

# RflySim ToolChain: a rapid development and validation toolchain for intelligent unmanned swarm systems

DAI Xunhua<sup>1</sup>, TU Jinhui<sup>2</sup>, and QUAN Quan<sup>3,\*</sup>

1. School of Computer Science and Engineering, Central South University, Changsha 410083, China;

2. School of Automation, Control Science and Engineering, Central South University, Changsha 410083, China;

3. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

**Abstract:** Developing intelligent unmanned swarm systems (IUSSs) is a highly intricate process. Although current simulators and toolchains have made a notable contribution to the development of algorithms for IUSSs, they tend to concentrate on isolated technical elements and are deficient in addressing the full spectrum of critical technologies and development needs in a systematic and integrative manner. Furthermore, the current suite of tools has not adequately addressed the challenge of bridging the gap between simulation and real-world deployment of algorithms. Therefore, a comprehensive solution must be developed that encompasses the entire IUSS development lifecycle. In this study, we present the RflySim ToolChain, which has been developed with the specific aim of facilitating the rapid development and validation of IUSSs. The RflySim ToolChain employs a model-based design (MBD) approach, integrating a modeling and simulation module, a lower reliable control module, and an upper swarm decision-making module. This comprehensive integration encompasses the entire process, from modeling and simulation to testing and deployment, thereby enabling users to rapidly construct and validate IUSSs. The principal advantages of the RflySim ToolChain are as follows: it provides a comprehensive solution that meets the full-stack development needs of IUSSs; the highly modular architecture and comprehensive software development kit (SDK) facilitate the automation of the entire IUSS development process. Furthermore, the high-fidelity model design and reliable architecture solution ensure a seamless transition from simulation to real-world deployment, which is known as the simulation to reality (Sim2Real) process. This paper presents a series of case studies that illustrate the effectiveness of the RflySim ToolChain in supporting the research and application of IUSSs.

**Keywords:** simulation, unmanned aerial vehicle (UAV), swarm, RflySim, hardware-in-the-loop (HIL), model-based design (MBD), simulation to reality (Sim2Real).

**DOI:** [10.23919/JSEE.2025.000079](https://doi.org/10.23919/JSEE.2025.000079)

---

Manuscript received August 13, 2024.

\*Corresponding author.

This work was supported by the National Natural Science Foundation of China (62406345).

## 1. Introduction

### 1.1 Background

Intelligent unmanned swarm systems (IUSSs) represent a sophisticated technological integration that combines unmanned platforms, such as unmanned vehicles and robots, with artificial intelligence to create intelligent swarms capable of autonomous or semi-autonomous task execution [1]. These systems are being deployed with increasing frequency in a variety of applications, including military reconnaissance, logistics delivery, environmental monitoring, and emergency rescue operations [2,3]. The IUSS is not only capable of performing tasks independently but also emphasizes collaborative cooperation among multiple agents, allowing them to adapt to diverse and complex environments. The unmanned systems integrated roadmap [4], as published by the U.S. Department of Defense, explicitly reveals four major development themes for future unmanned systems: interoperability, autonomy, cybersecurity, and human-machine collaboration. By integrating technologies in intelligent perception, decision-making, control, and communication, IUSSs exhibit high levels of autonomy, cooperativity, and parallelism. They are capable of executing tasks that are dirty, dull, and dangerous, thereby reducing human involvement and enhancing operational efficiency and safety [5,6].

Despite the significant potential of IUSSs, their development and testing present substantial challenges. Traditional experimental methods are limited by safety concerns, high costs, and spatial constraints [7], making them insufficient for the demands of large-scale swarm systems [8]. Although simulation methods offer some assistance [9], they remain inadequate for developing precise models, simulating intricate environmental interactions,

and effectively integrating them with real-world applications. As application scenarios become increasingly complex and diverse, the requirements for unmanned systems have expanded beyond single-system operations. There is now a critical need for highly coordinated, intelligent decision-making, and autonomously adaptive swarm operations in complex environments. This shift increases the difficulty of system design and implementation, underscoring the necessity for advanced, integrated approaches to IUSS development and testing.

Therefore, to accelerate the realization of a comprehensive system architecture and general technological framework for IUSSs, spanning the entire process from simulation to experimentation, and to shorten the cycle from laboratory research to practical application, it is crucial to achieve a high degree of automation and intelligence in the development of IUSSs.

## 1.2 Related work and motivation

The development and testing methods for IUSSs mainly encompass traditional flight testing and simulation platform testing. Traditional flight testing, performed in real-world environments, provides the most accurate data feedback [10]. However, it faces significant challenges due to the high integration of components, complex sensor types, and large numbers of unmanned units. These challenges include high costs, complicated airspace approvals, and considerable testing risks. Consequently, simulation platform testing has become the preferred approach. It enables researchers to comprehensively validate algorithms and system design in a virtual environment, reducing costs and risks while accelerating the development process [11].

### 1.2.1 Progress

In the field of IUSSs, a variety of development and testing platforms are available, ranging from basic physical simulations to complex swarm behavior simulations [8]. Classic platforms such as Gazebo [12], AirSim [13], FlightGoggles [14], and Flightmare [15] are widely used by researchers. More recently developed platforms like FastSim [16], RotorTM [17], and Aerostack2 [18] offer enhanced features and customization options. A recent systematic review by [19] analyzed a total of 43 simulators, including these well-known platforms. The study categorized the simulators based on their functional focus and applicable scenarios into seven types: universal simulators, sensor-focused, learning-focused, dynamics-focused, swarming, part of flight stacks, and flight simulators. Furthermore, the authors proposed ten criteria for

selecting simulators: physics fidelity, visual fidelity, autopilots, multiple vehicles, heterogeneity, sensors, unmanned aerial vehicle (UAV) models, simulation speed, application programming interfaces (APIs), and integration [19].

It is worth noting that in the development of IUSSs, the aforementioned categories and criteria are crucial considerations. However, existing platforms often focus on specific aspects or a limited number of factors, rather than offering a comprehensive solution that covers the entire IUSS development process. Platforms like PyFly [20], RotorPy [21], and the MATLAB UAV toolbox [22] emphasize high-precision physical dynamics simulation to support control algorithm research and development. AirSim, FlightGoggles, and FastSim primarily focus on providing detailed sensor models to aid in the testing of perception and localization algorithms. Conversely, platforms such as QuadSwarm [23], and Qplane [24] are designed to enhance simulation speed, catering to large-scale data-driven learning methods.

Gazebo emphasizes platform universality and supports multiple vehicles. FlightGoggles and AirSim excel in sensor simulation and dynamic modeling. RotorTM and the MATLAB UAV toolbox prioritize the realism of physical models and simulation speed. These platforms, while successful in their targeted areas, often lack a comprehensive approach to the entire development process of autonomous swarm systems. They are designed with a specific focus, which limits their ability to handle the complex and diverse requirements of IUSSs. However, single-focused simulators fall short of meeting the full development and testing needs of IUSSs. The existing challenges can be summarized as follows:

- (i) The focus is limited to single technological aspects, such as isolated flight control algorithm design, sensor data processing, or simple formation flight simulation. There is a lack of a comprehensive integrated solution to efficiently support the full-stack development process, from conceptual design to practical deployment.
- (ii) Development efficiency is low due to the absence of integrated tools and a systematic platform construction framework. Researchers must spend substantial time switching between different software and migrating data.
- (iii) Existing software and hardware components are disparate, lacking unified usage standards, software interfaces, and communication protocols. Additionally, relevant source code is not openly accessible, resulting in sig-

nificant difficulties in learning, mastering, and conducting secondary development.

(iv) The complexity and scale of the entire platform design toolchain make it time-consuming and labor-intensive to build from scratch.

### 1.2.2 Our previous work

In response to the development needs of IUSSs, our objective is to design a comprehensive toolchain that fully meets the criteria. The RflySim platform was initiated by the reliable flight control group (Rfly) in 2016, with the term “RflySim” first coined formally and its basic version developed in [25]. Our efforts have focused on platform design, credibility assessment, aircraft design methodology, and planning and control theory.

(i) In terms of platform design, we have developed the RflySim [26] platform. This platform employs a unified modeling framework tailored for various aircraft types, adhering to IUSS standards for multiple vehicles, sensors, and UAV models. Addressing scalability, we have implemented a field programmable gate array (FPGA)-based hardware-in-the-loop (HIL) testing method to accommodate different autopilot and sensor products, thereby fulfilling IUSS standards for autopilots and heterogeneity. This approach ensures a highly modular and adaptable simulation environment crucial for IUSS development.

(ii) In terms of simulation credibility, we propose an HIL-based assessment method [27]. This method rigorously evaluates software and hardware integrity across time-domain and frequency-domain perspectives, aligning with IUSS fidelity standards. It ensures the fidelity of physical models, sensor accuracy, and aircraft simulation, enabling seamless integration of the entire swarm system from simulation to real-world deployment.

(iii) In terms of aircraft design methodology, we have conducted extensive experiments [28]. These experiments encompass propulsion system design, motion modeling, sensor calibration, state estimation, and filtering. They provide a robust theoretical and practical basis for IUSS development, ensuring efficiency and precision throughout each phase. By adhering to standardized experimental setups and procedures, we facilitate a seamless Sim2Real transition plan, enhancing the practical viability and reliability of the system in real-world applications.

(iv) In terms of planning and control theory, we have a theory rooted in the concept of virtual tubes [29–31]. In

this theory, virtual tubes are designed in a high-dimensional space that can effectively limit the planning and control common in complex environments to a simplified and secure virtual pipeline environment. By adopting this approach, we succeed in diminishing computational complexity while markedly bolstering the safety, coordination, and efficiency of task execution for IUSSs. Furthermore, this framework bestows significant optimization and support for swarm decision-making processes.

### 1.2.3 Motivation

In light of our previous research, we have identified a crucial need for a platform that integrates a range of functionalities to provide comprehensive support for IUSS development across the simulation, algorithm design, and practical deployment phases. The platform must ensure seamless efficiency throughout the entire development lifecycle, while also enabling rapid adaptation to diverse development needs. The RflySim ToolChain is a pivotal solution designed precisely for these demands. It embraces a comprehensive, automated, and Sim2Real design approach, offering holistic technical support spanning from simulation environments to real-world implementation. This approach notably enhances development efficiency and system integration, thereby streamlining IUSS deployment with enhanced reliability and effectiveness. This paper will focus on addressing the following two challenges.

(i) How to construct an integrated full-stack toolchain platform that covers the entire lifecycle of IUSS development, considering system modeling, reliable lower control, upper swarm decision-making, and other problem requirements? The goal is to integrate development tools and resources, create an easily accessible and modular integrated platform, and accelerate the entire process from design to deployment.

(ii) How to simplify systems, unify standards and interface protocols, and provide user-friendly and intuitive user experiences, particularly for non-expert developers or beginners? The goal is to allow users to focus on specific modules of interest without requiring expertise in specialized knowledge, thereby simplifying the learning curve, accelerating technology sharing and secondary development, and enhancing the platform’s versatility and adaptability.

## 1.3 Contributions

In order to address the challenges and shortcomings inherent in the rapid development and validation of

IUSSs, this paper introduces the RflySim ToolChain. It has been developed with the specific objective of meeting the demands of developing unmanned platform control systems, coordinating large-scale swarms, advancing artificial intelligence vision, and other leading-edge research areas. The ToolChain is capable of supporting multiple phases of the development process, including the modeling of individual unmanned vehicles, the design and validation of control algorithms, and the collaboration and decision-making processes within swarm systems. The overarching objective is to develop an integrated design toolset that encompasses modeling, simulation, testing, and deployment. The RflySim ToolChain embraces a model-based design (MBD) philosophy and integrates established autopilot systems such as Pixhawk/PX4, robust algorithm design, simulation tools, and the robot operating system (ROS). It prioritizes compatibility with intelligent hardware, facilitating seamless transitions from simulation to HIL environments and eventually to real-world testing. This integrated approach ensures a smooth Sim2Real connection, significantly expediting the progression of unmanned swarm systems from conceptualization to practical implementation. Through the RflySim ToolChain, researchers and developers gain access to a comprehensive and efficient development environment covering modeling, simulation, algorithm design, and deployment. This toolchain effectively bridges the gap between simulation and real-world deployment, enabling rapid prototyping, rigorous testing, and validation of IUSSs. Ultimately, it supports the evolution of IUSSs from theoretical concepts to impactful real-world applications.

RflySim ToolChain's application-level advantages are summarized as follows:

(i) Highly modular comprehensive solution: By adopting advanced architectural design principles to seamlessly integrate three essential modules: modeling and simulation, lower reliable control, and upper swarm decision-making. This modular approach facilitates streamlined management across the entire lifecycle, from initial model creation and simulation verification to real-time testing and ultimate deployment. By enhancing system verifiability and flexibility, this modular design supports the comprehensive development and operational adaptability of IUSSs.

(ii) Plug-and-play automated closed-loop testing: The RflySim ToolChain features an intuitive graphical interface and streamlined interactive design, allowing users to initiate simulations effortlessly with a single click. This user-friendly approach simplifies tasks like

environment setup, model initialization, and sensor simulation through straightforward configurations, achieving seamless "plug-and-play" functionality. Within its closed-loop testing environment, the ToolChain facilitates convenient and efficient system testing and validation, thereby automating the entire development process for IUSSs.

(iii) Seamless Sim2Real transition: The RflySim ToolChain facilitates a smooth transition from simulation to real-world deployment by employing high-fidelity model design, a standardized and scalable system architecture, and performance verification at each phase. This integration guarantees the dependable and rapid deployment of unmanned systems from simulated environments to real-world applications, thereby greatly improving the efficiency of IUSS development.

(iv) Open interfaces and community support: The RflySim ToolChain provides a range of application programming interface (API) interfaces and software development kit (SDK) toolkits, enabling users to customize functional modules and integrate third-party software as needed. This capability enhances the platform's functionality and expands its application possibilities. Moreover, we have established a developer community to foster knowledge and technology exchange, thereby advancing the development of IUSS technologies.

## 1.4 Organization

The structure of this paper is outlined as follows: Section 2 provides an overview of RflySim ToolChain. Section 3 details the implementation approaches of the key modules. Section 4 presents case studies demonstrating the ToolChain's capabilities. Section 5 concludes this paper.

## 2. Overview of RflySim ToolChain

This section offers a comprehensive description of the RflySim ToolChain, encompassing its simulation platform architecture, key modules, and the typical development workflow.

### 2.1 Simulation platform architecture

As shown in Fig. 1, the platform diagram of RflySim ToolChain consists of several key modules, including the lower controller, motion simulation model, three-dimensional (3D) simulation model, and upper swarm decision-making controller. Together, these form a closed-loop simulation and development system compatible with software-in-the-loop (SIL) and HIL simulations.

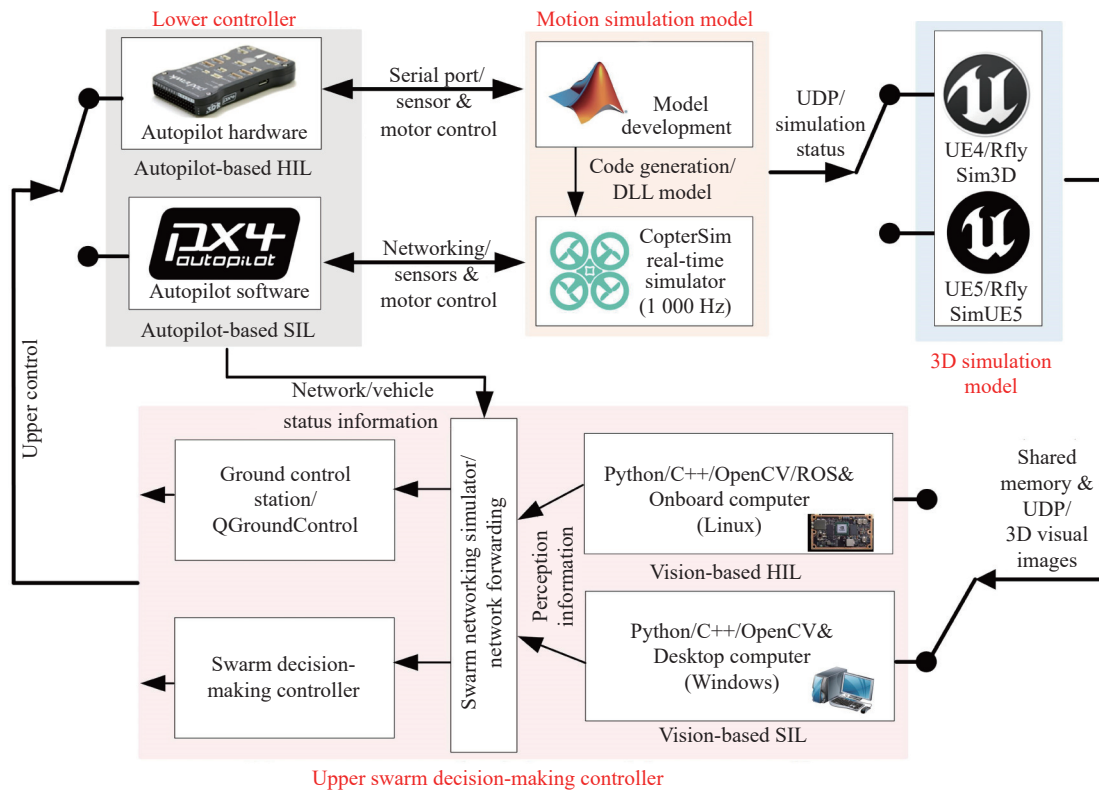


Fig. 1 Platform architecture of RflySim ToolChain

(i) Motion simulation model: This module focuses on developing a high-precision six-degree-of-freedom kinematic and dynamic vehicle model for unmanned systems. The model is then transformed into C/C++ code through automated code generation and compiled into a dynamic-link library (DLL). This DLL model is imported into the CopterSim simulation software to enable real-time simulation. CopterSim receives motor control signals from the “lower controller” and inputs them into the DLL model interface, driving the model in real time at a frequency of 1 000 Hz. It provides the “lower controller” with feedback from simulated sensor data that emulates real-world sensor readings. Simultaneously, the simulation state data is transmitted to the “3D simulation model” for visualization.

(ii) 3D simulation model: This module supports the RflySim3D engine based on UE4 and the RflySimUE5 engine based on UE5. It handles visual display, physical collision detection, network communication performance testing, loading global large-scale satellite imagery, and generating visual sensor data.

(iii) Lower controller: This module supports integration with open-source autopilot systems and allows users to customize controller designs. Leveraging the

automatic code generation capabilities, it deploys controllers directly onto hardware. Through USB connection and serial communication, this module closely coordinates with the “motion simulation model”, enabling a real-time exchange of control signals and sensor feedback.

(iv) Upper swarm decision-making controller: This module integrates the visual perception layer, network simulation layer, and swarm control layer. It processes input data primarily from visual sensors in the “3D simulation model” and vehicle state data from the “lower controller”, outputting higher-level control commands back to the “upper controller”. The visual perception layer employs visual perception algorithms on onboard sensor data to generate crucial information such as target localization, tracking, self-position estimation, and environmental situational awareness. The network simulation layer ensures efficient data transmission and sharing among multiple vehicles, managing dynamic network topologies and simulating communication interference. The swarm control layer takes a global perspective by considering the state of all vehicles, generating control commands based on specific mission requirements, and dispatching them to the corresponding “lower controller”.

## 2.2 Key modules

The RflySim ToolChain decomposes the core functionalities of the simulation platform depicted in Fig. 1, encompassing the modeling and simulation module (3D scene modeling, vehicle motion modeling), the lower reliable control module (lower control and filtering estimation, external control and trajectory planning, safety assessment and fault diagnosis), and the upper swarm decision-making module (visual perception and obstacle avoidance, communication protocols and swarm networking, swarm control and game confrontation).

The modeling and simulation module focuses on creating realistic 3D environments and dynamics models, providing a solid foundation for subsequent testing. The lower reliable control module is responsible for ensuring the real-time control and stability of UAVs. It handles tasks such as trajectory planning, filtering estimation, and safety assessments, thereby ensuring precise execution of flight maneuvers. Meanwhile, the upper swarm decision-making module coordinates the collaborative efforts of multiple UAVs through intelligent decision-making and efficient communication. By integrating these three modules, the RflySim ToolChain supports various phases of development—design, testing, control, and decision-making—significantly improving efficiency, stability, and safety. This architecture establishes a comprehensive chain of simulation environments that seamlessly connects lower-level control functions with upper-level decision-making processes. The detailed description of each is as follows.

(i) Modeling and simulation module: This module is responsible for creating motion models for unmanned vehicles and virtual environments. It involves 3D scene modeling and uses a physics engine to simulate environmental factors such as wind speed, lighting, structures, and terrain, thereby providing realistic flight testing scenarios for UAVs. Additionally, it establishes high-precision six-degree-of-freedom dynamic models for various UAV types, including rotorcraft, fixed-wing, and vertical takeoff and landing vehicles. This foundational modeling provides highly realistic testing environments for the subsequent control and decision-making modules.

(ii) Lower reliable control module: This module ensures real-time control and stability of unmanned vehicles. It integrates lower control algorithms and filtering estimation techniques. The control algorithms precisely regulate flight attitude, velocity, and position, while filtering techniques process sensor data to enhance command accuracy and system robustness against disturbances. Building on these foundations, the module imple-

ments external control and trajectory planning techniques, enabling advanced path planning and control. This allows unmanned systems to navigate autonomously, avoid obstacles, and execute complex tasks. Additionally, the module includes safety assessment and fault diagnosis functionalities. Using HIL simulation technology, potential system failures can be detected and addressed proactively, ensuring the reliable operation of unmanned vehicles.

(iii) Upper swarm decision-making module: This module facilitates efficient collaboration and intelligent decision-making among unmanned vehicles. It incorporates visual perception technologies, advanced communication protocols, and decision algorithms based on game theory to create a structured decision-making framework. By integrating high-precision visual sensor data and obstacle avoidance algorithms, UAVs can autonomously identify and track targets while making quick evasive decisions in complex environments, ensuring flight safety. Moreover, the communication protocol stack enables stable and efficient information exchange among UAVs, adapting to changing network conditions. This reliable communication underpins formation control, task allocation, and resource management in swarm operations. The incorporation of game-theoretic mechanisms allows the swarm system to adjust strategies automatically using optimization algorithms in situations with limited resources and conflicting objectives, thus maximizing overall effectiveness.

## 2.3 Typical development workflow

In this subsection, we outline a standardized process for the core modules of the RflySim ToolChain and their integration into a typical development workflow, as shown in Fig. 2. The core development process of the RflySim ToolChain comprises five phases:

(i) Modeling phase: The primary objectives of this phase include constructing 3D scene models using the Unreal Engine. By leveraging development tools and employing techniques such as mechanism analysis and system identification, a high-fidelity simulation model of unmanned vehicles is established. This phase provides a real-time and reliable foundation for models and a testing environment for subsequent phases.

(ii) Control and algorithm development phase: The primary objective of this phase is to develop the “brain” of the IUSS by implementing essential functionalities of the lower reliable control module and the upper swarm decision-making module. Control algorithm design and filtering estimation techniques are employed to ensure stable

maneuvering of unmanned vehicles in dynamic environments. External control and trajectory planning techniques establish efficient paths for the vehicles. Safety assessment and fault diagnosis systems are integrated to guarantee the safety of flight missions. The upper swarm decision-making module incorporates visual perception

and control algorithms to enable autonomous obstacle avoidance in complex environments. Communication protocols and swarm networking facilitate efficient coordination and information exchange among multiple unmanned vehicles, providing strategic support for swarm operations.

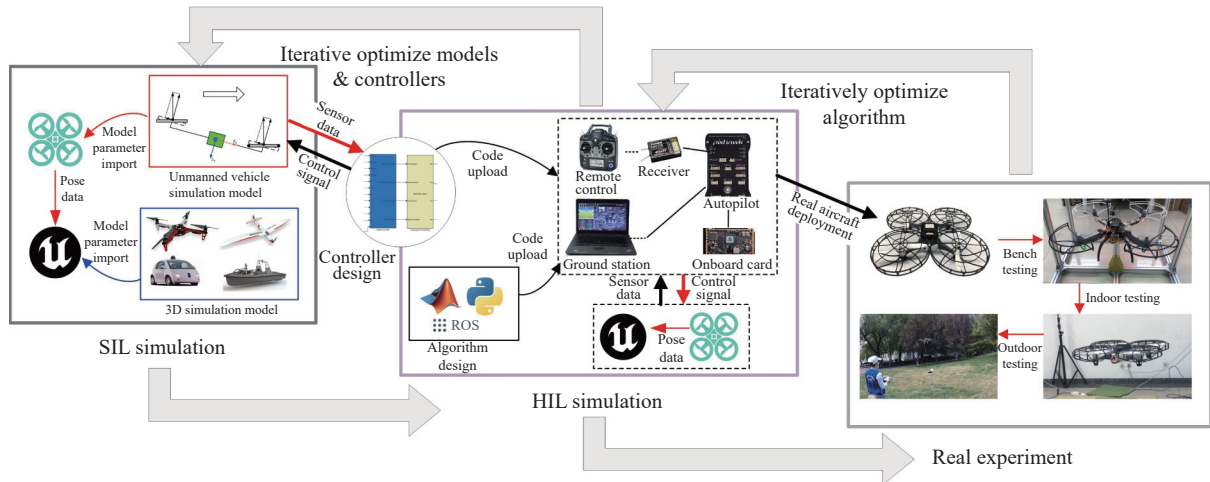


Fig. 2 Typical development process of RflySim ToolChain

(iii) SIL simulation phase: This phase aims to validate the designed control logic and algorithm models in a simulated environment. The Simulink vehicle model is compiled into a DLL and integrated into the CopterSim model simulator. The 3D scene models, created using the unreal engine, are imported into the RflySim3D/UE5 engine for real-time simulation, replicating the flight status of unmanned vehicles. Simulation data are fed back to the control algorithms, enabling closed-loop control. Throughout this phase, users can specify input parameters and closely observe the performance of the control and algorithms within the simulation software. Iterative refinement and optimization are carried out to meet the desired performance requirements.

(iv) HIL simulation phase: This phase involves integrating and testing the algorithms validated during the SIL simulation with actual hardware. The Simulink controller algorithm is converted into code and uploaded to the autopilot system. Virtual signal lines in Simulink are replaced with physical USB connections to simulate sensor inputs and control outputs during real flight processes. CopterSim transmits sensor data to the autopilot through the USB interface. The autopilot processes the sensor data, and the estimated state information is communicated back to the controller via an internal message bus. The controller then sends pulse-width modulation (PWM) control commands for each motor back to CopterSim through the USB, establishing a closed-loop HIL simula-

tion. This setup validates the algorithm's performance under conditions closely resembling real-world scenarios and tests the hardware's compatibility and reliability.

(v) Flight test phase: After successful validation in the preceding phases, the control algorithms are implemented on unmanned vehicles for real-world flight tests. This phase serves as the final verification of the entire development process, aiming to ensure the efficient and safe operation of unmanned vehicles in practical applications.

During the typical development process, we utilize credibility assessment theory [26] to measure reliability at each stage. This theory focuses on inputting identical signals into both the HIL simulation system and the real system to compare the resulting errors. We select input signals from various aspects, such as system performance, time-domain response, and frequency-domain response, to provide a comprehensive view of system characteristics. By employing the same control system in both simulation and experimental settings, we effectively reduce errors arising from hardware and software differences, thereby establishing a robust foundation for transitioning to the next stage. The theoretical model for credibility assessment is defined as follows:

$$\eta \triangleq f_{\text{norm}}(e, \varepsilon) = \frac{K_e \cdot \varepsilon}{\sqrt{(K_e \cdot \varepsilon)^2 + e^2}} \quad (1)$$

where  $f_{\text{norm}}(e, \varepsilon)$  represents a normalized mapping func-

tion that satisfies the following mapping relationship:

$$\begin{cases} e \rightarrow 0 \Rightarrow \eta \rightarrow 1 \\ e \rightarrow \infty \Rightarrow \eta \rightarrow 0 \end{cases} \quad (2)$$

This indicates that  $\eta \rightarrow 0$  represents a situation where the error is very large, indicating significant discrepancies between the simulation results and the actual system, resulting in low model accuracy and credibility. Conversely, when  $\eta \rightarrow 1$  reflects a small error, it signifies that the simulation results closely match the actual system, resulting in high model accuracy and credibility. In (1),  $K_e \in \mathbf{R}_+$  serves as a coefficient for the unified evaluation criteria mapping, while  $\eta_{\text{pass}}$  represents the unified evaluation standard line, typically aligned with the actual evaluation system. This line corresponds to a passing threshold of 60 points within a percentage-based evaluation framework.

In the theoretical model of credibility assessment, the normalized mapping function defines the relationship between simulation errors and those of the actual system, highlighting the model's accuracy and reliability. When significant errors occur, the model's credibility declines, which can adversely affect individual outcomes and potentially trigger cascading effects throughout the system. If errors or inaccuracies arise during the calibration phase and are not promptly identified and corrected, they can result in flawed input data, ultimately impacting the system's dynamic responses and decision-making processes.

Overall, by integrating the various phases described above, the RflySim ToolChain enables a fully automated progression in the development of IUSSs, creating a comprehensive ecosystem for system development and validation. To ensure the reliability of Sim2Real, we have implemented multi-layered safety mechanisms along with fault detection and recovery strategies. We have also enhanced accuracy through high-fidelity physical modeling and sensor simulation, which minimize discrepancies between simulated and real-world environments. Additionally, the simulation engine's performance has been optimized to support real-time simulation, thereby accelerating the development cycle. This hierarchical structure, which encompasses modeling, control, and decision-making, enables developers to perform full-stack development, testing, and validation on a single platform. As a result, this approach significantly improves development efficiency and system reliability, further advancing the application of Sim2Real.

#### 2.4 Target users and learning objectives

As illustrated in Table 1, the RflySim ToolChain provides diverse interactive experiences and objectives tai-

lored to users with varying backgrounds. For academic researchers, the ToolChain facilitates the development and validation of complex algorithms by offering detailed case studies, tutorials, comprehensive documentation, and sample code, which help users quickly familiarize themselves and engage in in-depth exploration. For industry engineers, the focus is on the practical application of the ToolChain in real-world engineering projects. This includes methods for system integration, testing, and deployment, supported by practical guides and technical assistance.

**Table 1** Target users and learning curve objective of RflySim ToolChain

Additional support	Target user	Learning objective
Provide introductory tutorials, video courses, advanced functions and customized solutions.	Academic researchers	Powerful support for complex algorithm development and validation, with detailed case studies, tutorials, extensive documentation, and sample code to get started quickly
Establish online forums and user communities to encourage communication and mutual assistance among users, and provide technical support and communication.	Industry engineer	Highlights methods for system integration, testing and deployment, and provides practical guidance, rich interfaces and technical support services to solve problems encountered in actual applications

### 3. Implementation of key modules in RflySim ToolChain

#### 3.1 Modeling and simulation module

Within the modeling and simulation module of the RflySim ToolChain, there is a strong interrelationship between 3D scene modeling and vehicle motion modeling. The 3D scene modeling involves creating highly accurate environmental elements, such as terrain, buildings, and vegetation, to establish a realistic operational environment for unmanned vehicles. Conversely, vehicle motion modeling integrates the dynamics model, sensor model, and fault model, ensuring the simulation accurately represents the vehicle's flight performance and response behavior.

##### 3.1.1 3D scene modeling

The 3D scene modeling framework encompasses the entire process from scene modeling to data integration, network synchronization, physics simulation, and visualization. Firstly, it combines blueprint modeling and XML description with tools like UE Marketplace, Twinmotion,

Cesium, and 3DsMax, enabling the construction of detailed scenes with weather effects, user interfaces, and static/dynamic obstacles. The use of 3D scanning technology further enhances the realism of these scenes. Additionally, the framework integrates a variety of visual and non-visual sensor data, such as red-blue-green (RGB) images, infrared thermal imaging, point cloud data, and sound information, to enhance environmental perception capabilities. For network data synchronization, the framework employs shared memory and user datagram protocol (UDP) communication to ensure seamless data exchange. It also utilizes terrain collision detection and synchronization servers to maintain consistency in scene data. At the simulation level, motion and collision models are developed using Simulink, supported by automatic code generation and dynamic library techniques. This approach facilitates the flexible importation of models and enables highly realistic physics simulation.

### 3.1.2 Vehicle motion modeling

The RflySim ToolChain employs the MBD approach to develop diverse templates for unmanned vehicles, including cars, fixed-wing aircraft, and vertical takeoff and landing (VTOL) aircraft. It offers a unified model structure and format. As shown in Fig. 3, the unified modeling framework for unmanned vehicles consists of three fundamental subsystems: the airframe subsystem, the sensor subsystem, and the 3D environment subsystem. The airframe subsystem receives control commands and produces flight states as outputs. The 3D environment subsystem uses the flight state data to generate real-time environmental feedback. The sensor subsystem converts this environmental feedback into sensor readings, which are then fed back to the control system to calculate subsequent control commands. This structured approach ensures a cohesive and efficient modeling process for various types of unmanned systems.

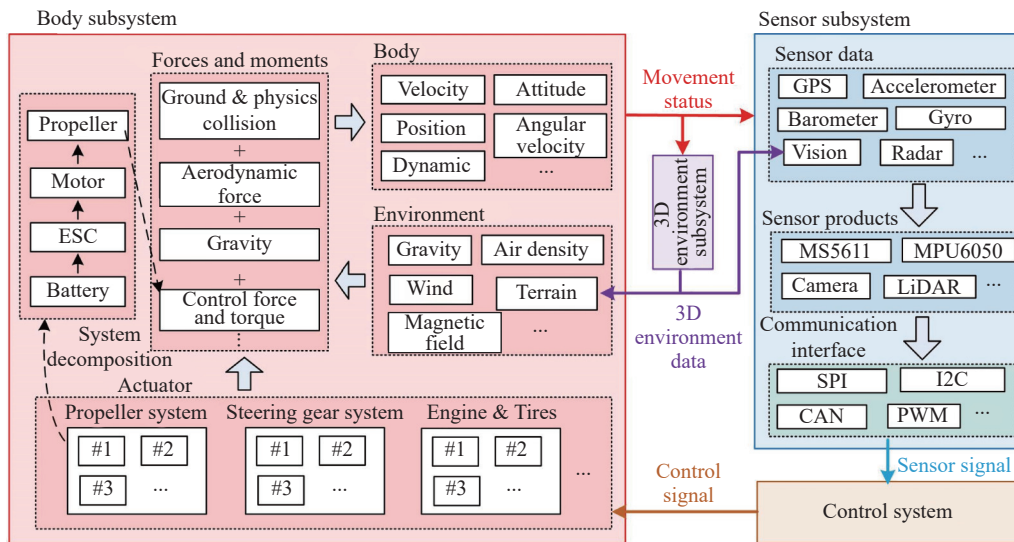


Fig. 3 Unified modeling framework for unmanned vehicles

The RflySim ToolChain adopts a comprehensive modeling approach that integrates energy consumption, motion, and fault models. The motion model captures the real-time dynamic behavior of individual components, while the energy consumption model emphasizes long-term system performance and stability. The fault model identifies deviations from normal operational states, caused by both internal and external factors. Together, these models provide a thorough understanding of the behavior of individual drone components as well as the overall system, enhancing both analysis and decision-making capabilities.

### 3.2 Lower reliable control module

Within the lower reliable control module of the RflySim ToolChain, there are interconnected components: the lower control and filtering estimation, external control and trajectory planning, and safety assessment and fault diagnosis modules. The lower control and filtering estimation module provides real-time and precise system state data, forming the basis for effective external control and trajectory planning. The external control and trajectory planning module performs complex tasks, utilizing the lower controllers and relying on accurate data from the control module for fine-tuning and optimization. The

safety assessment and fault diagnosis module is crucial throughout the control process, continuously monitoring system operations, evaluating and diagnosing potential faults, and ensuring the overall safety and reliability of the control system.

### 3.2.1 Lower control and filtering estimation

This module follows a structured development process as illustrated in Fig. 2, covering the workflow from SIL simulations to HIL simulations and culminating in real-world flight experiments. The process begins with development tools used to design a simulation environment and create a comprehensive model of the unmanned vehicle. Next, control algorithms are converted from Simulink models into executable C/C++ code using automatic code generation techniques. These codes are deployed onto open-source autopilot systems like Pixhawk and tested on an HIL simulation platform integrating the CopterSim model simulator and RflySim3D/UE5 engine. Here, the control algorithms interact with actual hardware components, including the flight control board, receiver, and telemetry, simulating various operational scenarios within a realistic flight environment. During actual flight testing, the control system's performance is validated through controlled indoor bench tests, followed by indoor and outdoor flight tests to assess the controller's stability and reliability under different flight conditions.

### 3.2.2 External control and trajectory planning

The autopilot system integrates advanced control and estimation algorithms with the airframe and propulsion, enabling the creation of intelligent agents capable of responsive actions such as adjusting attitude, velocity, and position based on external commands. In contrast, external control involves transmitting commands to these agents through interfaces, facilitating higher-level functionalities like trajectory planning. This framework supports two primary control modes: direct and indirect, catering to various mission needs and scenarios. Direct control encompasses real-time control via remote systems, mission planning through ground control, and complex task assignments via onboard systems. Indirect control is prevalent in unmanned swarm systems, characterized by centralized decision-making and decentralized execution.

### 3.2.3 Safety assessment and fault diagnosis

The safety assessment and fault diagnosis module within the RflySim ToolChain is designed with a highly inte-

grated and adaptable framework, structured around four key steps [29]. The initial phase of the process involves a comprehensive consideration of potential fault scenarios, encompassing a range of possibilities from propulsion system failures to environmental changes and communication interferences. This is achieved through the utilization of modeling, simulation, and fault model design techniques. This approach guarantees comprehensive preparation for a multitude of operational challenges. Secondly, the module integrates simulation with onboard hardware in a seamless manner, enabling the real-time injection of faults in a simulated environment to validate the system's response capabilities effectively. Thirdly, an efficient publish-subscribe mechanism is employed to facilitate the rapid dissemination of fault events to subsystems, thereby ensuring prompt responses and effective management of fault conditions system-wide. Lastly, real-time data analysis and predictive modeling techniques are employed to monitor and analyze any deviations that may occur during system operation. The integrated approach ensures the stable operation of unmanned vehicles across diverse and challenging environments, providing users with reliable operational assurance.

## 3.3 Upper swarm decision-making module

The upper swarm decision-making module, which forms part of the RflySim ToolChain, comprises three principal modules: visual perception and obstacle avoidance, communication protocols and swarm networking, and swarm control and game confrontation. The visual perception and obstacle avoidance module employs cameras and LiDAR sensors to capture and process real-time environmental data in order to facilitate the generation of a representation of the surrounding environment. Subsequently, this information is efficiently transmitted to other swarm vehicles via the communication protocols and swarm networking module, thereby ensuring reliable data exchange. The swarm control and game confrontation module employs this shared information to optimize global decision-making processes and engage in strategic interactions based on game theory, thereby enhancing the overall performance of swarm operations.

### 3.3.1 Visual perception and obstacle avoidance

Fig. 4 illustrates the architecture used for visual perception and obstacle avoidance, incorporating both SIL and HIL simulations. This architecture comprises the following components.

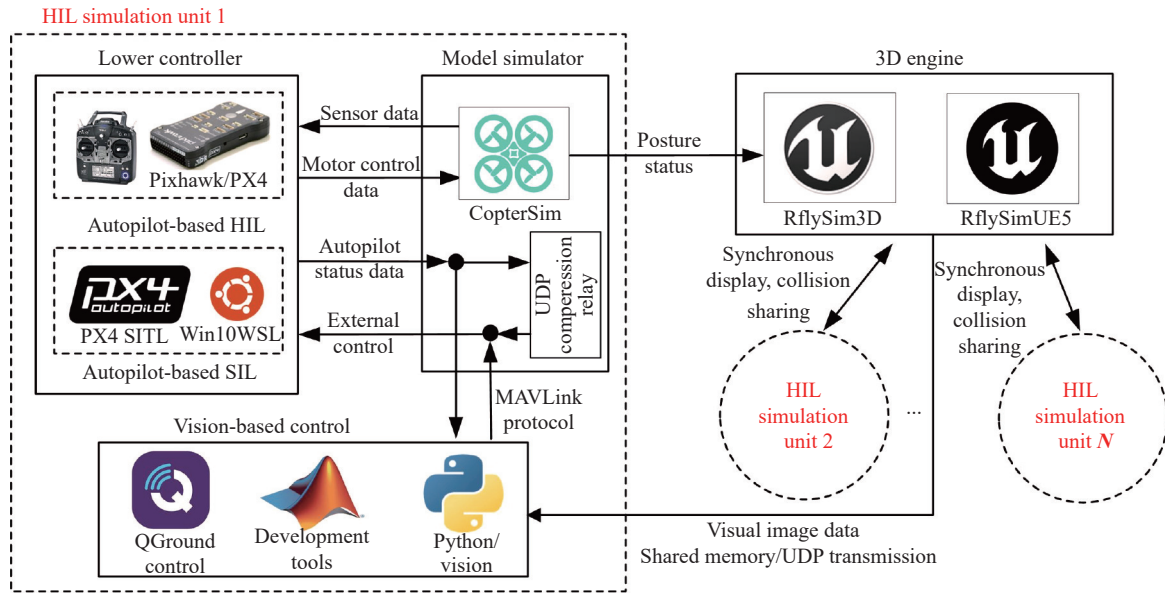


Fig. 4 Simulation framework of visual perception and obstacle avoidance

(i) Vision-based SIL: The algorithms run directly on the same PC environment as RflySim3D/UE5 and CopterSim, communicating with the lower controller via the MAVLink protocol. This communication facilitates the exchange of sensor data, control commands, and autopilot status information. Visual sensor data are efficiently processed by the visual perception algorithm through inter-process communication (IPC) interfaces like shared memory or local network interfaces. This setup enables rapid development and validation using Python environments provided by RflySim ToolChain.

(ii) Vision-based HIL: RflySim3D/UE5 facilitates the transmission of compressed visual images to the onboard computer via local area network communication methods, including UDP and Redis. Upon reception, the visual data is transformed into ROS image messages. Subsequently, the onboard visual perception program subscribes to the aforementioned messages, thereby enabling the performance of algorithmic processing. Subsequently, the processed visual perception information is disseminated to other programs via the utilization of specific protocols. The resulting control commands are relayed back to the lower controller, thereby enabling the operation of CopterSim and the execution of specific tasks. This configuration establishes a closed-loop control feedback system.

### 3.3.2 Communication protocols and swarm networking

RflySim ToolChain employs a hybrid simulation framework for the simulation of swarm networks. This frame-

work precludes direct communication between simulation or physical nodes, necessitating that all data transmission traverse a network simulator as an intermediary. This approach guarantees the control and precision of the communication process. Each simulation or physical node transmits data to the corresponding node within the network simulator, which then relays this data to the internal simulation network. The network simulator assesses the viability of communication between nodes based on internal protocols, simulating the data transmission process as it would occur in a real network environment. The efficacy of communication is contingent upon a number of factors, including the presence of physical obstructions, the potential for electromagnetic interference, the distance between nodes, the employed routing protocols, and the communication bandwidth. This simulation mechanism enables the RflySim ToolChain to accurately replicate communication challenges in complex environments. It is particularly well-suited for the testing of communication within swarm scenarios, and can be adapted for a variety of unmanned systems and application scenarios.

### 3.3.3 Swarm control and game confrontation

RflySim ToolChain utilizes internal shared memory communication and optimized network data transmission mechanisms to enable control and interaction within large-scale unmanned systems simulation environments. Fig. 5 illustrates how RflySim ToolChain's swarm control and game confrontation capabilities support efficient collaboration and competition among multi-agent systems

through advanced communication methods and streamlined data processing techniques. It allows for running multiple S/HIL simulations on each computer and facilitates simulations of expansive scenarios involving thousands of vehicles. In practice, the system integrates CopterSim and RflySim3D/UE5 to simulate unmanned vehicles and complex environmental interactions accurately. The consistency between simulation and actual hardware responses is maintained by integrating onboard controllers into the simulation process. Efficient data

exchange between onboard computers for visual perception processing and the simulation computer, coupled with mechanisms for data aggregation, compression, and request-based communication, ensures seamless and real-time data flow in large-scale simulations. The swarm command computer acts as the central unit responsible for data aggregation, overall control, and mission planning. It provides feedback and instructions to individual simulation units, thereby facilitating collaborative operations among unmanned vehicles.

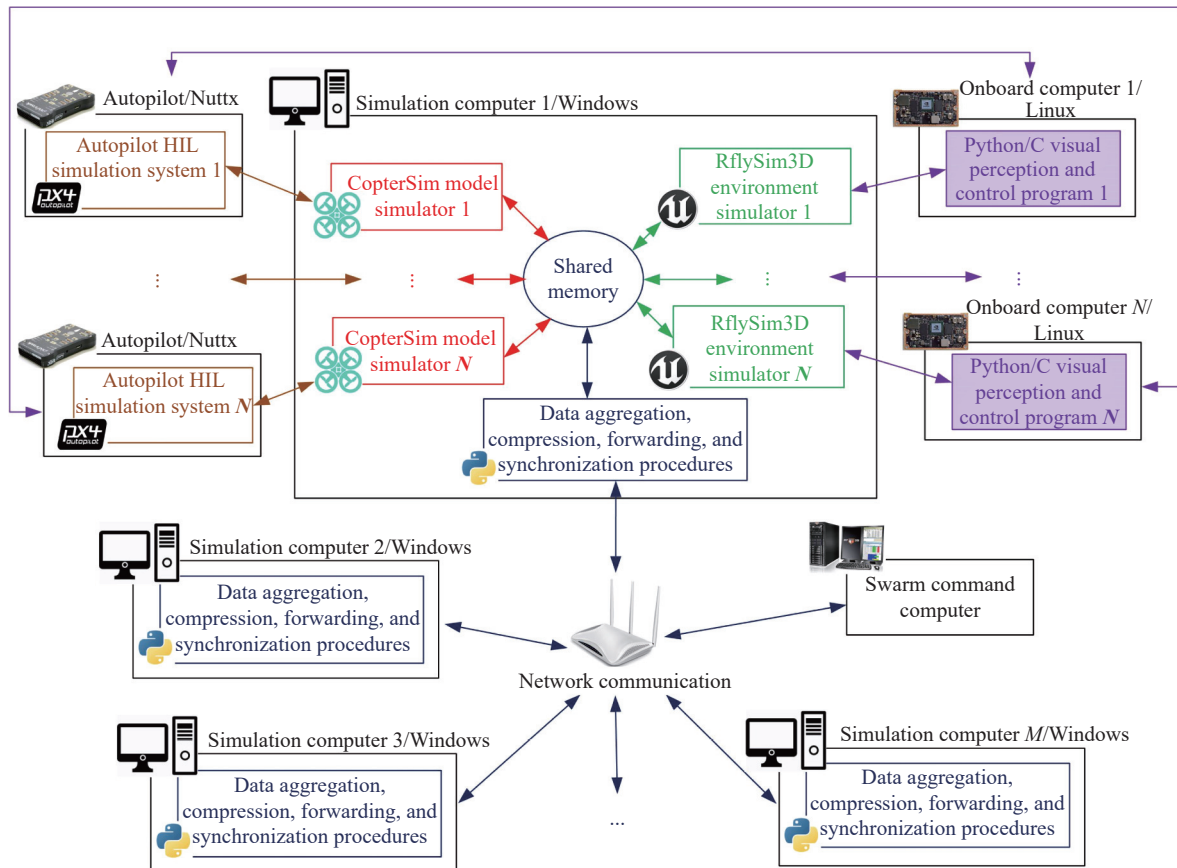


Fig. 5 Implementation mechanism of swarm control and game confrontation

#### 4. Applications of RflySim ToolChain

This section offers a detailed exploration of specific instances and application scenarios involving RflySim ToolChain, with the objective of highlighting its distinctive advantages and potential applications within the field of unmanned vehicle simulation and development. For a comprehensive understanding of RflySim ToolChain's capabilities and to view video demonstrations of its application cases, see [32].

To meet a diverse range of hardware requirements, we provide an extensive set of software and hardware interface options. This includes support for standard commu-

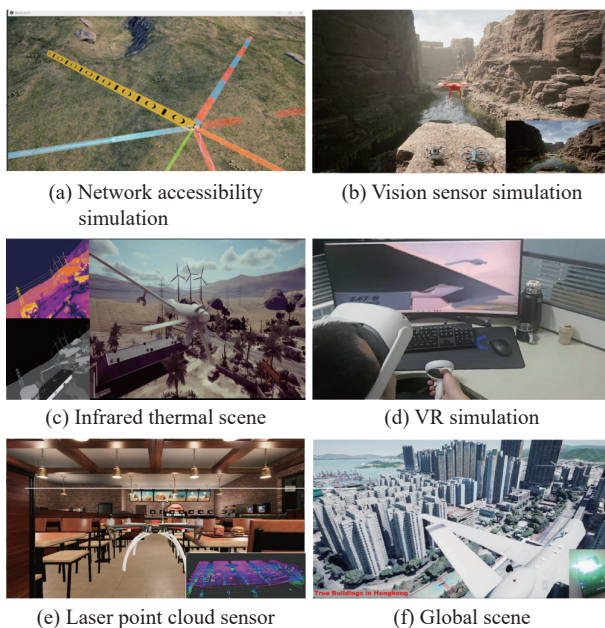
nication protocols like MavLink and ROS, along with compatibility with various operating systems such as Windows and Linux. The toolchain features a modular architecture that enables users to choose and integrate the modules that best fit their needs, while also allowing third-party developers to contribute new plugins or features. It is essential to note that the RflySim ToolChain platform is designed for high-fidelity simulations of large and complex unmanned systems and supports the development of visual and artificial intelligence (AI) algorithms. Consequently, this platform has specific system configuration requirements, which are outlined in Table 2.

**Table 2 Recommended computer configuration items**

Item	Minimum recommended specification
Operating system	Windows 10×64 (version 1 809 or higher)
CPU	Intel the 8th generation Core i7 processor or higher, or equivalent AMD processor
Graphics card	NVIDIA GTX 2060 or higher, or equivalent AMD graphics card
Memory	16 GB RAM or higher, DDR3 1 600 MHz or higher
Storage	At least 40 GB free space on the installation drive (SSD recommended)
Monitor	Resolution of at least 1 080 P (1 920×1 080) or higher (dual monitors recommended)
Interface	At least one USB Type A port (expandable via adapters)

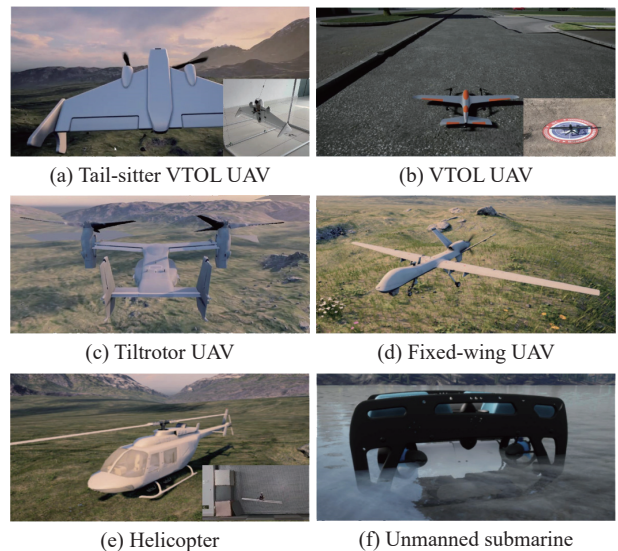
#### 4.1 Multi-modal multi-scale 3D scenes

Fig. 6 illustrates the diverse 3D scenario applications developed using RflySim ToolChain, which integrates several key functional modules. These include simulation of network communication reachability, visual sensors, infrared thermal scenes for electro-optical payloads, virtual reality (VR), and laser point cloud sensors. Importantly, its capability to import arbitrary scenes allows users to adapt it flexibly to various simulation needs. With these features, RflySim ToolChain supports a wide range of applications such as urban environment simulation, disaster response exercises, and other complex environments, fulfilling the collaborative simulation requirements of large-scale unmanned swarms. For detailed case studies, see [33].


**Fig. 6 3D scene modeling cases of RflySim ToolChain**

#### 4.2 Multi-source heterogeneous unmanned vehicle modeling

RflySim ToolChain refines its unified modeling framework, illustrated in Fig. 3, by organizing it into a series of digital unit modules within a comprehensive model library. This library not only simulates individual unmanned vehicles accurately but also encompasses the entire ecosystem. Fig. 7 showcases this approach effectively, featuring models of tail-sitter VTOL drones, helicopters, tiltrotor aircraft, and unmanned ground vehicles. These models are grounded in precise mathematical formulations, ensuring an accurate portrayal of their dynamic behaviors. In essence, RflySim ToolChain expedites the iterative process from initial design concepts to practical testing through its modular and scalable modeling approach. This advancement significantly enhances research capabilities in unmanned swarm cooperative systems. For in-depth case studies demonstrating the capabilities of RflySim ToolChain, please refer to [34].


**Fig. 7 Multi-configuration unmanned vehicle case of RflySim ToolChain**

#### 4.3 Swarm trajectory planning and control

RflySim ToolChain integrates a diverse library of control algorithms and a flexible framework for control strategies, supporting interfaces for ground stations, remote controllers, and customized algorithmic logic. It also provides comprehensive data analysis tools to quantitatively assess swarm performance metrics such as flight efficiency and path accuracy. Users can configure complex flight paths and mission commands via the ground station, facilitating macroscopic coordination of the swarm. Moreover, RflySim ToolChain supports customized algorithmic control, enabling swarms to dynamically adjust

flight trajectories in response to environmental changes or mission requirements. This capability enhances functionalities like intelligent obstacle avoidance and target tracking. These robust control interfaces and strategies significantly increase the autonomy and cooperative operational efficiency of unmanned swarms in intricate environments. For detailed case studies illustrating these capabilities, see [35].

#### 4.4 Multiple fault type simulation

RflySim ToolChain features a robust fault simulation module that encompasses scenarios such as propulsion unit failures, sensor anomalies, communication interruptions, and logical errors in decision-making units. This module aids researchers in understanding how individual faults impact overall unmanned system performance and assesses system stability and resilience against concurrent multiple faults. Additionally, RflySim ToolChain integrates an automated testing and data collection module that efficiently executes numerous fault simulation tests. It automates the recording and analysis of ground truth data and internal state estimation from the autopilot during simulations. This comparative analysis mechanism forms the basis for scientifically assessing unmanned system safety, allowing developers to pinpoint fault patterns and diagnose failure causes accurately. Moreover, RflySim ToolChain accumulates a substantial dataset of fault data, supporting the training of AI models for fault prediction. This capability enhances the platform's capacity to predict and preemptively address potential system failures. For detailed demonstrations illustrating these capabilities, see [36].

#### 4.5 Vision-based HIL algorithm verification

The platform integrates real-time 3D scene rendering techniques, ensuring visual algorithms remain responsive when processing dynamic data within realistic simulation scenarios. This real-time capability contrasts with traditional offline testing methods or static datasets, enabling algorithm validation in dynamically configured virtual environments without the need for extensive map preparation. By eliminating the requirement for pre-built global map data, RflySim ToolChain saves considerable time and resources typically spent on map creation. It serves as a robust platform for optimizing algorithm performance and enhancing the autonomous exploration and real-time decision-making capabilities of unmanned systems in complex and dynamic environments. For in-depth demonstrations illustrating these capabilities, see [37].

#### 4.6 Swarm networking communication delay simulation

The platform integrates real-time 3D scene rendering

techniques, ensuring visual algorithms remain responsive when processing dynamic data within realistic simulation scenarios. This real-time capability contrasts with traditional offline testing methods or static datasets, enabling algorithm validation in dynamically configured virtual environments without the need for extensive map preparation. By eliminating the requirement for pre-built global map data, RflySim ToolChain saves considerable time and resources typically spent on map creation. It serves as a robust platform for optimizing algorithm performance and enhancing the autonomous exploration and real-time decision-making capabilities of unmanned systems in complex and dynamic environments. For in-depth demonstrations illustrating these capabilities, see [37].

#### 4.7 Unmanned system collaborative control and decision making

Fig. 8 depicts a scenario illustrating the cooperative control of UAV swarms, showing their agility and precision in maintaining formations within complex environments. It highlights their efficient collaboration in executing missions like target search and strike through coordinated multi-vehicle operations. RflySim ToolChain also emphasizes the collaborative capabilities among heterogeneous unmanned vehicles, enabling seamless integration and cooperative operations within a unified command and control framework. This cross-platform collaboration enhances mission diversity and flexibility, supporting complex tasks such as supply transportation, search and rescue missions, and post-disaster environmental monitoring. RflySim ToolChain's advanced features in coordination and decision-making for unmanned systems drive its application in military reconnaissance, disaster response, and extend opportunities for civilian and commercial use. For a detailed exploration of these capabilities, refer to the case studies available at [38].

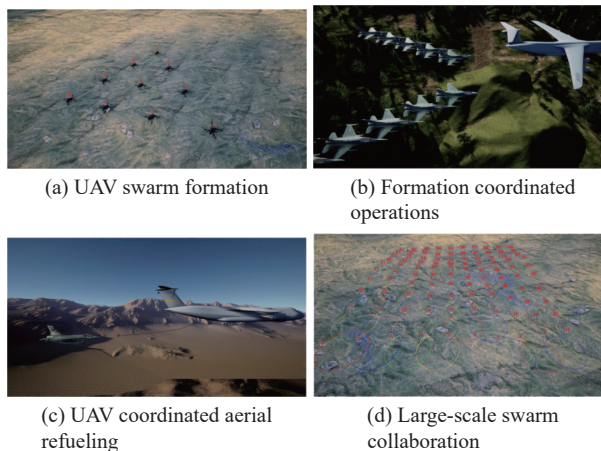
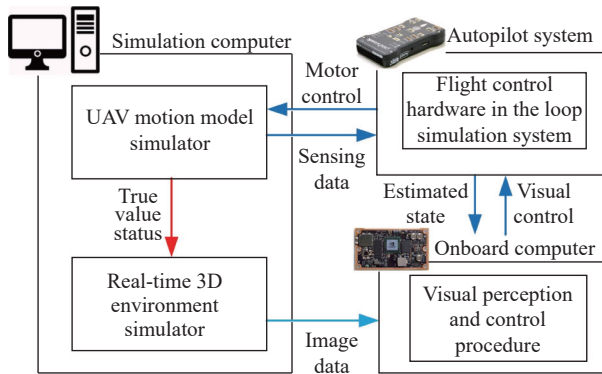


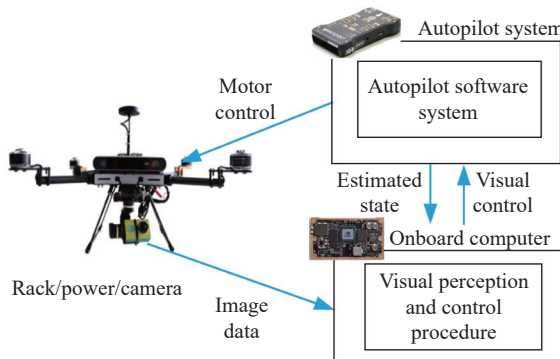
Fig. 8 Unmanned system collaborative control and decision-making case of RflySim ToolChain

#### 4.8 Rapid deployment and verification of Sim2Real

Fig. 9 presents the framework for rapidly transitioning from simulation to real-world implementation. In the HIL phase, we prioritize deploying the autopilot hardware and onboard computers, along with the corresponding frame model. Algorithms are developed and refined using a real-time simulator and rendering engine. This stage is crucial for ensuring seamless integration between hardware and software, allowing us to test algorithm performance within the simulation environment and swiftly identify and resolve potential issues. As the system progresses to the real experimental phase, the validated autopilot hardware and onboard computers are directly installed onto the frame. This approach eliminates the need for additional fine-tuning and hardware adjustments, greatly improving deployment efficiency. Additionally, by cross-validating with experimental data from the simulation phase, we can effectively evaluate the discrepancies between actual performance in real-world conditions and the simulation outcomes. This process provides essential data for further optimization. Through this rapid deployment and validation strategy, we can quickly transform simulations into practical applications in real-world settings.



(a) Hardware in the loop simulation of autopilot, communication link, and onboard computer



(b) Real machine experiment of rack, autopilot, communication link, and onboard computer

**Fig. 9** Rapid deployment architecture solution of Sim2Real

#### 4.9 Performance testing and development cases

As shown in Table 3, this section presents performance tests on key functionalities of the RflySim ToolChain across various scenarios. These include assessing computational resource and memory usage during swarm formation tasks, simulating network signal delays for multiple drones, analyzing root mean square errors (RMSE) in position control and trajectory tracking, and evaluating simulation accuracy under fault conditions. However, these tests only offer a glimpse of the toolchain's extensive capabilities. RflySim ToolChain goes beyond these specific scenarios, providing a comprehensive development environment that covers the entire lifecycle—from 3D scene modeling and low-level control with filtering and estimation, to high-level swarm decision-making and adversarial strategies based on game theory. With over 100 test cases and functional interfaces, the toolchain significantly broadens the range of applications and offers the flexibility needed for in-depth development and validation at various levels.

**Table 3** Performance test results in different scenarios

Mission scenario	Indicator	Value
Formation mission for a swarm of 100 UAVs	CPU usage/%	10
	Memory usage/GB	1.2
Simulation of a network of four UAVs with signal attenuation based on terrain	Average packet loss rate/%	27.6
	Average delay rate/ms	96.4
Validation of drone stationary flight accuracy (tracking error)	RMSE	$8e-3 \pm 2e-3$
Quality of data from an inertial measurement unit (error compared to ground truth)	RMSE	$2e-2 \pm 5e-3$
Safety test for a fault where a motor stops running	Simulation credibility/%	64.02

## 5. Conclusion and future work

This paper introduces RflySim ToolChain, a modular and standardized development toolkit designed for IUSSs. In addressing the common challenges inherent to IUSS development and application transformation, RflySim ToolChain offers cross-platform compatibility and modularization. The toolkit integrates three core modules—the modeling and simulation module, the lower reliable control module, and the upper swarm decision-making module—to support the entire lifecycle, from design to validation, thereby enhancing the efficiency of unmanned system development. A series of application cases demonstrates the robust applicability of RflySim

ToolChain in a variety of scenarios. Additionally, the toolchain incorporates comprehensive tutorials and application templates, accompanied by multiple interfaces to cater to diverse user development needs. This comprehensive approach aims to streamline IUSS development processes while ensuring practical usability and effectiveness.

While the RflySim ToolChain currently supports full-stack development processes within the IUSS framework, it faces limitations in autonomous intelligent applications. The existing decision-making modules lack sufficient speed and accuracy in complex, dynamic environments, which can result in suboptimal choices during critical situations. Additionally, usability issues in the user interface hinder the effective management of complex operations, impacting both development efficiency and user experience. In the future, the RflySim ToolChain plans to expand its capabilities and enhance its technical ecosystem. Key development areas will include improving intelligent decision-making and autonomous learning by integrating advanced machine learning and deep learning technologies. This will enhance the system's ability to understand its environment and adapt accordingly. Moreover, the incorporation of augmented reality and digital twin technologies will create a more intuitive and interactive development environment, facilitating more effective simulations and testing. To optimize user experience, the RflySim ToolChain will focus on streamlining operational processes, refining interface design, and making tools more accessible to enhance overall efficiency. Lastly, by fostering community support, encouraging user feedback, and promoting collaboration, we aim to drive collective technological innovation and progress. These efforts are geared towards further solidifying RflySim ToolChain's pivotal role in the field of IUSSs.

## References

- [1] PANG W J, MA X Y, LIANG X M, et al. Role-based bayesian decision framework for autonomous unmanned systems. *Journal of Systems Engineering and Electronics*, 2023, 34(6): 1397–1408.
- [2] MOHSAN S A H, OTHMAN N Q H, LI Y, et al. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics*, 2023, 16(1): 109–137.
- [3] OLLERO A, TOGNON M, SUAREZ A, et al. Past, present, and future of aerial robotic manipulators. *IEEE Trans. on Robotics*, 2021, 38(1): 626–645.
- [4] FAHEY K, MILLER M. Unmanned systems integrated roadmap FY 2017–2042. <https://apps.dtic.mil/sti/citations/AD1059546>.
- [5] ZHANG F B, YU J, LIN D F, et al. UnIC: towards unmanned intelligent cluster and its integration into society. *Engineering*, 2022, 12(8): 24–38.
- [6] HOWARD J, MURASHOV V, BRANCHE C M. Unmanned aerial vehicles in construction and worker safety. *American Journal of Industrial Medicine*, 2018, 61(1): 3–10.
- [7] CHAMOLA V, KOTESH P, AGARWAL A, et al. A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad Hoc Networks*, 2021, 111(1): 102324.
- [8] KE C X, CAI K Y, QUAN Q. Analysis of a uniform passive faule-tolerant control method for multiapters. *Journal of Systems Engineering and Electronics*, 2024, 35(6): 1574–1582.
- [9] WORTH D B, WOOLLEY B G, HODSON D D. SwarmSim: a framework for modeling swarming unmanned aerial vehicles using hardware-in-the-loop. *The Journal of Defense Modeling and Simulation*, 2021, 18(2): 105–124.
- [10] BELCASTRO C M, KLYDE D H, LOGAN M J, et al. Experimental flight testing for assessing the safety of unmanned aircraft system safety-critical operations. *Proc. of the 17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017. <https://doi.org/10.2514/6.2017-3274>.
- [11] NOUREEN S S, ROY V, BAYNE S B. An overall study of a real-time simulator and application of RT-LAB using MATLAB SimPowerSystems. *Proc. of the IEEE Green Energy and Smart Systems Conference*, 2017. <https://doi.org/10.1109/IGESC.2017.8283453>.
- [12] KOENIG N, HOWARD A. Design and use paradigms for gazebo, an open-source multi-robot simulator. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004: 2149–2154.
- [13] SHAH S, DEY D, LOVETT C, et al. AirSim: high-fidelity visual and physical simulation for autonomous vehicles. *Proc. of the Field and Service Robotics: Results of the 11th International Conference*, 2018: 621–635.
- [14] GUERRA W, TAL E, MURALI V, et al. FlightGoggles: photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019: 6941–6948.
- [15] SONG Y, NAJI S, KAUFMANN E, et al. Flightmare: a flexible quadrotor simulator. *Proc. of the Conference on Robot Learning*, 2021: 1147–1157.
- [16] CUI C, ZHOU X B, WANG M, et al. FastSim: a modular and plug-and-play simulator for aerial robots. *IEEE Robotics and Automation Letters*, 2024, 9(6): 5823–5830.
- [17] LI G R, LIU X Y, LOIANNO G. RotorTM: a flexible simulator for aerial transportation and manipulation. *IEEE Trans. on Robotics*, 2024, 40(2): 831–850.
- [18] FERNANDEZ C M, MOLINA M, ARIAS P, et al. Aerostack2: a software framework for developing multi-robot aerial systems. <https://arxiv.org/abs/2303.18237>.
- [19] DIMMIG C A, SILANO G, MCGUIRE K, et al. Survey of simulators for aerial robots. <https://arxiv.org/abs/2311.02296>.
- [20] TAI J J, WONG J, INNOCENTE M, et al. PyFlyt--UAV simulation environments for reinforcement learning research. <https://arxiv.org/abs/2304.01305>.
- [21] FOLK S, PAULO S J, KUMAR V. ROTORPY: a Python-based multirotor simulator with aerodynamics for education and research. <https://arxiv.org/abs/2306.04485>.
- [22] MATHWORKS. UAV toolbox. <https://www.mathworks.com/products/uav.html>.
- [23] HUANG Z H, BATRA S, CHEN T, et al. Quadswarm: a modular multi-quadrotor simulator for deep reinforcement learning with direct thrust control. <https://arxiv.org/abs/2306.09537>.
- [24] RICHTER D J, CALIX R A. Qplane: an open-source rein-

- forcement learning toolkit for autonomous fixed wing aircraft simulation. Proc. of the 12th ACM Multimedia Systems Conference, 2021: 261–266.
- [25] QUAN Q, DAI X H, WANG S. Multicopter design and control practice: a series experiments based on MATLAB and Pixhawk. Singapore: Springer Nature, 2020.
- [26] DAI X H, KE C X, QUAN Q, et al. RFLySim: automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. Aerospace Science and Technology, 2021, 114: 106727.
- [27] DAI X H, KE C X, QUAN Q, et al. Simulation credibility assessment methodology with FPGA-based hardware-in-the-loop platform. IEEE Trans. on Industrial Electronics, 2020, 68(4): 3282–3291.
- [28] WANG S, DAI X H, KE C X, et al. RflySim: a rapid multicopter development platform for education and research based on Pixhawk and MATLAB. Proc. of the IEEE International Conference on Unmanned Aircraft Systems, 2021: 1587–1594.
- [29] QUAN Q, FU R, LI M X, et al. Practical distributed control for VTOL UAVs to pass a virtual tube. IEEE Trans. on Intelligent Vehicles, 2021, 7(2): 342–353.
- [30] QUAN Q, GAO Y, BAI C G. Distributed control for a robotic swarm to pass through a curve virtual tube. Robotics and Autonomous Systems, 2023, 162: 104368.
- [31] MAO P D, FU R, QUAN Q. Optimal virtual tube planning and control for swarm robotics. *The International Journal of Robotics Research*, 2024, 43(5): 602–627.
- [32] FEISI L. Introduction: what is RflySim? <https://rflysim.com/doc/en/>.
- [33] FEISI L. Lesson 3: 3D scene modeling and simulation. <https://rflysim.com/doc/en/3/Intro.html>.
- [34] FEISI L. Lesson 4: modeling and simulation of vehicle motion. <https://rflysim.com/doc/en/4/Intro.html>.
- [35] FEISI L. Lesson 6: external control and trajectory planning. <https://rflysim.com/doc/en/6/Intro.html>.
- [36] FEISI L. Lesson 7: safety testing and health assessment. <https://rflysim.com/doc/en/7/Intro.html>.
- [37] FEISI L. Lesson 8: visual perception and obstacle avoidance decision-making. <https://rflysim.com/doc/en/8/Intro.html>.

- [38] FEISI L. Lesson 10: cluster control and game confrontation. <https://rflysim.com/doc/en/10/Intro.html>.

## Biographies



**DAI Xunhua** was born in 1990. He received his B.S., M.S., and Ph.D. degrees in control science and engineering at Beihang University, Beijing, China, in 2013, 2016, and 2020, respectively. Since 2020, he has been an associate professor with Central South University in computer science and engineering, 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 vehicle.

E-mail: dai.xh@csu.edu.cn



**TU Jinhu** was born in 1999. He received his B.S. degree in software engineering from Jiangxi University of Science and Technology in 2021 and M.S. degree in software engineering from Central South University in 2024. He is currently pursuing his Ph.D. degree in control science and engineering at Central South University. His main research interests include self-reinforcement

safety learning, health management, and low-altitude traffic of unmanned aerial vehicle.

E-mail: tjhesu@csu.edu.cn



**QUAN Quan** was born in 1981. He received his 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 with Beihang University in control science and engineering, where he is currently with the School of Automation Science and Electrical Engineering. His research

interests include reliable flight control, swarm intelligence, vision-based navigation, and health assessment.

E-mail: qq\_buaa@buaa.edu.cn