1. Background of the RflySim platform

Intelligent unmanned system is a complex system involving machinery, control, computers, communications and materials.

The technology in the field of unmanned systems is one of the key technologies required for the development of IUS, and AI is undoubtedly one of the key technologies required for the development of IUS. With the rise of a new generation of AI, we are facing a new industrial revolution, where humans have created various unmanned systems, and the technological level of unmanned systems is gradually increasing as human knowledge accumulates and develops. Autonomy and intelligence are the two most important features of intelligent unmanned systems, and the most effective way to realize and optimize these two features is to use various AI technologies, such as intelligent perception (image, speech recognition, etc.) human-computer interaction, intelligent decision-making, learning, and reasoning. Compared with traditional unmanned systems, intelligent unmanned systems have greater application potential, and the emergence of various types of intelligent unmanned systems will have a significant impact on human life and society. Currently, intelligent unmanned systems mainly include self-driving cars, drones, service-oriented robots, intelligent industrial robots, space robots, marine robots, and unmanned workshops/intelligent factories.

Intelligent unmanned systems development and testing are usually categorized into experiment-based and simulation-based. As shown in Table 1, taking UAV development as an example, although experiment-based development and testing is very direct, there are many pain points such as safety, space, time and cost, and the above pain points are even more "painful" for cluster flight testing. Simulation-based development and testing requires establishing a mathematical model of the unmanned system, developing and testing around the model, and ultimately returning to the real unmanned system. For simulation-based development and testing, the pain point is how to build a reasonable model. This leads to traditional simulation not being real and real being too expensive. However, while experiment-based development and testing is straightforward but "short-term profitable", simulation-based development and testing seems "troublesome" but "long-term profitable". For example, Tesla engineers say they spent 10 years modeling energy flow to achieve range increases without replacing battery packs. However, as far as I know, most small and medium-sized companies in China rely heavily on experiments for UAV development, and only large companies and aerospace institutes use model-based development processes for important national models.

Table 1 Comparison of experiment-based and simulation-based development and testing of UAVs

Experiment-based development and testing	**Simulation-based development and testing**
Safety pain point: rotor speed is high, the flight process is dangerous, especially for students in school.	indoors Only need computer and other equipment, low cost, small site restrictions ....

Space pain point: indoor space and inches of land, while outdoor airspace is difficult to apply for	Any fault can be simulated and automatically injected in the desired flight environment.
Time pain point: drones are often unstable, spend a huge amount of time debugging and testing, and spend most of their time debugging hardware rather than algorithms	All states, inputs, and outputs are accessible, and the truth value can be obtained at any time.
Cost pain points: high cost of hardware, debugging process often drop the machine, and hardware replacement time is rapid	This can be done at any time during the experimental development phase. The credibility of the results is difficult to guarantee and is usually used only for development and functional testing.

A typical unmanned intelligent body cluster cooperative control system architecture from simulation to experiment is shown in Fig. 1, which involves many hardware and software systems, including the design and construction of the unmanned intelligent body system, the design and construction of the communication system, the construction and design of the localization system, the construction and design of the navigation and motion control system, the construction and design of the load system, the construction and design of the mission planning system, and the construction and design of the integrated control system of the ground station, etc. It is a huge ecosystem and tool chain. It is a huge ecosystem and tool chain, including many hardware and software systems, such as navigation and motion control system, payload system, mission planning system, integrated control system of ground station, and so on.

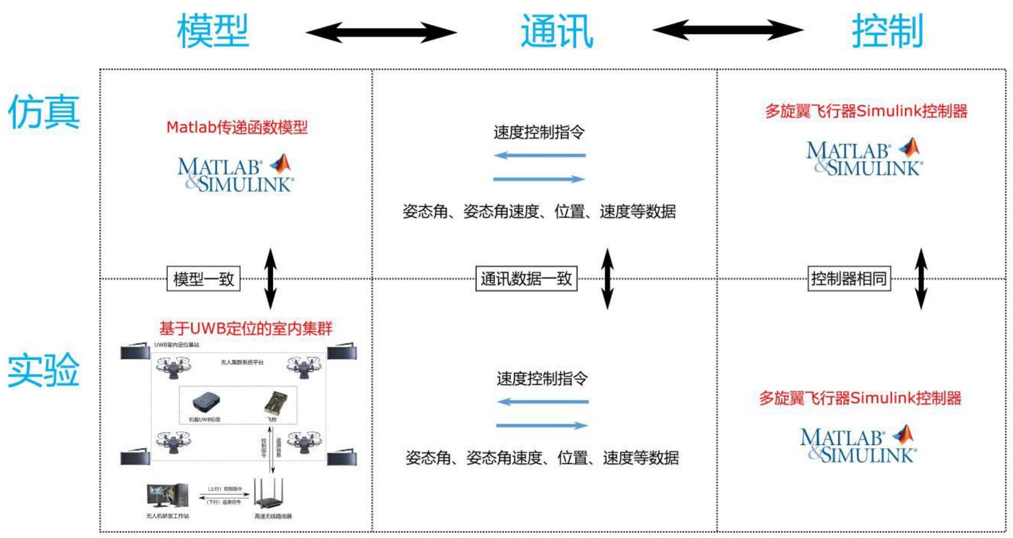


Fig. 1 Block diagram of a typical unmanned intelligent body cluster cooperative control simulation and experiment system

At present, most universities and research institutes tend to have the following difficulties in carrying out research areas related to cooperative control of multiple intelligences:

➢ The entire platform design tool chain is complex and large, and building it from scratch is time-consuming and laborious;

- Lack of systematic platform building power, and early in the research process, personnel energy is consumed in the direction of non-core research responsibilities;
- Existing decentralized hardware and software there is the use of standards, software interfaces, communication protocols are not uniform, the relevant source code is not open, learning to master and secondary development is more difficult.
- Part of the open-source platform, service support capacity is insufficient to meet the needs of local scientific research.

In the face of the above needs and shortcomings, there is an urgent need for a full-process software ecosystem or tool chain for unmanned systems development, simulation and testing.

2. RflySim Platform Features

RflySim is an ecosystem or toolchain published by the Reliable Flight Control Group at NATS. It was developed by Prof. Plenipotentiary Directed by Dr. Xunhua Dai, who led the development, and later taken over and promoted the development of advanced functions by Feisi Lab under Zhuoyi Intelligence, it is a set of highly credible unmanned control system development, testing and evaluation platform specially developed for the development of control system for unmanned platforms, large-scale cluster collaboration, AI vision and other cutting-edge research fields. The platform adopts the model-based (MBD) design concept, based on Pixhawk/PX4, MATLAB/Simulink and RO S, etc. as well as shelf intelligent hardware, etc., and can carry out (not limited to): simulation and real flight/motion of unmanned intelligent body control, simulation and real flight/motion of unmanned intelligent body clustering and simulation and real flight/motion based on the vision of unmanned intelligent body. For the research of the above problems, unmanned system modeling, controller design, Software-In-the-Loop (SIL) Hardware-In-the-Loop (HIL)can be carried out, and through the automatic code generation technology of MATLAB/Simulink, the controller can be easily and automatically downloaded into the hardware for HIL. Through the automatic code generation technology of MATLAB/Simulink, the controller can be easily and automatically downloaded into the hardware for HIL simulation and actual flight test, realizing Sim2Real. According to the characteristics of each simulation platform, as shown in Table 2, there are mainly the following characteristics of the platforms, for example:

- Harmonization. The entire research framework is extended to all unmanned control systems to form a standardized system of automated development, test and evaluation frameworks

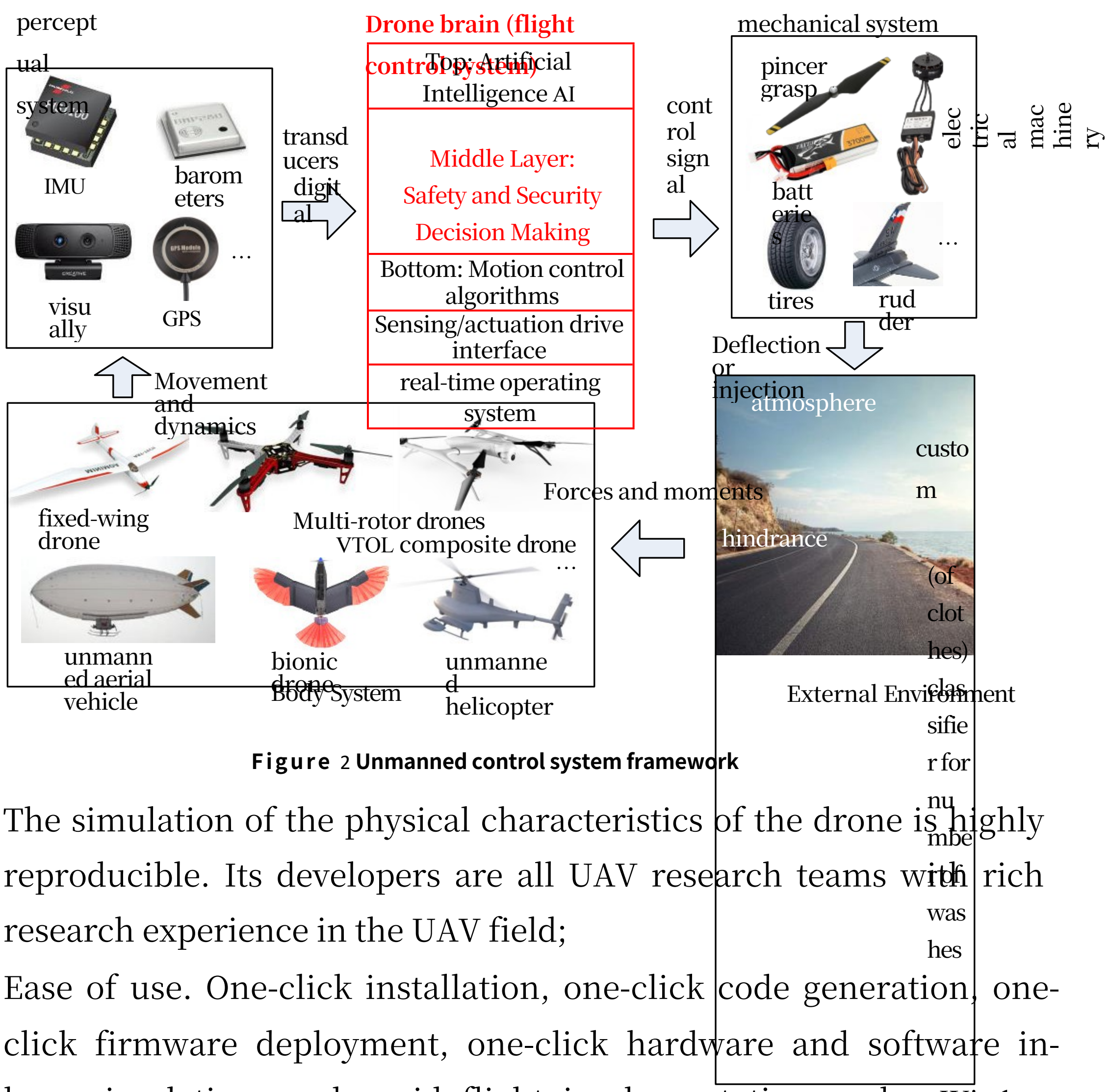
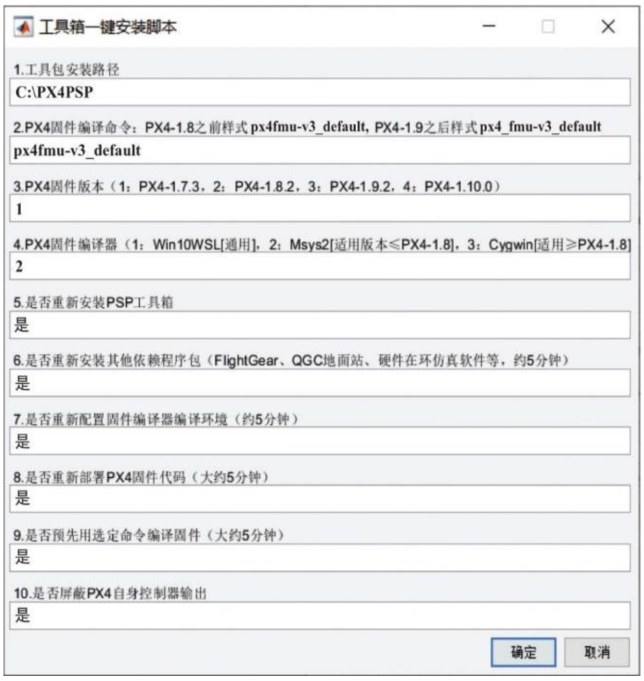


Figure 2 Unmanned control system framework

- The simulation of the physical characteristics of the drone is highly reproducible. Its developers are all UAV research teams with rich research experience in the UAV field;
- Ease of use. One-click installation, one-click code generation, one-click firmware deployment, one-click hardware and software in-loop simulation, and rapid flight implementation under Windows platform are very convenient and easy to use. Users do not need to know the source code of flight control, Linux programming, C/C++ programming, network communication, aircraft assembly and other underlying knowledge, but only need to have the basic knowledge of Simulink (or Python), you can quickly put your algorithms through the layers of validation and apply them to real aircraft, which can help you focus more on the development and testing of algorithms.


工具箱一键安装脚本

1.工具包安装路径
C:\PX4PSP

2.PX4固件编译命令：PX4-1.8之前样式px4fmu-v3_default, PX4-1.9之后样式px4_fmu-v3_default
px4fmu-v3_default

3.PX4固件版本（1：PX4-1.7.3，2：PX4-1.8.2，3：PX4-1.9.2，4：PX4-1.10.0）
1

4.PX4固件编译器（1：Win10WSL[通用]，2：Msys2[适用版本≤PX4-1.8]，3：Cygwin[适用≥PX4-1.8]
2

5.是否重新安装PSP工具箱
是

6.是否重新安装其他依赖程序包（FlightGear、QGC地面站、硬件在环仿真软件等，约5分钟）
是

7.是否重新配置固件编译器编译环境（约5分钟）
是

8.是否重新部署PX4固件代码（大约5分钟）
是

9.是否预先用选定命令编译固件（大约5分钟）
是

10.是否屏蔽PX4自身控制器输出
是

Figure 3 One-Click Installation Interface

- Fully distributed architecture. All applications can be opened on the same computer or multiple computers, and each application can send and receive messages to and from each other through the UDP network. This distributed architecture is very suitable for large-scale UAV cluster simulation tests with vision;

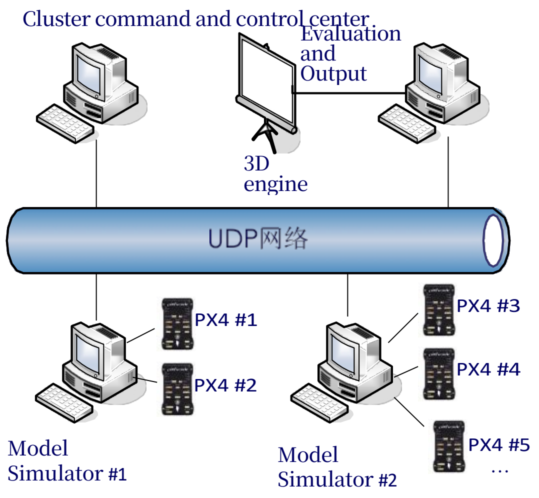


Fig. 4 Distributed simulation framework

- Support multiple aircraft types simulation. Supports a variety of aircraft such as carts, fixed-wing, vertical take-off and landing vehicles (VTOL), and so on.

Modeling. Users can build the rack model in Simulink according to the standardized interface, and then automatically generate D LL files for HIL simulation. The experimental platform can be extended to any unmanned system;

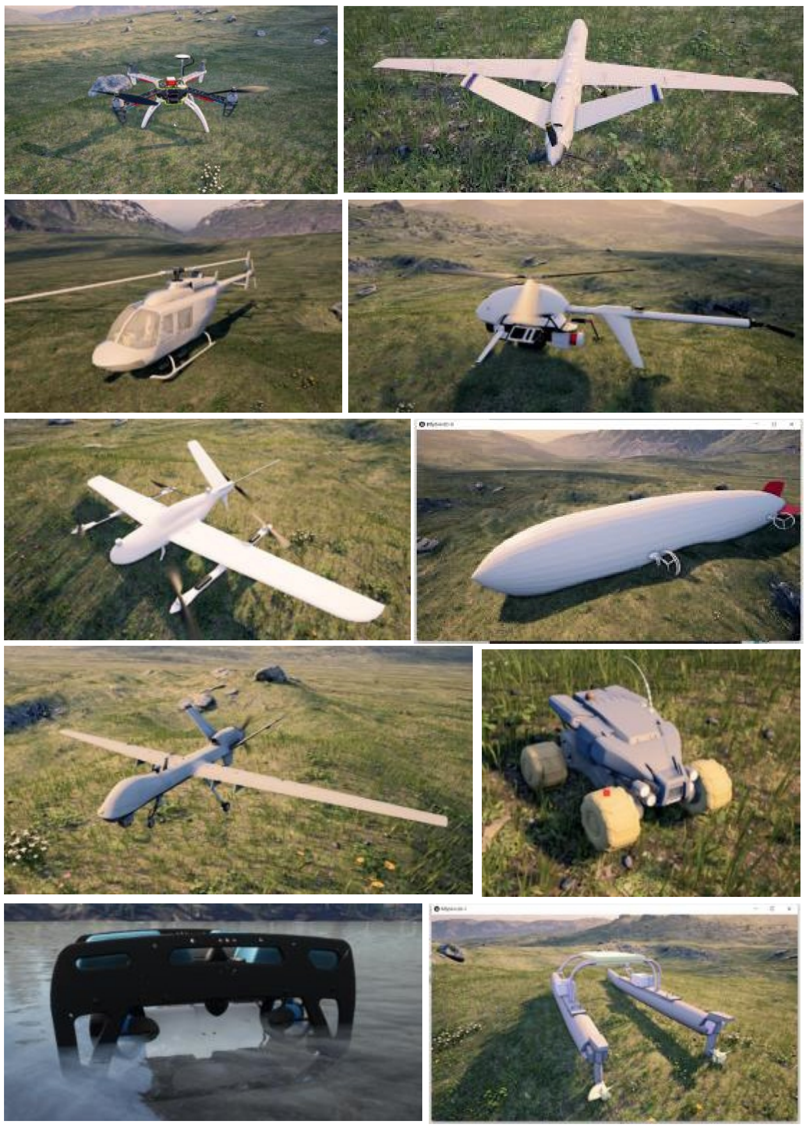


Figure 5 Multiple unmanned vehicles

- Support for UAV cluster simulation. Under the same LAN, developers can use CopterSim to connect multiple Pixhawk for hardware or software in-loop simulation. At the same time, the vehicle can be controlled using Simulink or C++ programs, and the control commands will be sent to the Pixhawk via serial (digital) or network (WIFI) using the Mavlink protocol;

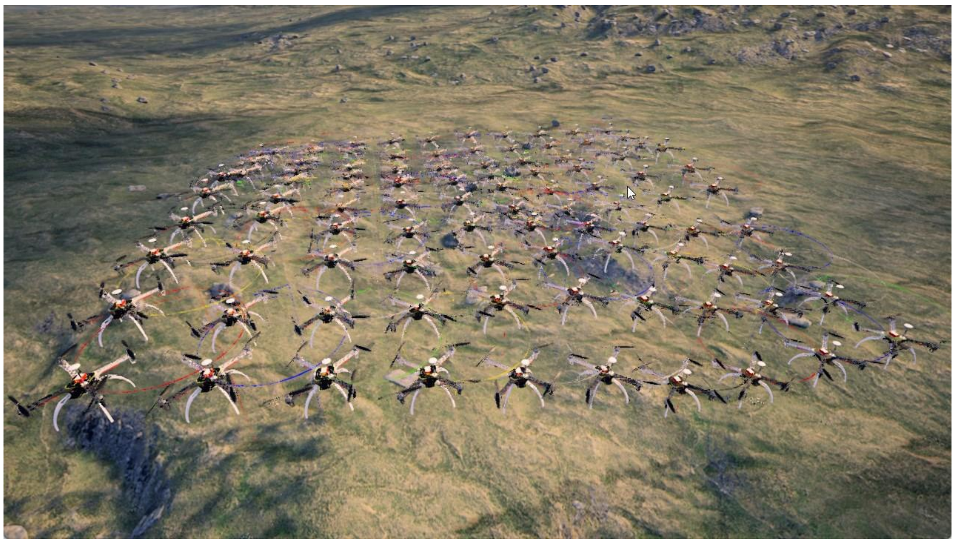


Fig. 6 Drone cluster simulation

- Provides highly realistic 3D scenes. Provide source code and tutorials to help developers build highly realistic 3D scenes in Unreal Engine (UE) for indoor and outdoor environment simulation or vision-based algorithm development; scenes support physical collision engine, global terrain and maps, OSGB+Cesium tilted photographic view maps imported, customized GPS coordinates, arbitrary multi-window

switching observation, RGB, depth, grayscale, IMU, LIDAR and other sensor data output, support for shared memory or UDP image direct sending to the specified IP address, can be used for on-board computer hardware in the loop SLAM simulation.

Fig. 7 Outdoor large scene tilt photography

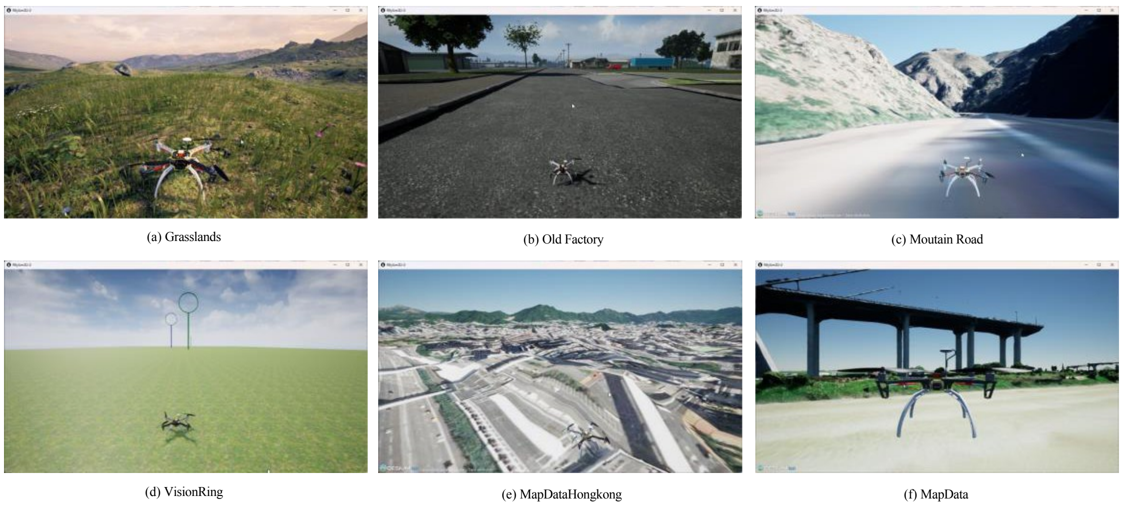
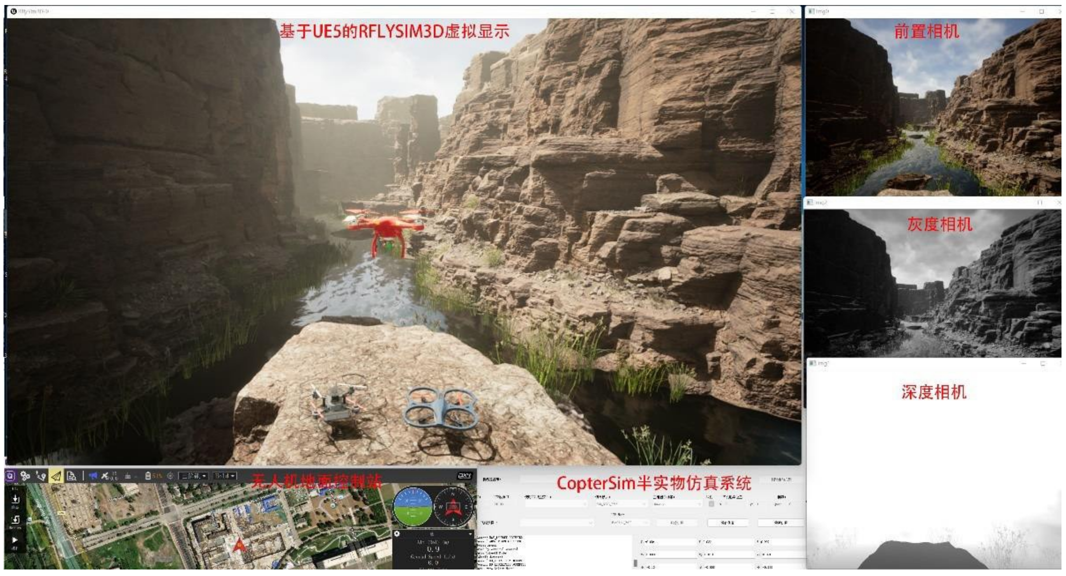


Fig. 8 **Multi-scene switching simulation**

- Supports vision-based control. The UE-based 3D vision platform also supports the viewpoint switching function, which can acquire image data from multiple viewpoints conveniently. It also supports real-time image data acquisition and processing in Simu link, Python, C/C++ and other code platforms by means of shared memory, and the processed visual data can be returned to CopterSim or Simulink control through UDP, forming a closed loop hardware-in-the-loop simulation with vision.



- Supports a wide range of fault injection, and the types of faults that can be realized include model faults, communication faults, and environmental faults.

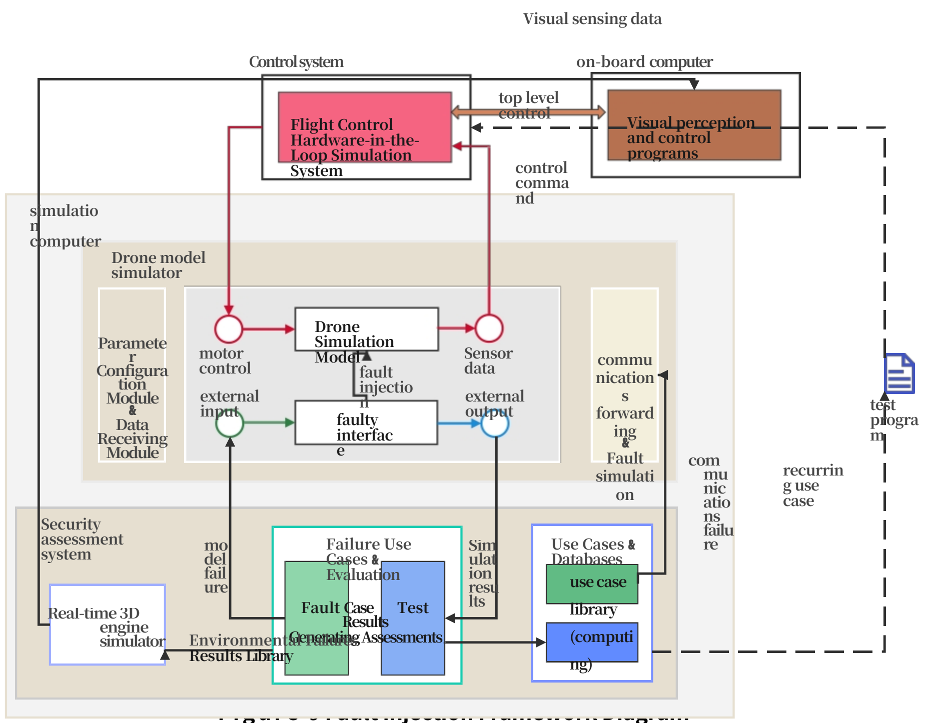


Figure 3-9 Fault Injection Framework Diagram

Also, compared to other unmanned system development and testing platforms, as shown in Table 2.

Table 2 Characteristics of each simulation platform

Simulation Platform	XPlan	FightGear	Jmavsim	Gazebo	AirSim	RflySim
Is it open source	expand one's financial resources	non-open source	expand one's financial resources	expand one's financial resources	expand one's financial resources	expand one's financial resources
Dominant Models	fixed wing	Drones, multi-rotors	quadrotor	Robotics, quadco	Automotive,	Quadrotor, fixed wing, composite wing,

				pter	quadc opter	unmanned vehicle
ROS Interface	unsupported	unsupported	be in favor of	be in favor of	be in favor of	be in favor of
Sensor Output	not have	not have	not have	there are	there are	there are
flight recorder	not have	not have	not have	not have	there are	there are
Physical touch support hit	clogged	clogged	clogged	be	be	be
Scene realism	your (honorific)	lower (one's head)	lower (one's head)	center	your (honorific)	your (honorific)
Scene richness	your (honorific)	lower (one's head)	lower (one's head)	center	your (honorific)	your (honorific)
distributed architecture	clogged	clogged	clogged	clogged	clogged	be

Hardware in ring imitation true	unsupported	unsupported	unsupported	unsupported	be in favor of	be in favor of
Physical Properties true reproduction	lower (one's head)	lower (one's head)	lower (one's head)	lower (one's head)	center	your (honorific)
Support Fault Note confirm or agree with	be	clogged	clogged	clogged	clogged	be

3. Introduction to the core components of the RflySim platform

The RflySim platform includes a wide range of tools for modeling, simulation, and algorithm validation of unmanned systems in the development process.

The core components are CopterSim, QGroundControl, and RFlySim3D. /RflySimUE5, Python38Env, Win10WSL subsystem, SITL/HITLRun one-click run scripts, MATLAB automatic code generation toolbox, Simulink cluster control interface, PX4 Firmware source code, R flySim supporting documents, and supporting hardware systems. By learning these core components, users can quickly get started with the development and testing of unmanned systems.

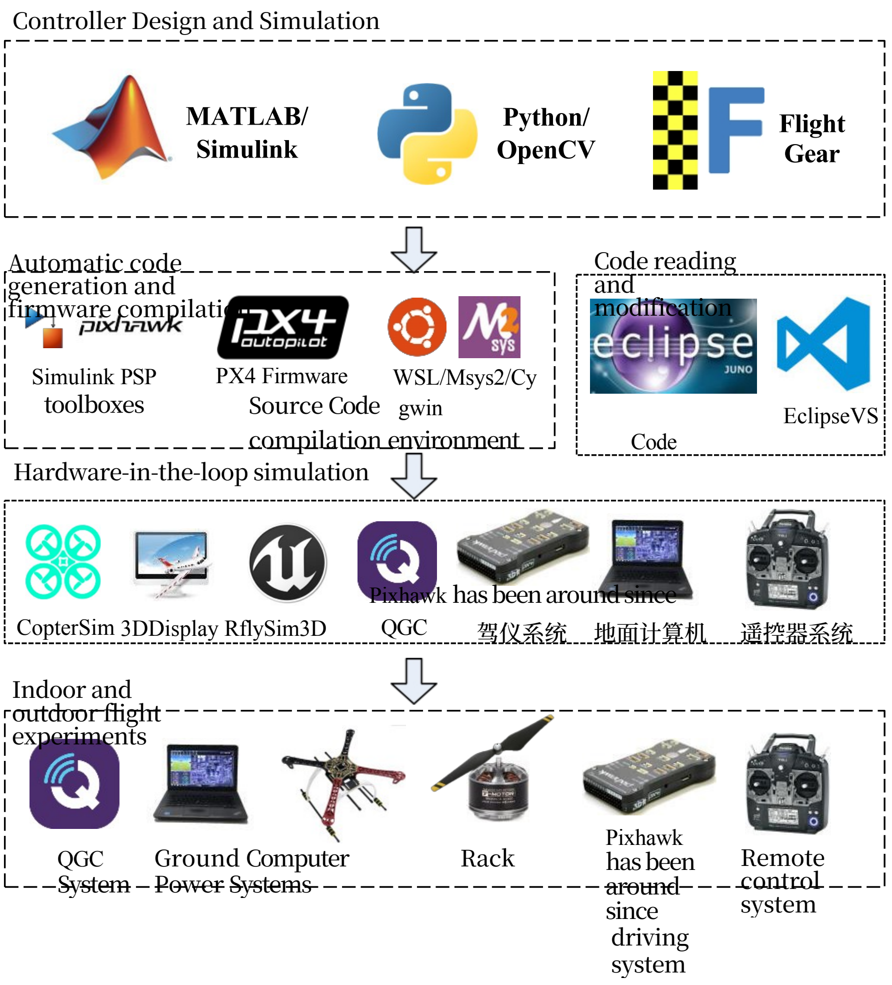

Controller Design and Simulation
MATLAB/ Simulink
Python/ OpenCV
Flight Gear
Automatic code generation and firmware compilati
Simulink PSP toolboxes
PX4 Firmware Source Code compilation environment
WSL/Msys2/Cy gwin
Code reading and modification
Code
EclipseVS
Hardware-in-the-loop simulation
CopterSim 3DDisplay RflySim3D
QGC
Pixhawk has been around since
驾仪系统
地面计算机
遥控器系统
Indoor and outdoor flight experiments
QGC System
Ground Computer Power Systems
Rack
Pixhawk has been around since driving system
Remote control system

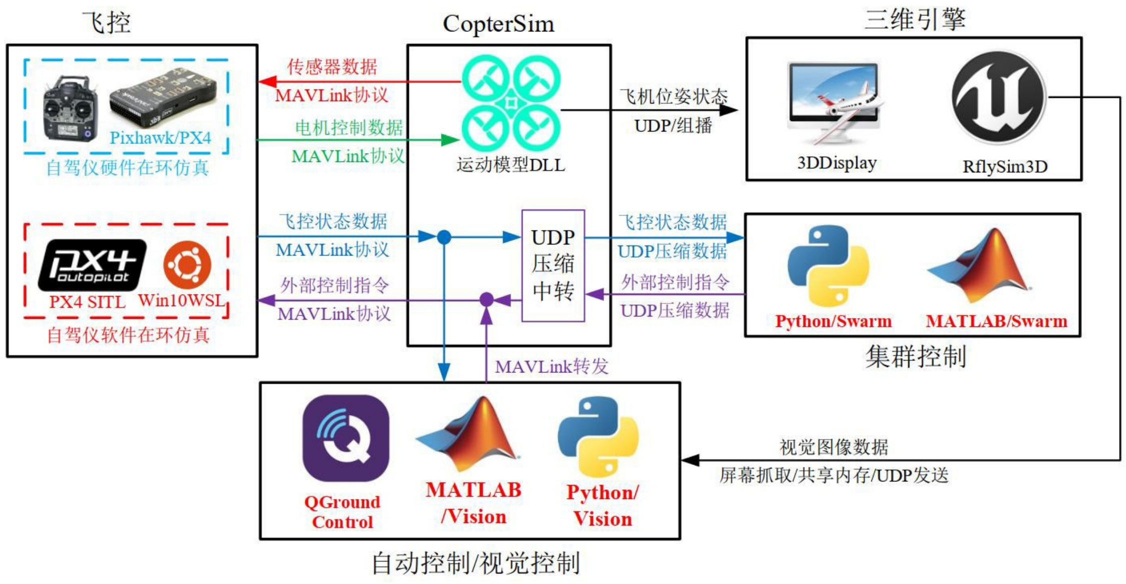
Fig. 10 **Interrelationship of RflySii hardware and software components to the overall process**

3.1 CopterSim

CopterSim is one of the core software of RflySim platform, which is a hardware-in-the-loop simulation software developed for Pixhawk/PX4 autopilot platform. You can configure a multi-rotor model in the software, and then connect it to the Pixhawk autopilot through the USB serial port to realize the hardware-in-the-loop simulation, so as to achieve the effect of indoor simulation and outdoor flight test. It consists of two main parts - model and communication. The model means that the simulation can be performed directly after calculation according to the set model parameters; it also supports the running of dynamic models (DLL), and together with other software, forms a softwaresimulation system.

/CopterSim is the center of all data communication; the flight controller and CopterSim communicate with each other through a serial port.
(hardware-in-the-loop HITL) or network TCP/UDP (software-in-the-loop SITL) to connect and use MAVLin k for data transfer, closed-loop control, and simulation of outdoor flight scenarios; CopterSim sends aircraft attitude and motor data to the 3D engine for visualization; and forwards MAVLink messages to the Python visual or Q

GC ground station, transmitting real-time status of aircraft, realizing top-level planning and control; and so on. At the same time, CopterSim software compresses the MAVLink data and sends it to the cluster control



software in the form of UDP structure, so as to achieve the purpose of communication streamlining (large-scale cluster requirements)

Fig. 11 Data communication structure of Copt@jSiii software

3.2 RflySim3D/RflySimUE5

Unreal Engine has a powerful graphics engine that supports high-quality 3D graphics and visual effects; a built-in blueprint visualization scripting system that allows developers to create complex logic and interactions graphically without writing code; a large library of community support and resources, including models, textures, sounds, plug-ins, and more, that can help developers Accelerates the development process and improves model quality; Supports multiple platforms, including PCs, consoles, mobile devices, and virtual reality devices; Developers can customize and extend the engine's features and tools according to their own needs, making the Unreal Engine suitable for a wide range of game and application development.

RflySim3D/RflySimUE5 is a high-fidelity simulation software for unmanned systems based on Unreal Engine, inheriting the various advantages of Unreal Engine, communicating with other software on the platform through UDP to realize high-fidelity simulation of unmanned systems, and transferring visual image data to QGroundControl, MATLAB, Python, and other software through screen capturing and shared memory, as shown in Fig. 12. At the same time, the visual data can be transferred to QGroundControl, MATLAB, Python and other software through screen capture, shared memory and so on, to realize the visual algorithm verification simulation of unmanned system, as shown in Fig. 12.

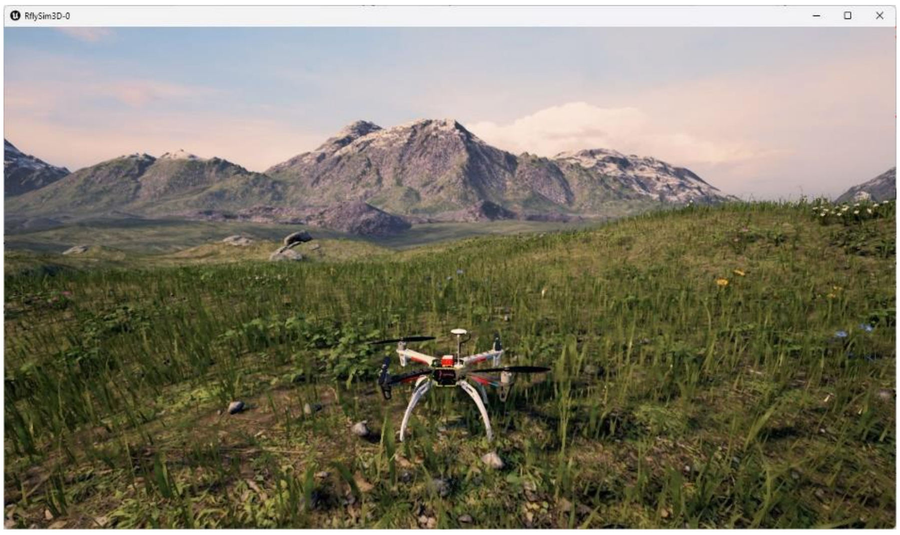


Fig. 12 RflySi¡3D/RflySi¡UE5 **Display Main Screen**

Meanwhile, for users with lower computer configurations, the RflySim platform offers two other 3D simulation software programs, FlightGear and 3DDisplay. the FlightGear development team comes from all over the world, including programmers, pilots, physicists, and experts in the field of aircraft manufacturers, and provides a variety of different types of aircraft models and scenarios, including A wide range of civil and military aircraft models, as well as many different scenarios and environment simulations. It is a very popular open-source flight simulator software that can receive the flight status sent by Simulink via UDP and conveniently observe the flight status of the aircraft during Simulink simulation. 3DDisplay is a virtual flight simulator software developed by Beihang's Reliable Flight Control Research Group, which provides three-dimensional models and virtual environments, and supports a wide range of aircraft models and scenarios. Users can freely switch between RflySim3 D/RflySimUE5, FlightGear and 3DDisplay simulation software according to the configuration of their personal computers.

3.3 QGroundControl ground station

The UAV ground station is a key component of the UAV application control system. The operator can operate the ground station through the mouse, touch screen, and remote control handle to control the UAV, and by setting the waypoint information and planning the routes on the ground station, the UAV can be made to fly according to the preset paths, and complete the waypoint tasks on the way to fly, including taking pictures, aircraft movements, and video recording, etc. Currently, the mainstream open source ground stations are QGroundC ontrol and MissionPlanner, while QGroundControl is an open source ground station specially designed for PX 4 software. Currently, the mainstream open source ground stations are QGroundC ontrol and MissionPlanner, while QGroundControl is an open source ground station specially designed for the latest architecture of PX 4 software, which uses the QT editor C++ to write its core code, and supports the source code modification and functionality of QGroundControl.

It is suitable for secondary development, i.e. for UAV ground station research experiments as well as for customization and modification of UAV ground station functions. The advantages of QGroundControl over other UAV ground stations are: 1) Open source: QGroundControl is a completely open source software, which means that users are free to modify and customize it according to their needs. 2) Ease of use: The user interface is very clear, modern and easy to use, which allows users to quickly carry out mission planning and flight scheduling. 3) Multi-tasking: The user interface is very clear, modern and easy to use.
Platform Support: QGroundControl can run on a variety of operating systems, such as Windows, Linux and M acOS, etc. 4) Modular Architecture: QGroundControl's modular architecture makes it easy for developers to add and expand new functionality without affecting existing functionality and performance. Overall, QGroundCo ntrol is a modern, easy-to-use, open-source and highly customizable ground station software, which has obvious advantages in terms of multi-platform support, multi-language support and modular architecture.

3.4 Python38Env

Python is a high-level, object-oriented interpreted programming language. Originally created by Guido van R ossum in 1989, it is now a popular programming language used for developing web applications, data analysis, artificial intelligence, scientific computing, network programming, and more.Python is an easy-to-learn, easy-to-read, and easy-to-write language, and as such, it is also widely used for instructional and introductory level programming.

Python38Env is a virtual environment for Python 3.8 programming language, including numpy, py mavlink, OpenCV, pyulog and other libraries, which can be used to quickly develop algorithms related to unmanned systems without the need for the user to deploy the python runtime environment and various functional libraries.

3.5 MATLAB Automatic Code Generation Toolkit

The MATLAB Automatic Code Generation Toolkit is an extended toolkit for MATLAB to generate various forms of executables such as C code, executables, static libraries and dynamic libraries from Sim ulink models. These executables can be run directly on embedded platforms without the need for manual writing and debugging. A wide range of embedded platforms are supported, including the ARM Cortex-M and A series processors, the NXP MPC55 xx and MPC56xx series, the Pixhawk series, and more.

The module library contains GPS data module, battery data module, uORB module and many other modules. Based on RflySim and Pixhawk Support Package platform, users can: (1) design and simulate control algorithms in Simulink; (2) automatically generate C code and PX4 firmware from Simulink models, and transfer them directly to PX4 firmware.

(i) Burn to the Pixhawk board; (ii) Configure and calibrate the Pixhawk board and its peripherals using MATLAB scripts and functions; (iii) Read and write data to and from the Pixhawk board in real time; and so on.

3.6 SITL/HITL batch scripts

Batch processing is a technique whereby a computer can process a collection of tasks in groups and the entire process is fully automated without human intervention, which can also be referred to as workload automation (WLA) and job scheduling. This can also be referred to as workload automation (WLA) and job scheduling. It offers the advantages of speed and cost savings, accuracy, and simplicity.

RflySim has developed numerous batch scripts based on batch processing technology, allowing users to quickly launch and deploy multiple unmanned systems, multiple and varied unmanned systems combinatorial simulations with a single click. This improves the speed of unmanned system development and simulation. The more commonly used batch scripts on the platform are: ① SITLRun.bat: it is a batch file to open the software in-loop simulation of multiple aircrafts, which is essentially a script to start and configure some of the software and options of the RflySim platform, ② HITLRun.bat: it is a batch file to open the hardware in-loop simulation of multiple aircrafts, and then after inserting multiple flight controllers, you can double-click on the batch file and enter the Pilot Programs you want to participate in the simulation according to the prompts. After inserting multiple flight controllers, double-click the batch file and enter the serial port number of the Pixhawk you want to participate in the simulation according to the prompts. In addition, the RflySim platform also provides many batch script files, such as SITLRunPos.bat, SITLRunLowGPU.bat, SITL

RunMAVLink.bat, HITLRunPos.bat, HITLPosSysID.bat, HITLPosStr.bat, and so on.

Users can open these files through the editor and modify the parameters therein according to their personal needs to realize custom development and quickly start simulation or verification of algorithms.

3.7 PX4 Firmware Source Code

PX4 is evolved from PIXHAWK, a software and hardware project of the Computer Vision and Geometry Laboratory of ETH Zurich, Switzerland. The flight control system is completely open source, providing a low-cost and high-performance high-end autopilot for flight control enthusiasts and research teams around the world. After years of development and refinement by world-class developers from industry and academia, the PX4 flight control system has formed a perfect and reasonable software architecture, with the Pixhawk series of self-pilot hardware platform, constituting the Pixhawk PX4 self-pilot software and hardware platform, can control multi-rotor, fixed-wing, airships and other carriers, is the world's widely used open source drone It is a widely used open source UAV hardware and software platform in the world.

The RflySim platform supports one-click deployment of PX4 compilation environments, which can be customized to select different PX4 firmware compilers.

translating commands and firmware versions, the platform will deploy the selected PX4 Firmware source code on the set installation path, and if the firmware exists, it will delete the old firmware folder and make a new deployment, which greatly improves the efficiency of PX4 environment deployment.

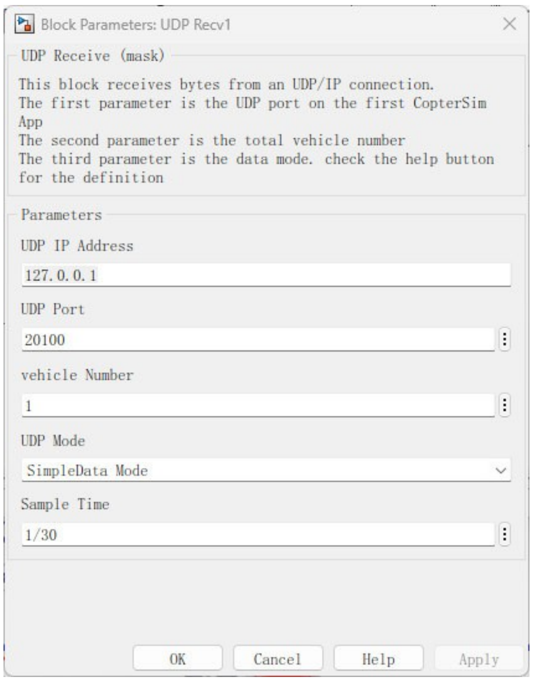
3.8 Win10WSL Subsystem

The Win10WSL subsystem is a subsystem on a Windows operating system that allows users to run Linux applications, use the Linux command line interface (CLI) and install Linux distributions on Window s. The Linux system installed on the RflySim platform with one-click is Ubuntu18.04.5, which is mainly used for PX4 source code compilation.

The platform also provides two other compilation environments to realize the simulation of Linux compilation environment under Windows platform, namely: Msys2Toolchain compilation environment based on Msys2 and CygwinT oolchain compiler based on Cygwin. Users can choose different compilation environments according to their own PX4 version, and switch between different compilation environments by selecting different options in the one-click deployment and installation interface.

3.9 Simulink Cluster Control Interface

RflySim platform has developed a cluster control interface based on Simulink S function, as shown in Fig. 13, which is realized by Simulink S function through C++ mixing, together with the advantages of Simulink's own UDP module, which has the advantages of high efficiency, small arithmetic, low latency, more reliable, and strong extensibility, etc. Users can load this module into their control system by copying and pasting. Users can load this module into their own control system by copying and pasting, which can help users quickly realize the development of unmanned system cluster control.



UDP Receive (mask)

This block receives bytes from an UDP/IP connection.
The first parameter is the UDP port on the first CopterSim App
The second parameter is the total vehicle number
The third parameter is the data mode. check the help button for the definition

Parameters

UDP IP Address: 127.0.0.1

UDP Port: 20100

vehicle Number: 1

UDP Mode: SimpleData Mode

Sample Time: 1/30

Figure 13 Cluster Control Interface

3.10 RflySim Companion Document

RflySim platform provides very perfect learning materials and routine files, through PPT courseware materials and Rf lySimAPIs routine files, so that users can build and develop their own unmanned systems in a one-stop way through gradual and progressive learning, from unmanned system bottom layer control algorithm → middle layer decision-making algorithm → top layer learning algorithm development and verification.

3.11 Supporting Hardware Systems

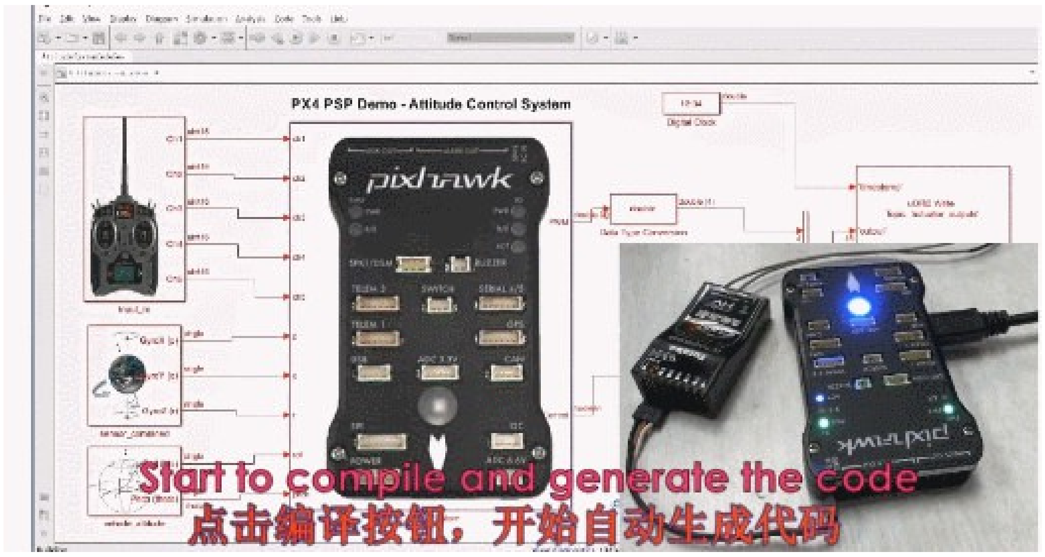
RflySim platform provides a complete set of supporting hardware system, including quadcopter UAV, flight control, remote control and other components. These components are perfectly compatible with the platform, and can be used to realize software and hardware in-loop simulation experiments in the RflySim platform, and to realize the flight of the UAV in the real environment based on the generated firmware.

At present, the supported aircrafts are FeiSi X150, FeiSi X200, FeiSi X450 and other quadcopters, among which FeiSi X150 is a newly designed micro quadcopter for indoor cluster control and scientific research, and the supported flight controllers are Pixhawk series, Joywing racer, Joywing H7, MindPX, MindRacer and so on, and the Pixhawk series is currently supported. The Pixhawk series currently supports Pixhawk1, Pixhawk 4, Pixhawk 6X, and will support Pixhawk 1 for a long time, Pixhawk 6X, ZhuoYi H7 three kinds of flight control. Supported remote controllers are Lodi AT9S Pro, Skyflight ET07, Fosse i6s, Futuba T14SG and so on.

4. Introduction to the main features of the RflySim platform

4.1 Flight Control Bottom-Level Development Functions

RflySim adopts the idea of Model-Based Design (MBD), which can be used for control and safety testing of unmanned systems. It goes through the following five phases: modeling phase, controller design phase, Software-In-the-Loop (SIL) simulation phase, Hardware-In-the-Loop (HIL) simulation phase, and real flight test phase. The controller can be easily and automatically downloaded into the hardware for HIL simulation and real flight test through the automatic code generation technique of MATLAB/Simulink.


PX4 PSP Demo - Attitude Control System
pixhawk
Start to compile and generate the code
点击编译按钮，开始自动生成代码

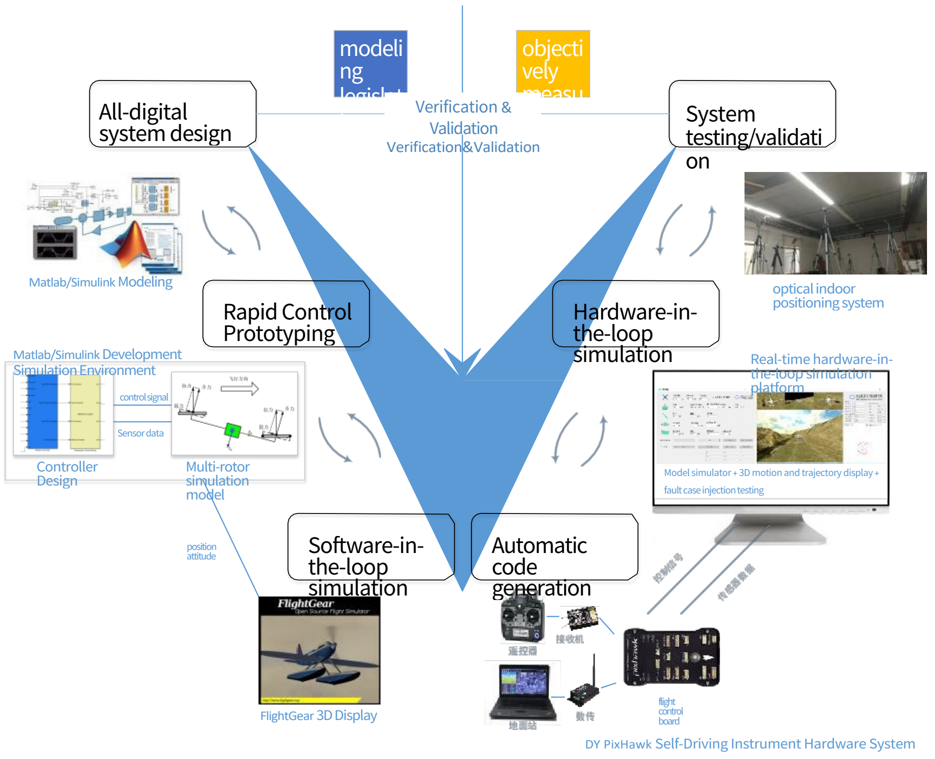

modeling legislat
objectively measu
All-digital system design
Verification & Validation
Verification&Validation
System testing/validation
Matlab/Simulink Modeling
optical indoor positioning system
Rapid Control Prototyping
Hardware-in-the-loop simulation
Matlab/Simulink Development Simulation Environment
control signal
Sensor data
Controller Design
Multi-rotor simulation model
Real-time hardware-in-the-loop simulation platform
Model simulator + 3D motion and trajectory display + fault case injection testing
position attitude
Software-in-the-loop simulation
Automatic code generation
控制信号
传感器数据
FlightGear
遥控器
接收机
地面站
数传
flight control board
FlightGear 3D Display
DY PixHawk Self-Driving Instrument Hardware System

Fig. 14 RflySim Model-based design framework

MATLAB/Simulink provides an interface for the design of multi-rotor controllers, so that users (beginners, students or engineers) can utilize their own knowledge to quickly design and verify the controllers. After the controller design is completed, the platform provides code generation and download function, which can generate C/C++ code from the designed Simulink control algorithm, and then compile it into the firmware of PX4 autopilot and download it into the autopilot automatically. The platform also provides hardware-in-the-loop simulation test function, which allows users to conduct preliminary simulation tests on the real Pixhawk self-pilot system to eliminate various problems that may exist in the real flight experiment. After passing the test, the Pixhawk autopilot can be mounted on a multi-rotor hardware system to conduct indoor and outdoor flight experiments to evaluate the performance of the designed control algorithms through experimental validation.

4.2 Unified Modeling Framework for Unmanned Vehicle Systems

The Unified Modeling Framework for Unmanned Vehicle Systems (UVS) decomposes the entire UVS into two parts: the airframe system and the control system. Sensor data and control signals are carried out between the airframe system and the control system. The airframe system, in turn, can be subdivided into four subsystems: the airframe subsystem, the actuator subsystem, the 3D environment subsystem, and the sensor data and control signals.

Sensor subsystem.

• The body subsystem contains internal subsystem modules such as the body, operating environment, forces and moments, and is an overall description of the motion, energy consumption and failure characteristics of the body in its environment;

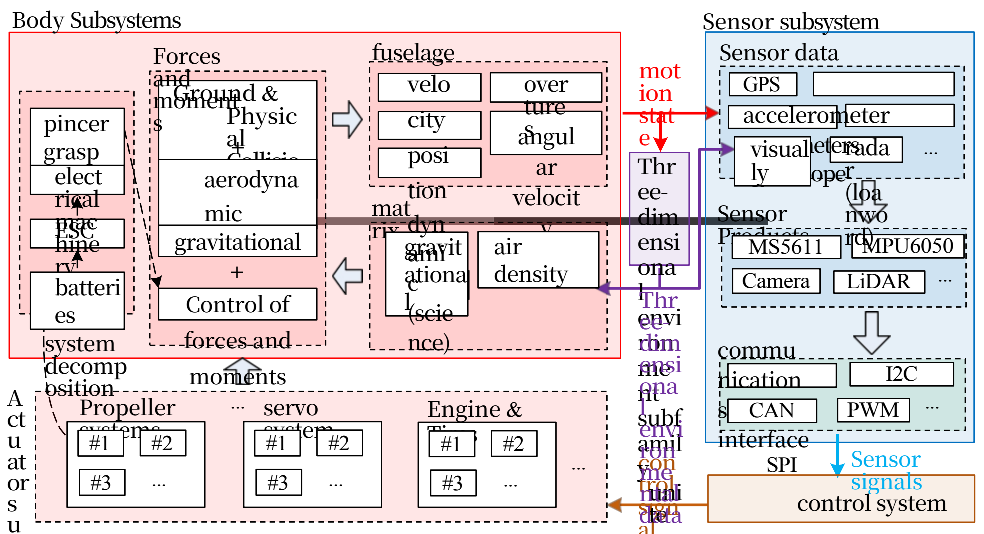
• The actuator subsystem encompasses the interaction of the vehicle with the external environment, which receives control signals from the control system and then generates forces and moments to drive the airframe motion;

• The sensor subsystem is mainly used to describe all electronic hardware models other than the control software, and mainly contains characteristics such as sensor data, communication protocols, and connection interfaces;

• The 3D environment subsystem is mainly used to describe the 3D view environment (including trees, obstacles, roads, etc.)of the unmanned flight, which is used to provide visual data for the autonomous control system to simulate.

In the whole modeling framework, the airframe system needs to be modeled with high accuracy and in real-time simulation computer

It is realized in the control system software or hardware, and finally connected to the control system software or hardware, constituting a software-in-the-loop simulation or hardware-in-the-loop simulation closed loop.



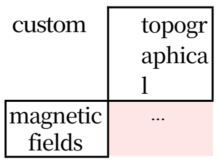


Figure 15 **Unified Modeling Framework for Unmanned Vehicle Systems**

The above modeling framework can be quickly implemented in Simulink and other visual modeling and simulation software. After the whole simulation model is built, the automatic code generation method can be used to generate simulation software under different real-time simulation computer environments, and different types of vehicle models can be quickly extended by replacing specific subsystem modules. This is an example of building a multi-rotor simulation model in Simulink, which contains basic

The motion simulation function and fault injection function can simulate the motion dynamics of various multi-rotors very realistically. At the same time, RFlySim provides Simulink unmanned dynamics modeling templates, which support a variety of rotor types, making it easy to simulate the dynamics of various rotors.

Provide standard input/output interfaces to build unmanned models of any configuration in Simulink; support the automatic code generation of Simulink as DLL files to be imported into hardware-in-the-loop simulators; the database of powertrain components covers more than 2,000 components on the market, and supports the selection of suitable motors, propellers, and other components for assembly from the database. Different configurations of multi-rotors (three, four, six, eight rotors) and estimate the performance (hovering time, maximum pull force, etc.) and model parameters (mass, moment of inertia, propeller pull force coefficient, etc.)for multi-rotor dynamics simulation.

4.3 Unmanned Systems Simulation Scenario Development Function

Based on Unreal Engine real-time 3D rendering engine, RflySim has developed RflySim3D/RflySimUE5, an unmanned real-time vision software, which can receive all the aircraft data in the local area network and centrally display it in real-time in the scene, and also supports individual configuration and acquisition of specific on-board cameras for specific aircraft. The software also supports designing and importing your own 3D scenery and aircraft models with very simple operation. It supports to use the huge amount of 3D scenes and aircraft models in the UE4 scene library directly, and it also supports to use 3Ds Max and AutoCAD to create 3D scenes and aircraft models by yourself and import them into

UE4. The following pictures show the 3D scenes of quadcopter indoor flight, car driving in the neighborhood, six-rotor forest shuttling, and fixed-wing mountain range cruising that the platform comes with at the moment, respectively.

Front camera view
Vehicle chase view
Fixed monitoring view
Car chase view
室内场景
Push throttle to take off
小车室外场景
Arm the drone and take off
多旋翼森林场景
固定翼山脉场景

Fig. 16 **Example of a Highly Realistic 3D Scene**

At the same time, the typical scene building design module can customize the scene type, provide typical scene module design SDK, and customize the environment, facilities and pedestrians in the scene. It supports dynamically changing the scene map via UDP/ROS, changing the 3D style of aircraft, dynamically creating obstacles (other aircraft, tracking targets, people, calibration boards, tables and chairs, etc.) dynamically changing the viewpoint of the aircraft (position, direction, focus, etc.) and changing the resolution of the output image of the 3D engine.

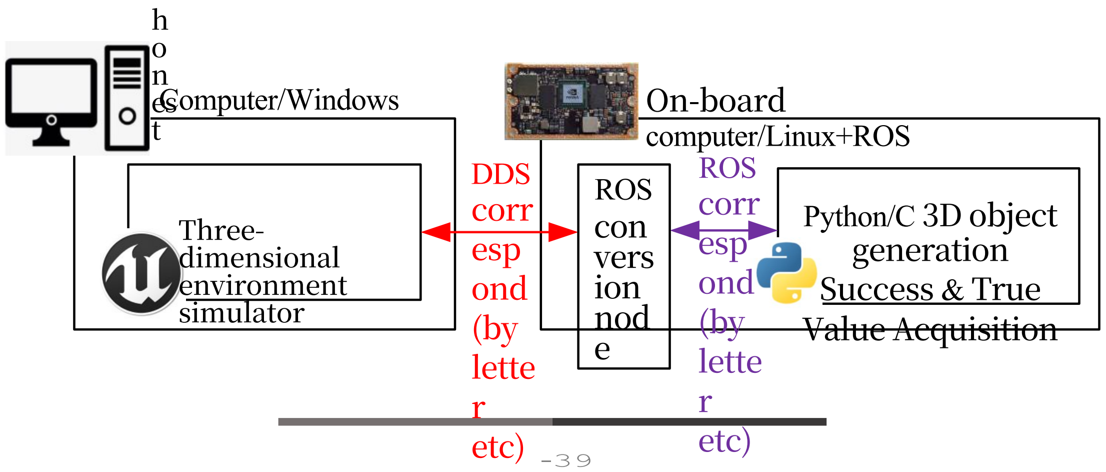
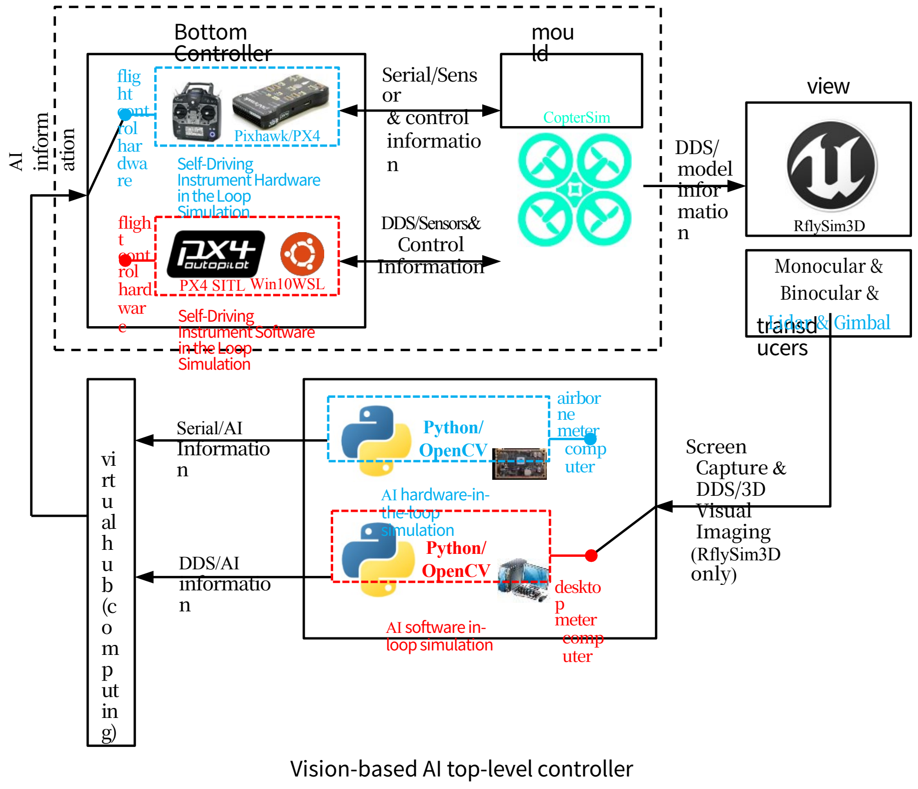

honest Computer/Windows
Three-dimensional environment simulator
DDS correspond (by letter etc)
ROS conversion node
On-board computer/Linux+ROS
ROS correspond (by letter etc)
Python/C 3D object generation Success & True Value Acquisition

Fig. 17 RflySi¡3D/RflySi¡UE5 **Communication Framework**

4.4 Unmanned Systems Visual Control Development Function

Visual control of unmanned systems is mainly based on the perception of the unmanned environment through network communication nodes that receive visual

The visual sensor data of the unmanned sent by the perception module, using image algorithms or artificial intelligence to complete the navigation or situational awareness of the vision. The top-level control module completes the upper-level decision-making of the unmanned by synthesizing the command information of the manipulator, the state information of the unmanned (attitude, speed, position, power, health, etc.) and the result of the visual perception, and sends the bottom-level control instructions to the unmanned self-pilot.


Bottom Controller
mou ld
view
AI information
flight control hardware
Pixhawk/PX4
Self-Driving Instrument Hardware in the Loop Simulation
Serial/Sensor & control information
CopterSim
DDS/ model information
RflySim3D
flight control hardware
PX4 SITL Win10WSL
Self-Driving Instrument Software in the Loop Simulation
DDS/Sensors& Control Information
Monocular & Binocular & Lidar & Gimbal
transducers
virtual hub (computing)
Serial/AI Information
Python/ OpenCV
airborne meter computer
AI hardware-in-the-loop simulation
DDS/AI information
Python/ OpenCV
desktop meter computer
AI software in-loop simulation
Screen Capture & DDS/3D Visual Imaging (RflySim3D only)
Vision-based AI top-level controller

F i g . 18 Communication framework of RflySii platform vision module

1. Run HilScript and input Com number
1.输入HIL脚本并输入飞控串口号

F i g u r e 19 RflySi i platform visual loop-throughs

The RflySim platform supports the access of external sensors, which are divided into two categories: external sensors directly connected to the flight control (magnetic compass, differential GPS, optical flow velocimetry, etc.) and visual sensors directly connected to the on-board computer (binoculars, Lidar, depth camera, etc.). The flight control sensors are generated by Simulink and other programs and transmitted directly to the Pixhawk flight control. The visual sensors are generated by the 3D environment engine and transmitted to the on-board computer along with the images, as shown in Figure 20. RflySim provides the SDK of the depth camera sensor module with the basic parameters of the sensors and the mounting positions, so that the user can customize the on-board vision module by setting the relevant parameters of the UAV-mounted visual sensors to design the orientation, the focal length, the field-of-view, and so on; and provide the decision-making system input/output according to the decision-making input/output interface protocols. According to the input/output protocol of decision-making, it provides input/output interface of decision-making system.

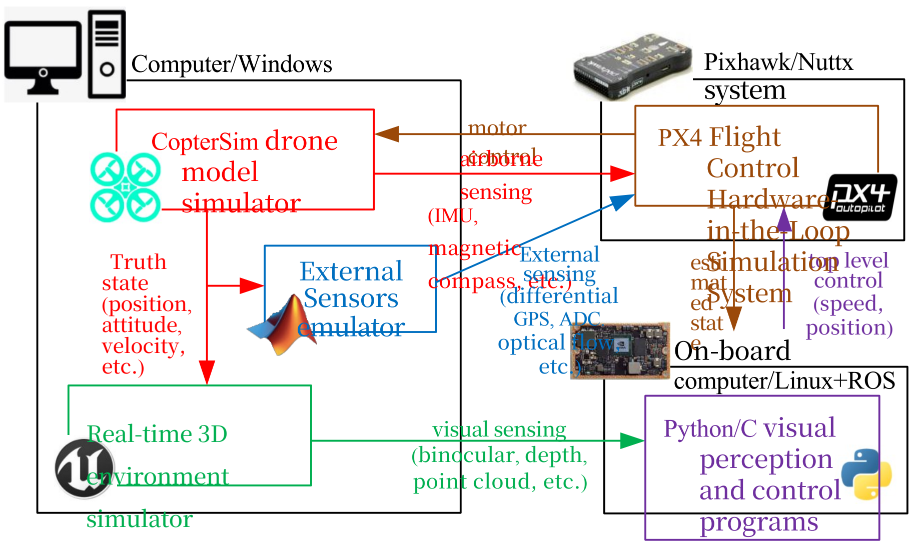

Computer/Windows
Pixhawk/Nuttx system
CopterSim drone model simulator
motor control
airborne sensing (IMU, magnetic compass, etc.)
PX4 Flight Control Hardware-in-the-Loop Simulation System
External Sensors emulator
External sensing (differential GPS, ADC, optical flow, etc.)
Truth state (position, attitude, velocity, etc.)
estimated state
top level control (speed, position)
On-board computer/Linux+ROS
Real-time 3D environment simulator
visual sensing (binocular, depth, point cloud, etc.)
Python/C visual perception and control programs

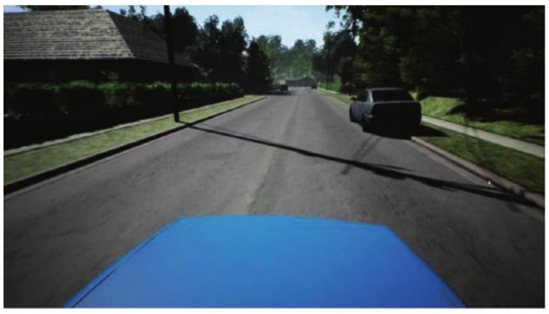
Fig. 20 RflySim platform vision module simulation framework

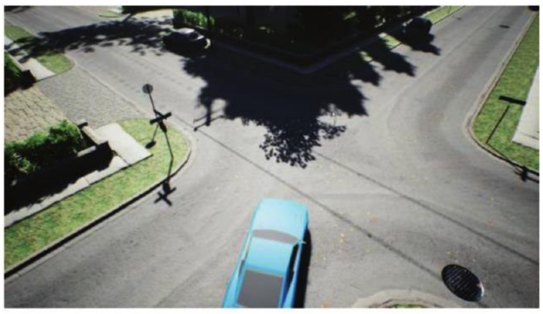
The RflySim3D-based 3D visualization platform supports viewpoint switching, which makes it easy to acquire image data from multiple viewpoints. It also supports real-time image data acquisition and processing in Simulink, Python, C/C++ and other code platforms by means of shared memory or window image capture, and the processed visual data can be returned to CopterSim or Simulink via UDP to form a closed loop hardware-in-the-loop simulation with vision.

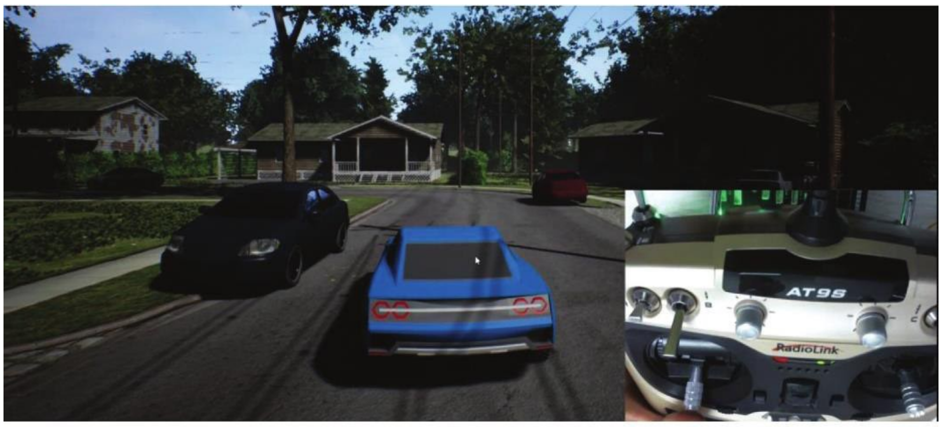
- Adopting Python/C/Simulink interface to read RflySim3D window images directly, reducing the intermediate links, the 720P multi-window image reading consumes less than 5ms (200Hz or more) and does not interfere with the operation of RflySim3D 3D simulation program.
- Supports opening any RflySim3D window, each window can be independently configured to display the viewpoint (simulator)

(on-board camera or ground observation view, etc.)

- ➢ Supports adjustments via keyboard shortcuts, as well as sending commands via UDP to control RflySim3D's viewpoint/on-board camera display parameters.
- ➢ The control interface is provided to send and receive Mavlink data directly at the bottom, and since it uses the cross-platform Python language, it can be used by copying directly to the on-board computer(A Simulink visualization interface will be provided later to support code generation.)


(a) 前视摄像头视角


(b) 道路监控视角



(c) 用于实时硬件在环的跟随视角

4.5 Unmanned Systems Cluster Control Development Function

RflySim supports one-key start multi-aircraft cluster simulation function, supports MATLAB/Simulink, Python end cluster simulation development, supports multiple software-in-the-loop, hardware-in-the-loop, and software-hardware combination of virtual and real cluster simulation, and supports distributed cluster simulation of multiple computers in LAN. At the same time, with the increase of the number of aircraft, the network communication load is getting bigger and bigger, in order to realize more number of UAV cluster simulation under the limited bandwidth, it is necessary to optimize the communication, at present, there are two main data protocols in the platform: MAVLink data and UDP compressed structure, based on these two data protocols, RflySim proposes 5 kinds of compressed data protocols to realize hundreds of

A drone cluster simulation for racks.

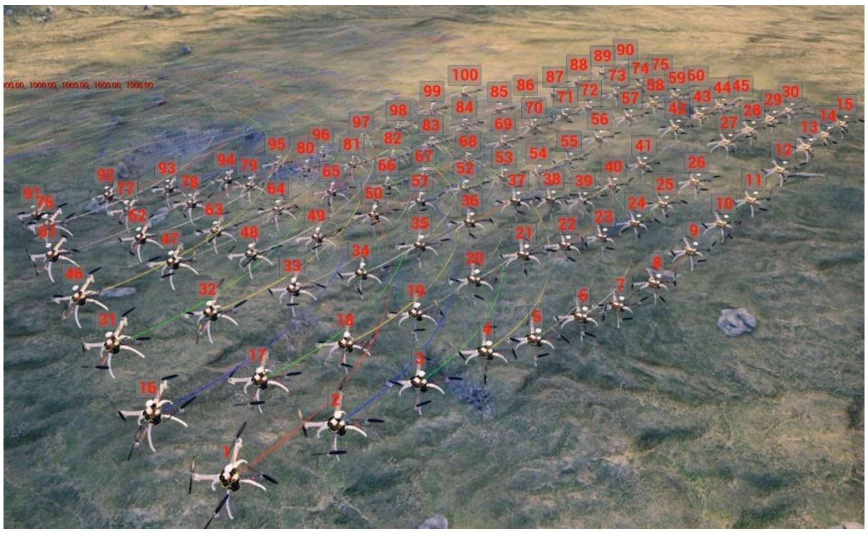


Figure 21 Large-scale UAV cluster simulation

Figure 22 RflySii platform cluster hardware-in-the-loop simulation

At the same time, the system can realize the functions of large-scale UAV cluster simulation, distributed UAV cluster control research, distributed UAV cluster visual perception algorithm verification, UAV flight control and communication algorithm verification with the help of large-scale cluster distributed control simulation system platform. The system framework is shown in Figure 23.

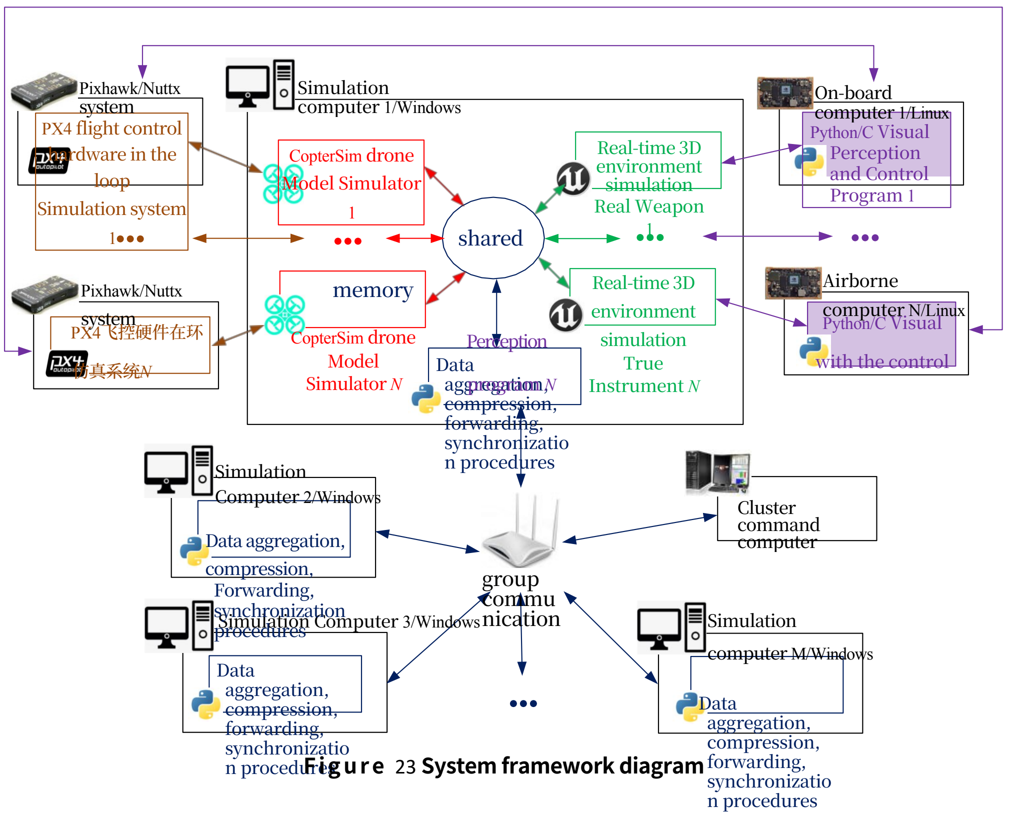

Pixhawk/Nuttx system
PX4 flight control hardware in the loop
Simulation system 1
Pixhawk/Nuttx system
PX4 飞控硬件在环
仿真系统N
Simulation computer 1/Windows
CopterSim drone Model Simulator 1
CopterSim drone Model Simulator N
shared memory
Real-time 3D environment simulation Real Weapon
Real-time 3D environment simulation True Instrument N
Perception
Data aggregation, compression, forwarding, synchronization procedures
On-board computer 1/Linux
Python/C Visual Perception and Control Program 1
Airborne computer N/Linux
Python/C Visual with the control
Simulation Computer 2/Windows
Data aggregation, compression, Forwarding, synchronization procedures
Simulation Computer 3/Windows
Data aggregation, compression, forwarding, synchronization procedures
group communication
Cluster command computer
Simulation computer M/Windows
Data aggregation, compression, forwarding, synchronization procedures

Figure 23 System framework diagram

4.6 Unmanned cooperative control decision-making algorithm design function

RflySim's unmanned cooperative control decision-making algorithm design module has a large-scale unmanned cluster control system with a top cluster planning and command layer, a networking communication node layer, a cluster decision-making layer, and a bottom control layer.

1) Supporting centralized unmanned cluster control strategy, all unmanned data can be accessed in the command center and unified trajectory planning and security protection;

2) Supporting a distributed unmanned cluster control strategy, each aircraft can acquire information about the neighboring unmanned and make autonomous decisions;

3) Supports software/hardware in-loop simulation verification of

centralized control algorithms;

4) Support multiple software/hardware in-the-loop simulation systems networked on multiple computers in the LAN to form an overall cluster simulation system and realize centralized display and control.

5) Supports rapid porting of cluster control algorithms to real aircraft flight platforms for test flight testing;

4.7 Distributed Network Communication Module

RflySim utilizes a distributed networking architecture, where different simulation models can run on the same computer or on different computers. Opening multiple model simulators and connecting multiple Pixhawk/PX4 autopilot hardware creates a multi-unmanned cluster simulation environment. Since the performance of a single computer is limited, the overall number of airplanes can be further expanded by having multiple computers communicate with each other over a local area network (LAN).

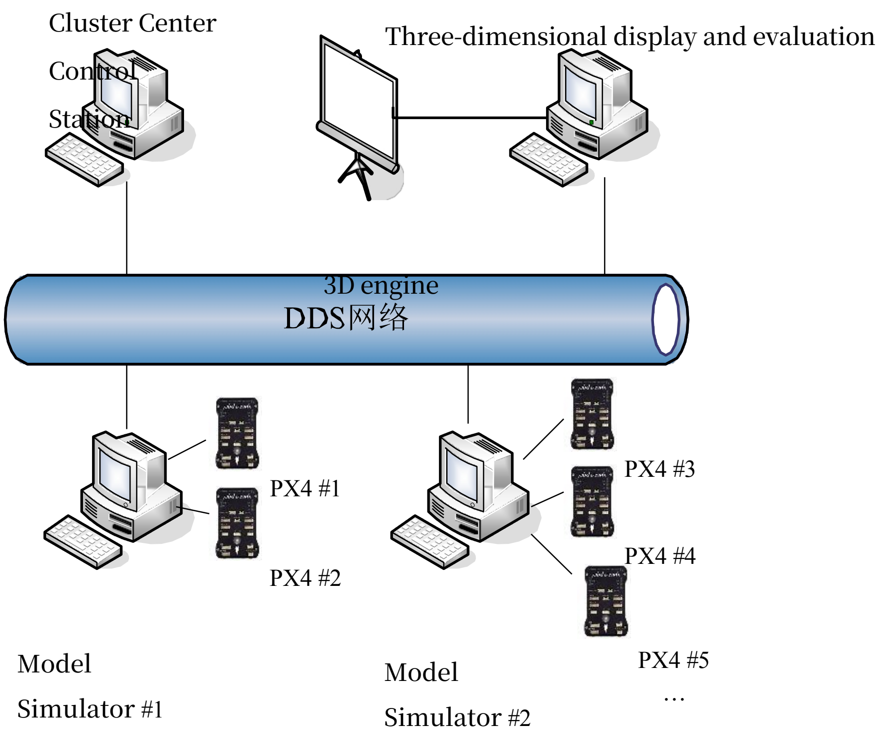
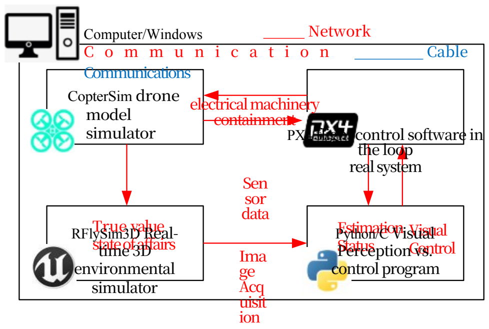


Fig. 24 Multi-unmanned cluster model simulation topology

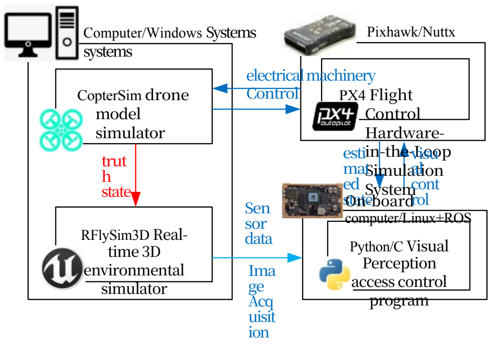
5. Preview of Future Features of the RflySim Platform

5.1 ROS-compatible top-level vision/decision algorithm development

The RflySim platform can develop and test a single vision function under the local computer, and the whole development process is carried out under Windows, which is characterized by ease of use, high efficiency, low entry threshold and low cost. After completing the pure software-in-the-loop simulation on the local computer, the Python/C visual perception and control program can be directly deployed to Linux/ROS on the onboard computer, and the flight control firmware can be used to replace the software-in-the-loop simulation mode to realize the migration of algorithms. Further, the PWM output of the Pixhawk/PX4 flight controller can be directly plugged into the rack power system, and the image acquisition interface can be connected to the camera to complete the migration to the real aircraft. The entire migration process is seamless and does not require any additional modifications.

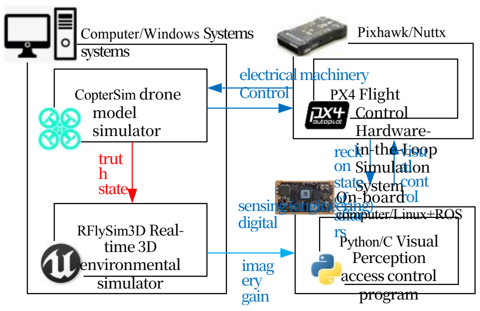

Computer/Windows
Network
Communication
Cable
Communications
CopterSim drone model simulator
electrical machinery containment
PX4 control software in the loop real system
Sen sor data
True value state of affairs
RFlySim3D Real-time 3D environmental simulator
Estimation Visual Status Control
Python/C Visual Perception vs. control program
Ima ge Acq uisit ion

(a) Pure software-in-the-loop development model for a single computer under Windows simulation of " autopilot + communication link + on-board computer".

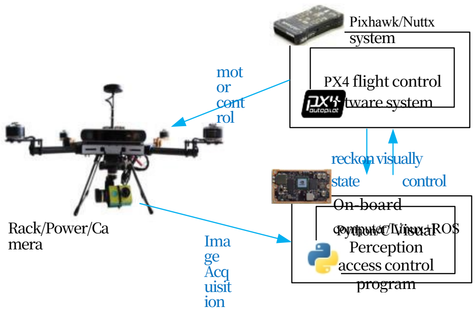

Computer/Windows Systems systems
Pixhawk/Nuttx
electrical machinery Control
PX4 Flight Control Hardware-in-the-Loop Simulation System
CopterSim drone model simulator
trut h state
esti mat ed state
cont rol
On-board computer/Linux+ROS
Sen sor data
RFlySim3D Real-time 3D environmental simulator
Ima ge Acq uisit ion
Python/C Visual Perception access control program

(b) Hardware-in-the-loop

Figure 25 Software-in-the-loop to hardware-in-the-loop migration


Computer/Windows Systems systems
Pixhawk/Nuttx
electrical machinery Control
PX4 Flight Control Hardware-in-the-Loop Simulation System
CopterSim drone model simulator
trut h state
On-board computer/Linux+ROS
sensing digital
RFlySim3D Real-time 3D environmental simulator
imag ery gain
Python/C Visual Perception access control program

(b) Hardware-in-the-loop simulation of "autopilot + communication link + on-board computer"

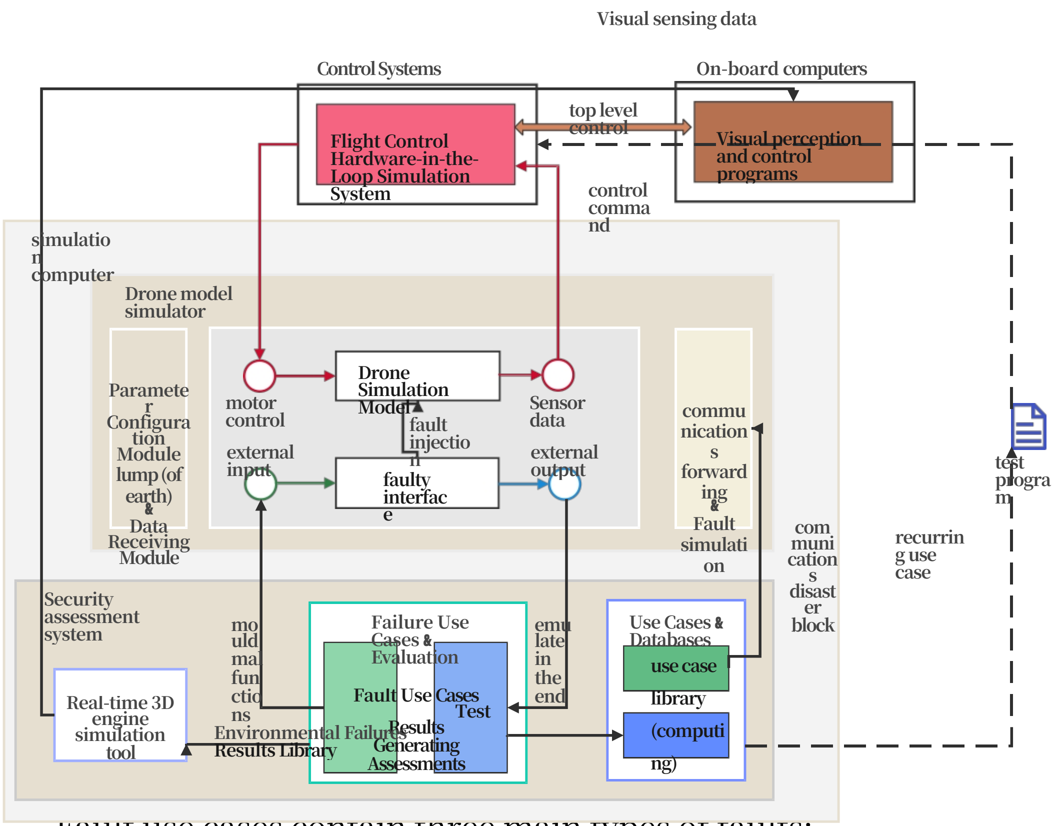

Pixhawk/Nuttx system
PX4 flight control software system
mot or cont rol
reckon visually
state
control
On-board computer/Linux+ROS
Python/C Visual Perception access control program
Rack/Power/Camera
Ima ge Acq uisit ion

(c) "Rack + autopilot + communication link + on-board computer" real-aircraft experiments

Figure 26 Hardware-in-the-loop migration to the real machine

5.2 Unmanned Systems Failure Modeling and Injection Development

The RflySim fault injection architecture is shown on the right. It consists of physical module, simulation module and evaluation module. The physical module consists of the flight control hardware, which is responsible for connecting with the simulation computer, receiving control commands from the outside and responding to the attitude, forming a semi-physical simulation closed-loop, which can be used to perform real-time fault injection in the hardware in the loop through the flight control. The simulation module consists of CopterSim, RflySim3D, and QGC, which is responsible for sending fault messages and injecting 3D faults into the whole aircraft for real-time fault simulation. The evaluation module is responsible for outputting the safety condition after fault injection. The automation test platform, which can define the test cases in the database in advance and generate data by automatic cycle testing.



Visual sensing data
Control Systems
On-board computers
Flight Control Hardware-in-the-Loop Simulation System
top level control
Visual perception and control programs
control command
simulation computer
Drone model simulator
Parameter Configuration Module lump (of earth) & Data Receiving Module
motor control
Drone Simulation Model
fault injection
Sensor data
external input
faulty interface
external output
communications forwarding & Fault simulation
communications disaster block
test program
recurring use case
Security assessment system
mould malfunctions
Failure Use Cases & Evaluation
Fault Use Cases Test
Results Generating Assessments
emulate in the end
Use Cases & Databases
use case library
(computing)
Real-time 3D engine simulation tool
Environmental Failures Results Library

Fault use cases contain three main types of faults:

Model Failure: Failure models are developed in Simulink and simulated by generating dynamic DLL models embedded in Copt erSim's model interface. Simulink model and external trigger interface are provided, and any type of fault can be added by yourself.

Communication Failure: All communication links are forwarded through the unified interface, which can simulate failures such as delay and packet loss. Environment Failure: By creating 3D obstacles in UE and importing them into RflySim3D, it is possible to simulate the environment failure when the airplane executes any

Injecting 3D obstacle faults in service.

5.3 Unmanned systems safety and health assessment system development

In the abstract, safety is the absence of system loss, personnel injury, mission compromise, or damage caused by human, machine, or medium interactions, and is generalized to refer to the state of being free of hazards and accidents. And it is actually a two-valued logic that the UAV is in a safe or unsafe state. However, in engineering, this two-valued assumption is not suitable for describing complex control systems, and the RflySim platform introduces a fuzzy risk space to describe the safety of UAVs, the

Based on the fuzzy comprehensive evaluation of the output safety level, a set of safety assessment scheme from the definition of fail-safe level→characteristic risk index analysis→health state division→automated testing and evaluation is finally formed.

5.4 FPGA-based real-time simulation system development

FPGA (Field Programmable Gate Array) chip is a kind of programmable logic device, which has the following features: Reconfigurability: FPGAs can be reprogrammed to realize different logic functions, which makes them very flexible and adaptable to different application requirements; High performance: Since FPGAs can be customized to design, they can realize very efficient High performance: Since FPGAs can be customized, very efficient logic operations can be achieved. In addition, FPGAs often have parallel computing capabilities and can process large amounts of data, making them faster than traditional processors in some applications. Low power consumption: Because FPGAs can be programmed to perform specific tasks, they can utilize energy more efficiently, resulting in less power consumption. Real-time: FPGAs can process input data in real-time, which gives them a big advantage in applications that require real-time response.

Based on the RflySim platform at the software level, the trustworthy hardware platform, the trustworthy software platform based on MATLAB, and the trustworthy simulation model form a real-time FPGA-based simulation system.

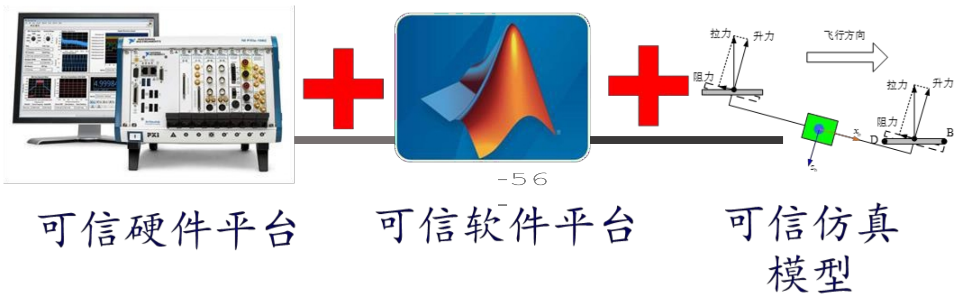


Fig. 27 **Trusted simulation system**

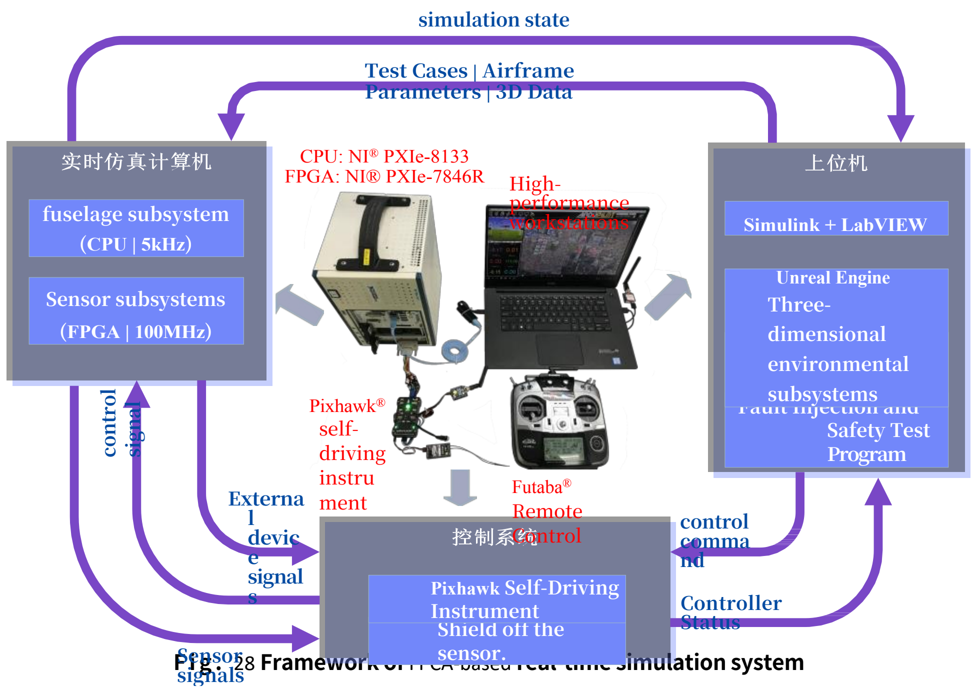

simulation state
Test Cases | Airframe Parameters | 3D Data
实时仿真计算机
fuselage subsystem (CPU | 5kHz)
Sensor subsystems (FPGA | 100MHz)
CPU: NI® PXIe-8133
FPGA: NI® PXIe-7846R
High-performance workstations
上位机
Simulink + LabVIEW
Unreal Engine Three-dimensional environmental subsystems
Fault injection and Safety Test Program
Pixhawk® self-driving instrument
Futaba® Remote Control
control signal
External device signals
控制系统
Pixhawk Self-Driving Instrument
Shield off the sensor.
control command
Controller Status
Sensor signals

Fig. 28 Framework of FPGA-based real-time simulation system

5.5 RflySim Cloud Product Overview

With the increasing demand for large-scale cluster simulation, intelligent reinforcement learning platform, one-click deployment/no deployment function, and hardware-in-the-loop virtual-reality combination simulation, the RflySim Cloud product will also be released in the future. The product will provide mechanisms for rapid deployment, continuous integration and update, and seamless expansion, and can increase the basic characteristics of the product, enhance user experience, and reduce development, operation, and maintenance costs. Through cloud-based and SaaS-based construction, the connection between simulation system and artificial intelligence system can be built quickly to form an integration advantage, and the overall cloud environment deployment framework of the product is shown in Figure 29.

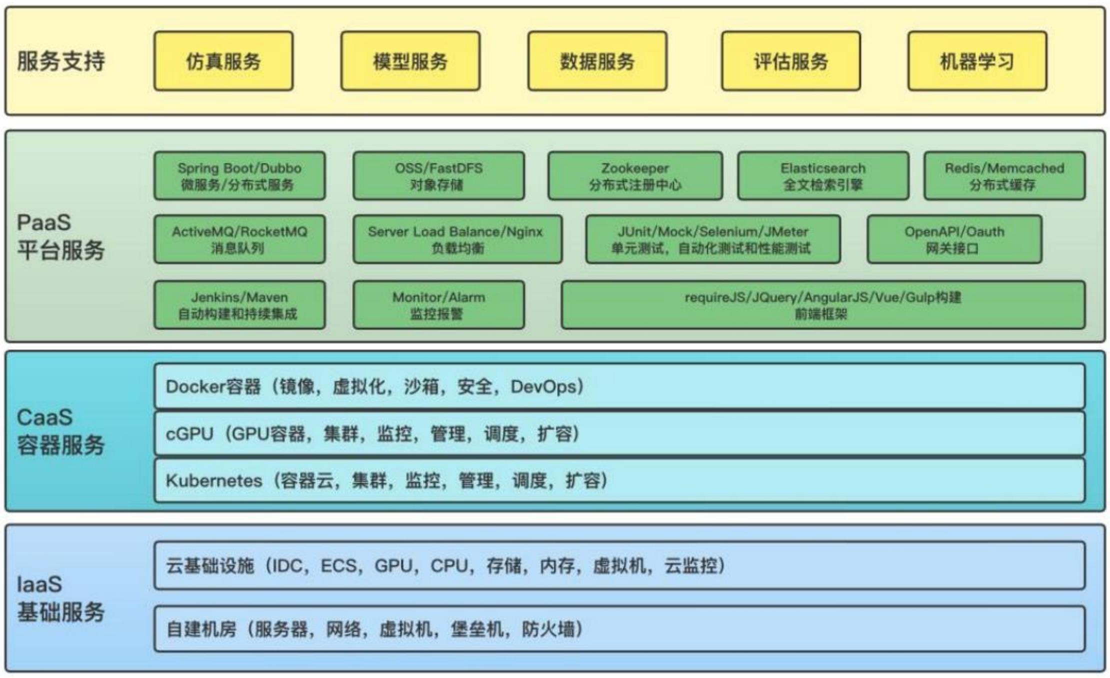
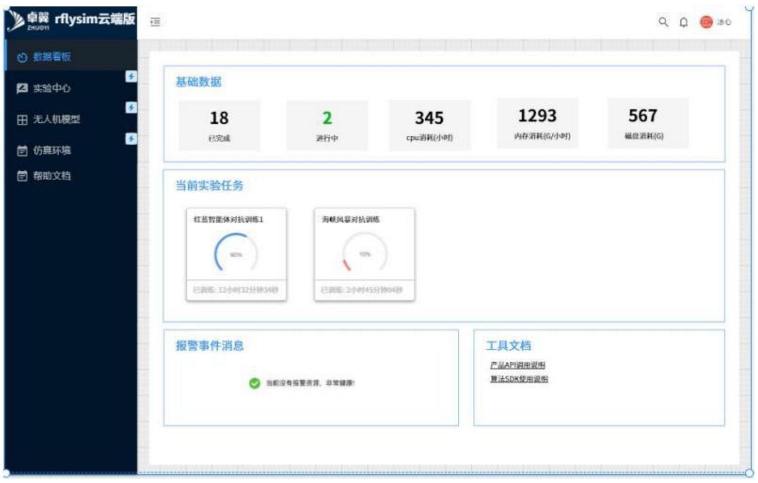

服务支持
仿真服务
模型服务
数据服务
评估服务
机器学习
PaaS
平台服务
Spring Boot/Dubbo 微服务/分布式服务
OSS/FastDFS 对象存储
Zookeeper 分布式注册中心
Elasticsearch 全文检索引擎
Redis/Memcached 分布式缓存
ActiveMQ/RocketMQ 消息队列
Server Load Balance/Nginx 负载均衡
JUnit/Mock/Selenium/JMeter 单元测试，自动化测试和性能测试
OpenAPI/Oauth 网关接口
Jenkins/Maven 自动构建和持续集成
Monitor/Alarm 监控报警
requireJS/JQuery/AngularJS/Vue/Gulp构建 前端框架
CaaS
容器服务
Docker容器（镜像，虚拟化，沙箱，安全，DevOps）
cGPU（GPU容器，集群，监控，管理，调度，扩容）
Kubernetes（容器云，集群，监控，管理，调度，扩容）
IaaS
基础服务
云基础设施（IDC，ECS，GPU，CPU，存储，内存，虚拟机，云监控）
自建机房（服务器，网络，虚拟机，堡垒机，防火墙）

Figure 29: Cloud environment deployment

The RflySim Cloud product will continue the many advantages of the RflySim platform, and at the same time, the product will provide a huge amount of model data and a complete set of equipment/environment models; it can be deployed based on docker and other rapid virtualization, eliminating the need for complex development environment deployment. The simulation performance can be expanded horizontally and resources can be used on demand; the built-in pre-trained basic algorithm models of intelligent perception, image recognition, communication links, path planning, etc. with massive data help users to build simulation applications quickly.



F i g u r e 30 RflySi i Cloud **Cloud-based Product Interface**