient c_T; and (iii) sensor error parameters, such as zero bias $\mathrm{n}_{\mathrm{b}_*}$ and scale factor K_*.

A. Software Deployment and Adaptation

The software deployment and adaptation phase aims to facilitate rapid modeling when a user specifies a particular aircraft. This process includes mathematical modeling, 3D modeling, and parameter measurement, as well as fast control prototyping and digital simulation verification to ensure model accuracy and reliability. The outputs from this phase serve as high-fidelity motion model inputs for the subsequent hardware deployment and adaptation phase.

1) Modeling: To streamline the research and development cycle while maintaining model stability and reliability, we utilize a unified modeling framework [38], as illustrated in Fig. 6(a). This framework encompasses the body subsystem, sensor subsystem, and 3D environment subsystem. Based on the interconnections among vehicle simulation subsystems shown in Fig. 6(b), a set of plug-and-play templates for various aircraft configurations including multirotor, fixed-wing, and VTOL UAVs are provided.

Fig. 6(c) further details the physical modeling architecture of the proposed platform using a representative multirotor example, highlighting the hierarchical structure and interconnections among the motion dynamics, actuator, sensor, and power subsystems. In this framework, the motion model adopts the classical six-degree-of-freedom dynamic equations coupled with empirical aerodynamic coefficients, ensuring a balance between fidelity and computational efficiency. The sensor model incorporates Gaussian noise and bias drift characteristics derived from calibration data to capture typical measurement uncertainty. The battery subsystem employs an SoC-based equivalent circuit model that estimates the discharge voltage and residual capacity under dynamic loading conditions. It is noteworthy that the actuator and battery fault models are currently simplified to maintain generality and computational stability.

2) Rapid Prototyping and Simulation Verification: We offer a generic PID position and attitude controller that allows users to rapidly deploy controllers by swapping in their specific component prototypes. Sensor data from the vehicle model or state estimation information—such as attitude angles, angular rates, position, and velocity—are transmitted to the controller. The controller then sends control inputs for each actuator back to the model, creating a SIL closed-loop system. This setup enables users to efficiently prototype and validate the effectiveness of their model or system design, with the capability to specify input parameters and monitor control and model performance within the simulation software.

B. Hardware Deployment and Adaptation

The objective of hardware deployment and adaptation is to facilitate the rapid integration of a specified autopilot system, utilizing the high-fidelity models developed during the software deployment phase to satisfy the requirements for HIL simulation. This phase also involves verifying model credibility to ensure accuracy and reliability. Ultimately, hardware deployment and adaptation provide validated aircraft

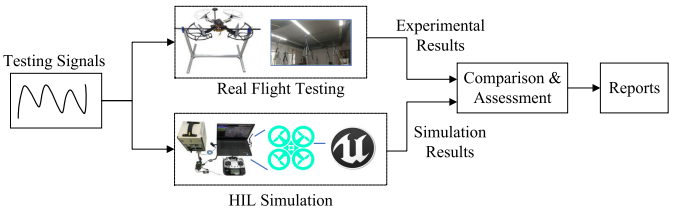


Fig. 7. HIL credibility verification method.

models and autopilot systems essential for safety testing and assessment.

1) Hardware Deployment and Adaptation: The process of platform hardware deployment and adaptation is outlined in Fig. 5. Users begin by importing the UAV model into the motion simulator as a dynamic link library, leveraging automatic code generation techniques. The controller algorithm is then downloaded onto the designated autopilot hardware. Physical USB connections replace the virtual signal lines used in SIL simulations, enabling data exchange between the simulation model and the controller. Sensor data—such as information from accelerometers, barometers, and magnetometers—is transmitted to the autopilot via USB. This requires manual modifications to the source code to disable sensor drivers and allow the reception of simulated sensor data. For FPGA-based platforms, users must provide the necessary sensor chip interfaces and protocols for integration. The autopilot processes and estimates the incoming sensor data, transmitting the estimated state information to the controller via the internal uORB message bus. The controller, in turn, sends PWM control commands for each motor back to the motion simulator through USB, creating a HIL closed-loop simulation. Additionally, the platform features rich visual interfaces and system integration—utilizing the Ubuntu operating system and the ROS control interface—enabling a seamless transition and comparative verification from simulation to real flight using high-performance embedded boards like the NVIDIA® Jetson Xavier™ NX.

2) HIL Credibility Verification: As illustrated in Fig. 7, the theoretical approach for HIL credibility assessment involves feeding identical signals to the HIL simulation system and the real system to compare their result discrepancies. In our previous work [39], we introduced a method for normalizing scales and assessment criteria to compare the disparities between the simulation system and the real system. The formula for this normalization is provided below:

$$\begin{aligned} \varepsilon_{\mathrm{p}} &= K_{\mathrm{p}} \cdot |p_{\mathrm{e}}| \\ \eta_{\mathrm{p}} &= f_{\mathrm{norm}}(e_{\mathrm{p}}, \varepsilon_{\mathrm{p}}) = \frac{K_{\mathrm{e}} \cdot \varepsilon_{\mathrm{p}}}{\sqrt{K_{\mathrm{e}}^2 \cdot {\varepsilon_{\mathrm{p}}}^2 + {e_{\mathrm{p}}}^2}}, \end{aligned} \tag{7}$$

where ε_{P} is the error threshold, p_{e} is the performance parameters of the real system, K_{p} is the percentage coefficient, $\eta_{\mathrm{p}} \in (0, 1]$ is the normalized performance credibility (%), K_{e} is the mapping coefficient, and e_{p} represents the error. When the condition η_{p} falls within the acceptable threshold, it is considered that the simulation system's trustworthiness is within an acceptable range.

C. Safety Testing and Assessment

After successful HIL reliability verification, which demonstrates the fidelity of the vehicle model to the real system and the reliability of hardware adaptation, the system is considered to have high simulation credibility. Users can then conduct a large number of nominal and off-nominal scenario tests using the HIL method, and validate their risk levels in different scenarios based on the unified safety assessment framework. This not only reduces testing costs but also identifies safety vulnerabilities in UAVs, thereby enhancing their safety.

V. Experiment and Verification

In this section, we will design experiments at multiple levels and from various perspectives, guided by the Unified Safety Testing and Assessment Process outlined in Section IV. Our aim is to thoroughly validate model reliability, data credibility, and platform scalability at every stage, from initial model development to real-flight safety testing. The specific objectives are as follows:

i) **Objective 1: Rapid Prototype Development.** Rapidly develop high-precision models based on the provided user information regarding software and hardware, ensuring the reliability of the models.
ii) **Objective 2: Data Credibility Verification.** Data credibility directly reflects model reliability. Therefore, we will compare results from different stages of SIL and HIL simulations against real-world data to verify their trustworthiness.
iii) **Objective 3: Failure Sim2Real.** This objective focuses on assessing differences in fault characteristics between simulated and real-world scenarios. The feedback from these tests will help bridge the gaps in simulation accuracy, making it more representative of real-world conditions.
iv) **Objective 4: Safety Testing and Assessment.** We will explore various failure modes and conduct effectiveness analyses. Based on the results, we will assess safety levels under different failure modes and establish minimum safety standards.

A. Experimental Configuration

1) Hardware: To alleviate user burden, we opted not to employ an FPGA-based HIL simulation platform. The hardware configuration consists of a simulation computer, a high-performance workstation with a clock frequency of 2.9GHz. The autopilot hardware utilizes the open-source PixHawk, while the frame is provided by Beijing Zhuoyi Corporation's FS-X200 and FS-X450 models. Information exchange and control are achieved through a USB data connection between the simulation computer and the hardware components.

2) Scenario Construction and Parameter Specification: The design of safety testing scenarios and the specification of key parameters in our platform are grounded in regulatory guidelines and domain expertise. Drawing upon established airworthiness standards and national UAV safety regulations, we extract formalized safety assurance requirements—such as

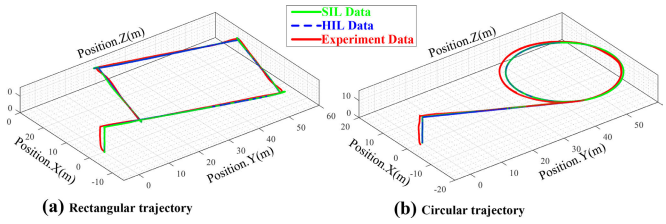


Fig. 8. Trajectory tracking errors under two different task scenarios (rectangular and circular trajectories) for SIL simulation, HIL simulation, and real flight experiments.

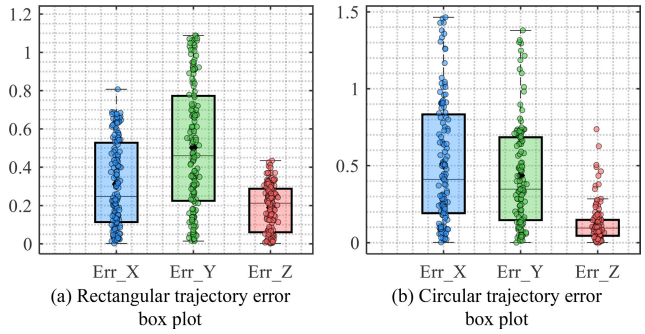


Fig. 9. Error distribution box plot of rectangular trajectory and circular trajectory, where the error is the three-axis trajectory error between HIL and real experiment.

operational boundary conditions, fault tolerance thresholds, and risk exposure metrics—which serve as the foundation for scenario construction. These requirements are systematically translated into executable test cases that emulate mission-critical operations under varying environmental and fault conditions. Scenario deployment is further guided by the principle of representativeness, ensuring that both nominal and adverse conditions reflect the operational complexity of real-world UAV missions. Concurrently, the definition and configuration of model, fault, and environment parameters are informed by a synthesis of regulatory norms and expert knowledge, balancing realism, safety coverage, and tractability. This integrated methodology ensures that all scenarios and parameters within the platform are not only standardized and reproducible, but also aligned with safety validation goals recognized by industry.

B. Implementation

1) Rapid Prototype Development: To achieve **Objective 1**, we rapidly constructed the motion model and 3D model for the FS-X series quadcopter using the model templates provided in Section IV. As illustrated in Fig. 6(c), the motion model comprises modules for the power unit, environmental, sensors, structure, body, and motion. The 3D model receives attitude information computed by the motion module for visualization, while the autopilot hardware processes data from the sensor module for attitude and control calculations. The generated throttle signal is then transmitted to the power unit, establishing a closed-loop simulation. For model details, please visit: **https://github.com/RflySim/RflySimSafe**.

2) Data Credibility Verification: To achieve **Objective 2**, trajectory tracking experiments were designed in this section to validate the reliability of the data. The results, illustrated in Fig. 8 and Fig. 9, compare the performance of tracking rectangular and circular trajectories across different experimental stages from the controller's perspective. The figure demonstrates that trajectory tracking errors at each stage remain within an acceptable range. We further evaluated the reliability of the experiments using Equation (7). The average tracking error was selected as the error metric e_{p}, and the final results are presented in Table V. The table presents credibility scores (η_{p}, %) for key performance indicators (mean error $e_{\mathrm{p}}^{\mathrm{Me}}$ and standard deviation error $e_{\mathrm{p}}^{\mathrm{Se}}$) between HIL simulation and real flight under different failure modes. The "Pass?" column indicates whether the platform meets the predefined credibility threshold (60%) for each situation. The table reveals

TABLE V

DATA CREDIBILITY VERIFICATION RESULTS BASED ON ERROR DISTRIBUTION ANALYSIS. $e_{\mathrm{p}}^{\mathrm{ME}}$ AND $e_{\mathrm{p}}^{\mathrm{SE}}$ REPRESENT THE MEAN ERROR AND STANDARD DEVIATION ERROR RESPECTIVELY

Target	Error e_{p}	Threshold ε_{p}	Credibility $\eta_{\mathrm{p}}(\%)$	Passed?
Fig. 8(a)	$e_{\mathrm{p}}^{\mathrm{Me}}: 3.3 \times 10^{-1}$	6×10^{-1}	80.3%	✓
Fig. 8(b)	$e_{\mathrm{p}}^{\mathrm{Me}}: 3.5 \times 10^{-1}$	6×10^{-1}	78.4%	✓
Fig. 8(a)	$e_{\mathrm{p}}^{\mathrm{Se}}: 2.2 \times 10^{-1}$	6×10^{-1}	89.6%	✓
Fig. 8(b)	$e_{\mathrm{p}}^{\mathrm{Se}}: 2.8 \times 10^{-1}$	6×10^{-1}	84.4%	✓

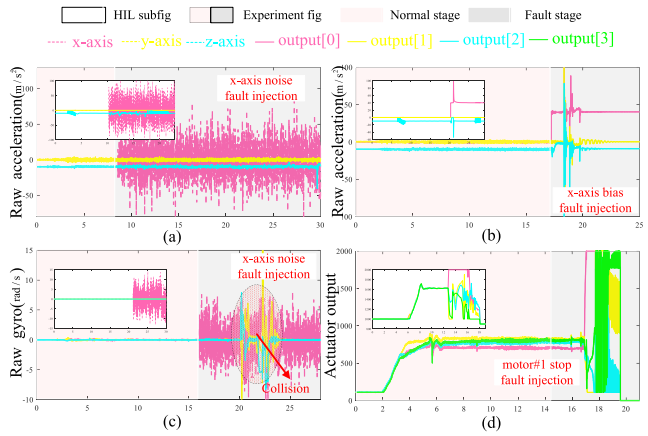


Fig. 10. Comparative data analysis plots of HIL simulation and real flight safety testing under different failure modes, including accelerometer noise, accelerometer bias, gyroscope noise, and motor failure. Each figure comprises two parts: data plots for HIL (subplot) and real-flight scenarios. Figure (a) illustrates the outcomes after injecting accelerometer x-axis Gaussian noise faults, corresponding to parts (a) and (b) of Fig. 11. Figure (b) illustrates the results following the injection of accelerometer x-axis bias faults, corresponding to parts (c) and (d) of Fig. 11. Figure (c) illustrates the outcomes after injecting gyroscope x-axis Gaussian noise, corresponding to parts (e) and (f) of Fig. 11. Figure (d) illustrates the results after injecting motor#1 stop faults, corresponding to parts (g) and (h) of Fig. 11.

that the experiment achieved a high level of reliability, closely matching real-world conditions.

3) Failure Sim2Real: To achieve **Objective 3**, we designed four comparative experiments to evaluate the safety characteristics of simulation versus real-flight under various failure modes. The results are presented in Fig. 11 and Fig. 10. Fig. 11 shows keyframe analysis results for different failure modes, while Fig. 10 displays the data comparisons. Notably,

Fig. 11. Keyframe visualization comparisons of HIL simulation and real flight under identical task conditions for four failure modes: accelerometer noise, accelerometer bias, gyroscope noise, and motor failure. Figures (a) and (b) respectively depict the results after injecting accelerometer Gaussian noise faults (mean: 0 m/s^2, variance: 50 m/s^2) in HIL and real-flight scenarios. Figures (c) and (d) depict the results after injecting accelerometer x-axis bias faults (bias magnitude: 40 m/s^2) in HIL and real-flight scenarios. Figures (e) and (f) depict the outcomes following the injection of gyroscope Gaussian noise faults (mean: 0 rad/s, variance: 5 rad/s) in HIL and real-flight scenarios. Figures (g) and (h) depict the results after injecting motor#1 stop faults in HIL and real-flight scenarios. The yellow segments represent critical frames in the aircraft's motion, while the circular regions denote the flight states following the injection of faults.

TABLE VI
EXPECTED SAFETY STANDARDS UNDER DIFFERENT FAILURE MODES AND POPULATION DENSITIES

P_{mit} / ρ_{peo}	t_{mtbf1}			t_{mtbf2}			t_{mtbf3}			t_{mtbf4}		
	0.8	0.6	0.4	0.8	0.6	0.4	0.8	0.6	0.4	0.8	0.6	0.4
2753×10^{-6}	152	305	457	166	333	500	166	333	500	173	346	519
11299×10^{-6}	625	1250	1875	683	1368	2051	684	1368	2053	710	1421	2132
15286×10^{-6}	845	1691	2537	925	1850	2775	925	1851	2777	961	1923	2884
918×10^{-6}	50	102	152	55	111	167	55	111	167	57	116	173
4385×10^{-6}	242	485	728	265	531	796	265	531	797	275	552	827
Parameters & Values	$P_{\text{crash1}} = 87\%$ $P_{\text{crash2}} = 95.18\%$ $P_{\text{crash3}} = 95.23\%$ $P_{\text{crash4}} = 98.9\%$ $k_1 = 0.6$ $k_2 = 0.6$ $k_3 = 0.6$ $k_4 = 0.9$ $P_{\text{elos}} = 10^{-7}$ $P_{\text{pen}} = 0.05$ $A_{\text{exp}} = 0.64$											

both simulation and real-flight demonstrate similar fault flight states across the various failure modes. The data comparison confirms that the discrepancies between simulation and real-flight remain within an acceptable range, underscoring the platform's credibility. These findings provide reliable data support for subsequent safety testing and assessment.

Additionally, we conducted a safety assessment of these four experiments using the online assessment method from the RflySimSA module outlined in Section III. The assessment results, shown in Table VII, indicate that the failure safety levels for all experiments were categorized as catastrophic using Equation (2). Except for the RT001 experiment, all other failure scenarios resulted in aircraft crashes, which is unacceptable. Furthermore, we analyzed the four failure modes using Equations (5) and (6) and assessed the expected minimum safety indicators across varying population densities. The densities ρ_{peo} listed in Table VI represent those of different districts in Changsha City: Yuelu, Tianxin, Furong, Wangcheng, and Kaifu. The minimum safety indicators for different faults—accelerometer Gaussian noise, accelerometer bias, gyroscope Gaussian noise, and motor failure—are all set at t_{mtbf1}, t_{mtbf2}, t_{mtbf3}, and t_{mtbf4}, respectively. The results in

TABLE VII
FAULT TEST CASES AND SAFETY ASSESSMENT RESULTS OF FS-X200

CaseID	FaultID	Case Description	Control Sequence	Result
RT001	**1E298**	•**Frame:** FS-X200 •**Fault Mode:** Accelerometer noise interference •**Data Required:** Vehicle angular velocity	•**0s:** Armed! •**1s:** Take off and hover at a height of 1 meters above the ground. •**8s:** Send fault injection command •**30s:** Exit test!	•**Is Fall:** No •**Fall Time:** - •**Fall Vel:** - •**Flight Status After Fault Injection:** Shake sharply •**Failure Safety Score:** 0.251 •**Failure Safety Level:** Catastrophic
RT002	**1E297**	•**Frame:** FS-X200 •**Fault Mode:** Accelerometer bias •**Data Required:** Vehicle angular velocity	•**0s:** Armed! •**1s:** Take off and hover at a height of 1 meters above the ground. •**17s:** Send fault injection command •**25s:** Exit test!	•**Is Fall:** Yes •**Fall Time:** 22s •**Fall Vel:** 1.875m/s •**Flight Status After Fault Injection:** Lost control and crashed rapidly •**Failure Safety Score:** 0.097 •**Failure Safety Level:** Catastrophic
RT003	**1E299**	•**Frame:** FS-X200 •**Fault Mode:** Gyroscope noise interference •**Data Required:** Vehicle angular velocity	•**0s:** Armed! •**1s:** Take off and hover at a height of 1 meters above the ground. •**16s:** Send fault injection command •**28s:** Exit test!	•**Is Fall:** Yes (Collision, manual kill) •**Fall Time:** 28s •**Fall Vel:** 4.286m/s •**Flight Status After Fault Injection:** Lost control and collision •**Failure Safety Score:** 0.079 •**Failure Safety Level:** Catastrophic
RT004	**1E23A**	•**Frame:** FS-X200 •**Fault Mode:** motor#1 stop •**Data Required:** Vehicle angular velocity	•**0s:** Armed! •**1s:** Take off and hover at a height of 1 meters above the ground. •**17s:** Send fault injection command •**21s:** Exit test!	•**Is Fall:** Yes •**Fall Time:** 18s •**Fall Vel:** 2.727m/s •**Flight Status After Fault Injection:** Lost control and crash •**Failure Safety Score:** 0.093 •**Failure Safety Level:** Catastrophic

TABLE VIII
FS-X450 QUADCOPTERS FAILURE MODE AND EFFECT ANALYSIS

Components	Failure Modes	Effects Analysis	O	S	D	RL	RPN	AP
Motor	Circuit failure	Motor stop and crash	3	5	4	R	60	H
	Loss of efficiency	Low control efficiency	3	3	4	Y	36	M
	Bearing failure	Shaky and unstable flight	3	4	3	R	36	H
Battery	Low voltage	Low-pressure alarm	3	2	3	Y	18	L
	Battery failure	Lose power and crash	2	5	5	R	50	H
Environment	Gust interference	Attitude instability	4	3	1	R	12	M
	Atmospheric turbulence	Completely out of control	2	5	1	R	10	M
IMU	Large noise	Low flight performance	4	4	3	R	48	H
	Large bias	Low flight performance	4	4	3	R	48	H
	Data loss	Out of control and crash	2	5	3	R	30	H
	Data error	Out of control and crash	2	5	3	R	30	H
GPS	Satellite search failed	Positioning failed	2	5	2	R	20	M
	Signal Loss	Off track, mission failed	3	5	4	R	60	H
	Electromagnetic Interference	Large positioning error	4	4	4	R	64	H
	Data error	Off track, out of control	2	5	5	R	50	H

The classification of OSD, RL, and AP are determined based on Table II, III, and IV,respectively.

Table VI demonstrate that higher population densities correlate with increased expected safety indicators. In addition, from the perspective of the shielding factor P_{mit}, the higher its value (i.e., the mitigation measure), the lower the mean time between failures t_{mtbf}. These safety analysis outcomes are essential for guiding UAV manufacturers in the design of critical components, suggesting that adherence to safety indicators during design ensures the aircraft's safety level remains within acceptable standards.

4) Safety Testing and Assessment: Upon achieving Objectives 1 through 3, it is demonstrated that the platform is fully equipped for a seamless transition from simulation to real-flight testing, attaining a high level of credibility. Building upon this foundation, extensive safety testing experiments were conducted, combining simulation and real-flight scenarios. The safety assessment of the test results, informed by expert insights, is presented in Table VIII. Here we assume that failures are triggered according to the probability of MTBF.

As shown in the table, the experimental outcomes indicate significant safety issues for UAVs under most failure modes, encompassing performance degradation, system failures, and other critical concerns. Assessing RL and AP across different failure modes aids in implementing timely preventive and corrective measures to mitigate the risks associated with fault occurrences.

C. Source Code, Videos, and Examples

For source code, please visit **https://github.com/RflySim/RflySimSafe**. For videos, please visit **https://rflysim.com/doc/en/7/Intro.html**. For development cases, please visit **https://rflysim.com/doc/en/7/CoreExp.html**.

VI. Conclusion and Future Work

In response to the challenges associated with UAV safety assessment, this paper introduces RflySimSaT, a high-fidelity, fully functional, and scalable safety assessment platform. Utilizing a framework and model-based approach, RflySimSaT aims to significantly enhance UAV safety levels. Extensive experiments have validated the reliability and versatility of RflySimSaT, and detailed user instructions and tutorials are available for various application templates. The platform integrates numerous interfaces for algorithm deployment, custom fault injection, and data validation, maximizing user satisfaction and supporting the advancement of the UAV industry.

The modular RflySimSaT platform allows easy adaptation to different UAV types by replacing models within the physical layer, without major changes to other components. Our vision for releasing RflySimSaT is to establish it as a credible and authoritative platform for safety testing, verification, and assessment of UAV systems. We aim to enhance the platform to provide high-fidelity safety assessments, reducing reliance on costly and constrained physical flight tests. In future work, we plan to expand the fault coverage of RflySimSaT to encompass a broader range of critical and realistic failure modes, such as propeller blade fracture, communication and software failures, structural damage under extreme load conditions, and complex inter-system faults. These efforts will further enhance the platform's capability to support comprehensive fault modeling and safety validation, advancing its applicability across both academic research and industrial certification scenarios.

References

[1] S. Boopathi, "Advancements in machine learning and AI for intelligent systems in drone applications for smart city developments," in *Futuristic E-Governance Security With Deep Learning Applications*. Hershey, PA, USA: IGI Global, 2024, pp. 15–45.

[2] S. O. Ajakwe, D.-S. Kim, and J.-M. Lee, "Drone transportation system: Systematic review of security dynamics for smart mobility," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14462–14482, Aug. 2023.

[3] D. Wanner, H. A. Hashim, S. Srivastava, and A. Steinhauer, "UAV avionics safety, certification, accidents, redundancy, integrity, and reliability: A comprehensive review and future trends," *Drone Syst. Appl.*, vol. 12, pp. 1–23, Jan. 2024.

[4] *Department of Introduction (DODI) 6055.07.mishap Notification, Investigation, Reporting, and Record Keeping*, U.S.Department of Defense, Washington, DC, USA, Jun. 2011.

[5] H. Liang, L. Chen, X. Yang, and T. Huang, "Finite-time fault-tolerant consensus of UAVs: A switching event-triggered fuzzy control scheme," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 6554–6568, 2025, doi: 10.1109/TASE.2024.3447277.

[6] Z. Ma, J. You, Y. Zhang, Y. Cheng, and J. Shao, "Reinforcement learning-based dynamic coverage control of multi-rotor UAVs with safety priority," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 17474–17485, 2025, doi: 10.1109/TASE.2024.3420094.

[7] A. Ghaffari, "Analytical design and experimental verification of geofencing control for aerial applications," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 1106–1117, Apr. 2021.

[8] S. D. Young et al., "Flight testing in-time safety assurance technologies for UAS operations," in *Proc. AIAA AVIATION Forum*, Jun. 2022, pp. 2022–3458.

[9] C. M. Belcastro, D. H. Klyde, M. J. Logan, R. L. Newman, and J. V. Foster, "Experimental flight testing for assessing the safety of unmanned aircraft system safety-critical operations," in *Proc. 17th AIAA Aviation Technol., Integr., Operations Conf.*, Jun. 2017, pp. 2017–3274.

[10] F. Nex et al., "UAV in the advent of the twenties: Where we stand and what is next," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 215–242, Feb. 2022.

[11] E. Johnson and S. Mishra, "Flight simulation for the development of an experimental UAV," in *Proc. AIAA Model. Simul. Technol. Conf. Exhib.*, Aug. 2002, pp. 2002–4975.

[12] S. Khatiri, S. Panichella, and P. Tonella, "Simulation-based test case generation for unmanned aerial vehicles in the neighborhood of real flights," in *Proc. IEEE Conf. Softw. Test., Verification Validation (ICST)*, Apr. 2023, pp. 281–292.

[13] J. W. Xiang, K. Zi, H. Y. Shao, H. D. Li, X. Dong, and D. C. Li, "A review of key technologies for long-endurance unmanned aerial vehicle," *Harbin Gongye Daxue Xuebao/J. Harbin Inst. Technol.*, vol. 52, no. 6, pp. 57–77, 2020.

[14] B. Yang et al., "Distributed cooperative framework for multiple UAVs safety: A capability-triggered mechanism," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 8303–8316, 2025, doi: 10.1109/TASE.2024.3483932.

[15] B. Yang, E. Yang, L. Yu, and C. Niu, "Adaptive extended Kalman filter-based fusion approach for high-precision UAV positioning in extremely confined environments," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 1, pp. 543–554, Feb. 2023.

[16] U. B. Mehta, D. R. Eklund, V. J. Romero, J. A. Pearce, and N. S. Keim, "Simulation credibility: Advances in verification, validation, and uncertainty quantification," NASA, USA, Tech. Rep. NASA/TP-2016-219422, Nov. 2016.

[17] O. Lisagor, T. Kelly, and R. Niu, "Model-based safety assessment: Review of the discipline and its challenges," in *Proc. 9th Int. Conf. Rel., Maintainability Saf.*, Jun. 2011, pp. 625–632.

[18] G. Gil, M. Yoo, and J. S. Park, "A study on the development of airworthiness standards for VTOL UAS," *J. Aerosp. Syst. Eng.*, vol. 14, no. 1, pp. 44–53, 2020.

[19] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, "A survey of open-source UAV flight controllers and flight simulators," *Microprocessors Microsystems*, vol. 61, pp. 11–20, Sep. 2018.

[20] A. Kadian et al., "Sim2real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6670–6677, Apr. 2020.

[21] F. Dettù, S. Formentin, and S. M. Savaresi, "The twin-in-the-loop approach for vehicle dynamics control," *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 2, pp. 1217–1228, Apr. 2024.

[22] J. V. Foster and D. Hartman, "High-fidelity multi-rotor unmanned aircraft system (UAS) simulation development for trajectory prediction under off-nominal flight dynamics," in *Proc. 17th AIAA Aviation Technol., Integr., Operations Conf.*, Jun. 2017, p. 3271.

[23] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, "State of the art of virtual reality technology," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–19.

[24] S. S. Noureen, V. Roy, and S. B. Bayne, "An overall study of a real-time simulator and application of RT-LAB using MATLAB simpowersystems," in *Proc. IEEE Green Energy Smart Syst. Conf. (IGESSC)*, Nov. 2017, pp. 1–5.

[25] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Service Robot., Results 11th Int. Conf.*, 2018, pp. 621–635.

[26] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6941–6948.

[27] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct. 2004, pp. 2149–2154.

[28] O. Michel, "Webots: Symbiosis between virtual and real mobile robots," in *Proc. 1st Int. Conf. Virtual Worlds*. Cham, Switzerland: Springer, 1998, pp. 254–263.

[29] I. Lugo-Cárdenas, G. Flores, and R. Lozano, "The MAV3DSim: A simulation platform for research, education and validation of UAV controllers," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 713–717, 2014.

[30] F. Furrer, M. Burri, M. Achtelik, and Roland Siegwart, "Rotors—A modular gazebo MAV simulator framework," *Robot Operating Syst.*, vol. 1, pp. 595–625, Jul. 2016.

[31] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "XTDrone: A customizable multi-rotor UAVs simulation platform," in *Proc. 4th Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2020, pp. 55–61.

[32] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Proc. Conf. Robot Learn.*, 2020, pp. 1147–1157.

[33] R. Weibel and R. J. Hansman, "Safety considerations for operation of different classes of UAVs in the NAS," in *Proc. AIAA 4th Aviation Technol., Integr. Operations (ATIO) Forum*, Sep. 2004, pp. 1–11.

[34] M. H. Che Man, H. Haoliang, and K. H. Low, "Crash area estimation for ground risk of small unmanned aerial vehicles due to propulsion system failures," in *Proc. AIAA SCITECH Forum*, Jan. 2022, pp. 1–15.
[35] D. W. King, A. Bertapelle, and C. Moses, "UAV failure rate criteria for equivalent level of safety," in *Proc. Int. Helicopter Saf. Symp.*, 2005, pp. 9–18.
[36] Z. Zhao, Q. Quan, and K.-Y. Cai, "A profust reliability based approach to prognostics and health management," *IEEE Trans. Rel.*, vol. 63, no. 1, pp. 26–41, Mar. 2014.
[37] M. Wei, L. Zheng, Y. Wu, H. Liu, and H. Cheng, "Safe learning-based control for multiple UAVs under uncertain disturbances," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 4, pp. 7349–7362, Oct. 2024.
[38] X. Dai, C. Ke, Q. Quan, and K.-Y. Cai, "RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations," *Aerosp. Sci. Technol.*, vol. 114, Jul. 2021, Art. no. 106727.
[39] X. Dai, C. Ke, Q. Quan, and K.-Y. Cai, "Simulation credibility assessment methodology with FPGA-based hardware-in-the-loop platform," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3282–3291, Apr. 2021.
[40] X. Dai, J. Tu, and Q. Quan, "Safety assessment approach to UAVs based on profust safety index and HIL simulation," *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 5, pp. 3336–3347, Oct. 2024.
[41] P. Freeman and G. J. Balas, "Actuation failure modes and effects analysis for a small UAV," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 1292–1297.
[42] D. Plinta, E. Golinska, and L. Dulina, "Practical application of the new approach to FMEA method according to AIAG and VDA reference manual," *Commun.-Sci. Lett. Univ. Zilina*, vol. 23, no. 4, pp. B325–B335, Oct. 2021.
[43] N. Osmic, A. Tahirbegovic, A. Tahirovic, and S. Bogdan, "Failure mode and effects analysis for large scale multirotor unmanned aerial vehicle controlled by moving mass system," in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, Oct. 2018, pp. 1–8.
[44] T. Schneider, S. Bouabdallah, and G. D. K. Rudin, "Fault-tolerant multirotor systems," M.Sc. thesis, Dept. Mech. Eng., ETH Zurich, Zürich, Switzerland, 2011.
[45] IEC Central Office., Standard ISO/IEC 30100, 2016. [Online]. Available: https://www.isa.org.jm/wp-content/uploads/2022/12/isoiec31010.pdf
[46] *Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications ARP5580*, SAE International, Warrendale, PA, USA, 2020, doi: 10.4271/ARP5580.

Xunhua Dai received the B.S., M.S., and Ph.D. degrees in control science and engineering from Beihang University, Beijing, China, in 2013, 2016, and 2020, respectively. Since 2020, he has been an Associate Professor in computer science and engineering with Central South University, where he is currently with the School of Computer Science and Engineering. His main research interests include reliable intelligent control, safety assessment, and design optimization of unmanned aerial robotics.

Jinhu Tu received the B.S. degree from the School of Software Engineering, Jiangxi University of Science and Technology, in 2021, and the M.S. degree from the School of Computer Science, Central South University, in 2024. He is currently pursuing the Ph.D. degree with the School of Automation, Central South University. His main research interests include UAV digital twins, health assessment and diagnosis, and self-safety enhancement.

Yong Chen received the Ph.D. degree in mechanical engineering from The University of Hong Kong, Hong Kong, in 2016. From 2016 to 2018, he was a Research Assistant with the Department of Mechanical Engineering, The University of Hong Kong. Since 2018, he has been a Lecture with the College of Automation, Central South University, Changsha, China. His current research interests include reachable set estimation for dynamic systems, distributed optimization of multiagent systems, and the applications in aerial robotics.

Quan Quan (Senior Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from Beihang University, Beijing, China, in 2004, and 2010, respectively. Since 2022, he has been a Professor in control science and engineering with Beihang University, where he is currently with the School of Automation Science and Electrical Engineering. His research interests include reliable fight control, swarm intelligence, vision-based navigation, and health assessment.