RflySim platform introduction documentation

RflySim platform involves three-dimensional scene construction of unmanned system development, dynamic model establishment of unmanned system, underlying control, intelligent perception, health assessment, network simulation, cluster control and so on. As shown in the table below, detailed instructions are given for each subfolder in the "[Installation Directory]\PX4PSP\RflySimAPIs" folder.

Platform Learning Route: Learn what you are interested in from Part 1 and 2 first, and get familiar with the use of the platform. Then, directly jump to the chapters of interest, in order to learn Intro, PPT, API, Readme.

Key chapter index table

Num.	Chapter name	Introduction	Routine directory name	This chapter explains	Companion Courseware	API	Routine retrieval
1	Lecture 1 - Introduction	RflySim platform introduction, version differences, installation and various functional features.	1.RflySimIntro/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
2	Lecture 2 - Experimental Platform Configuration	RflySim platform configuration process, the use method of core components and experimental process, etc.	2.RflySimUsage/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
3	Lecture 3-3D Scene Modeling and Simulation	The architecture and functions of RflySim3D software, the use of 3D modeling and scene development software for unmanned systems, etc.	3.RflySim3DUE/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
4	Lecture 4 - Vehicle motion Modeling and simulation	Unmanned system vehicle control model construction, RflySim platform model development process, etc.	4.RflySimModel/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf

5	Lecture 5 - Pose Control and filter Estimation	This chapter contains a large number of low-level development routines for unmanned systems, providing code generation and download functions, which can be used to generate PX4 firmware from the designed Simulink control algorithm and burn it in the autopilot. The basic experimental process of Sim2Real is realized.	5.RflySimFlyCtrl/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
6	Lecture 6 - External Control and Trajectory Planning	This chapter uses external control interfaces to send commands to agents to implement higher-level control functions such as trajectory planning.	6.RflySimExtCtrl/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
7	Lecture 7 - Safety Testing and Health Assessment	This chapter addresses the process of software unit and integration verification, embedded software and hardware verification, software and hardware integration verification to complete machine integration and test verification in unmanned system development. Fault injection and safety testing are realized for all the above development stages.	7.RflySimPHM/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
8	Lecture 8 - Visual Perception and Decision-making for obstacle Avoidance	This chapter describes visual sensors and related theory, such as: carrier and each sensor coordinate system, common sensors for visual control, etc. The environment configuration methods of	8.RflySimVision/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf

		Linux, ROS, MAVROS and other relat ed vision development and the visual i nterface of RflySim platform are intro duced.					
9	Lecture 9 - Communicatio n Protocols and cluster net working	The mode and status of unmanned syst em networking, the system architectur e of cluster communication in RflySim platform, and the simulation routines of unmanned system networking.	9.RflySimComm/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf
10	Lecture 10 - Cluster Contr ol and Game Countermeas ure	Focusing on the development of multi-agent swarm control for unmanned sys tems, this paper introduces the technol ogies of swarm formation, mission pla nning and game playing, and focuses o n the distributed control framework of UAV swarm system on RflySim platfo rm and the MATLAB/ Python-based cl uster control interface. Some cases, su ch as multi-UAV mission planning bas ed on ant algorithm, multi-UAV format ion, curve pipeline control, and large-s cale UAV swarm control, are provided to help readers understand the principl e and implementation of swarm contro l.	10.RflySimSwarm/	Intro.pdf	PPT.pdf	API.pdf	Index.pdf

At the same time, the internal structure of the routines folder for each lesson is shown in the table below. Different folders contain experiments of different difficulty, which are designed to help users learn the relevant content of this lesson step by step.

Sequence number	Name	Folder/file
1	Basic interface routines folder	0.ApiExps
2	Base routines folder	1.BasicExps
3	Advanced Routines folder	2.AdvExps
4	Custom Routines folder (full version only)	3.CustExps
5	This lecture introduces documents	Intro.pdf
6	This lecture interface specification file	API.pdf
7	This lecture is accompanied by courseware files	PPT.pdf
8	All routines in this section retrieve files	Readme.pdf

Key software introduction

Serial number	Subsoftware name	Introduction	Documentation
1	3DDisplay	The left side of the main interface window displays the current flight state of the multi-rotor in 3D graphics; The upper right window of the 3DDisplay main interface shows the basic flight data, including motor speed, position information, attitude information, etc. The bottom-right window of the 3DDisplay main interface displays the flight trajectory of the multi-rotor.	Readme
2	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. The multi-rotor model can be configured in the software, and the hardware-in-the-loop simulation can be realized by connecting with	Readme

		the Pixhawk autopilot through the USB serial port. To achieve the effect of indoor simulation of outdoor flight test. It mainly consists of two parts: model and communication. Model means that s imulation can be carried out directly after calculation according to the model parameters set. It sup ports running dynamic model (DLL) and forms software/hardware in the loop simulation together with other software.	
3	drivers	Drivers folder for hardware links such as flight control.	Readme
4	examples	PSP official folder for MATLAB	Readme
5	Firmware	PX4 evolved from PIXHAWK, a software and hardware project of the Computer Vision and Geo metry Laboratory of the Swiss Federal Institute of Technology Zurich (ETH). The flight control sy stem is completely open source and provides a low-cost and high-performance high-end autopilot for flight control enthusiasts and research teams around the world.	Readme
6	FlightGear 2016.1.2	FlightGear is inherited from LaRCsim and incorporated into Navion aircraft. This came with a lot of limitations (most importantly many features were hardcoded rather than configuration files) an d multiple attempts to develop or add additional flight models. The result now is that FlightGear c an support multiple different flight models, which can be selected at run time	Readme
7	Python38	Python38Env is a virtual environment based on the Python 3.8 programming language (different p rojects can have their own separate library versions and Settings without interfering with each oth er). Python38Env contains a number of specific Python libraries, Such as numpy, pymavlink, Ope nCV, pyulog, torchvision, pyyaml, utils, open3d, pytorch, tensorflow, gym and other libraries com monly used in the algorithm development of unmanned systems.	Readme
8	QGroundControl	QGroundControl is an open source ground station designed for the latest architecture of PX 4 soft ware. It uses QT editor C++ language to write its core code. It supports source code modification and function secondary development, which is suitable for UAV ground station research experime nts and also suitable for customization and modification of UAV ground station functions.	Readme
9	RflySim3D	RflySim3D uses UDP communication and is able to accept some commands from external source	Readme

		s, such as switching scenes, creating drones, and starting the built-in physical collision of UE. The details of commands will be described in the introduction of RflySim3D interface and usage. In s hort, RflySim3D can accept UDP commands from CopterSim, Python, Simulink, and return collis ion/terrain data as well as visual image data.	
10	RflySimAPIs	Routines folder	Readme
11	RflySimUE5	RflySimUE5 is a visualization software developed based on UE5.3	Readme
12	WinWSL	Windows subsystem WSL.	Readme

RflySim Platform Getting Started using documentation

1. RflySim background

The development and testing of intelligent unmanned systems are usually divided into experiment-based and simulation-based. As shown in Table 1, taking UAV development as an example, although the development and testing based on experiments is very direct, there are many pain points such as safety, space, time and cost, which are more "painful" for cluster flight testing. Table 1 Comparison between experiment-based development and simulation-based development and testing of Uavs1Simulation-based development and testing need to establish a mathematical model of the unmanned system, develop and test around the model, and finally return 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 is not true, and real is too expensive. However, the experiment-based development and testing is direct but "short-term profit" behavior, while the simulation-based development and testing seems to be "troublesome" but "long-term profit" behavior. For example, Tesla engineers say they spent 10 years modeling energy flow to achieve range improvement without replacing the battery pack. However, as far as the author knows, most small and medium-sized companies in China rely heavily on experiments for UAV development, and only large companies and aerospace institutes in developing important national models will adopt a model-based development process.

Table 1 Comparison between experiment-based development and simulation-based development and testing of Uavs1

Experiment-based development and testing	Simulation-based development and testing
Safety pain point: The rotor speed is high, and the flight process is dangerous, especially for school students	Indoors Only need computer and other equipment, low cost, limited space small... .
Space pain point: indoor space is expensive, and outdoor airspace is difficult to apply	Any failure can be simulated and injected automatically in the desired flight environment.
Time pain points: Drones are often unstable, debug test time is expensive, and most of the time is spent on hardware debugging rather than algorithms	All states, inputs, and outputs can be obtained, and the truth value can be obtained at any time.
Cost pain point: the cost of hardware is high, the debugging process often falls off, and the hardware replacement time is fast	It can be carried out at any time in the experimental development stage. The reliability of the results is difficult to guarantee and is usually only used for development and functional testing.

Figure 1 shows the whole process and whole module system architecture of a typical unmanned agent cluster cooperative control from simulation to experiment. FIG. 1 Block diagram of a typical unmanned agent swarm cooperative control simulation and experiment system1Involving the d esign and construction of unmanned agent system, the design and construction of communication system, the construction and design of positioning system, the construction and design of navigation and motion control system, the construction and design of load system, the construction and design of mission planning system, the construction and design of ground station integrated control system, and so on. Is a huge ecosystem and tool chain.

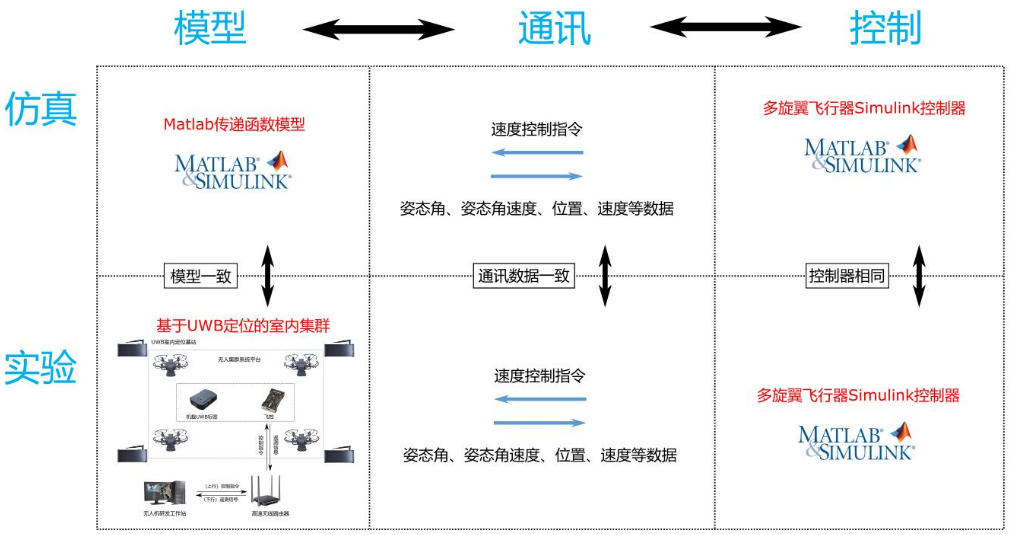


FIG. 1 Block diagram of a typical unmanned agent swarm cooperative control simulation and experiment system1

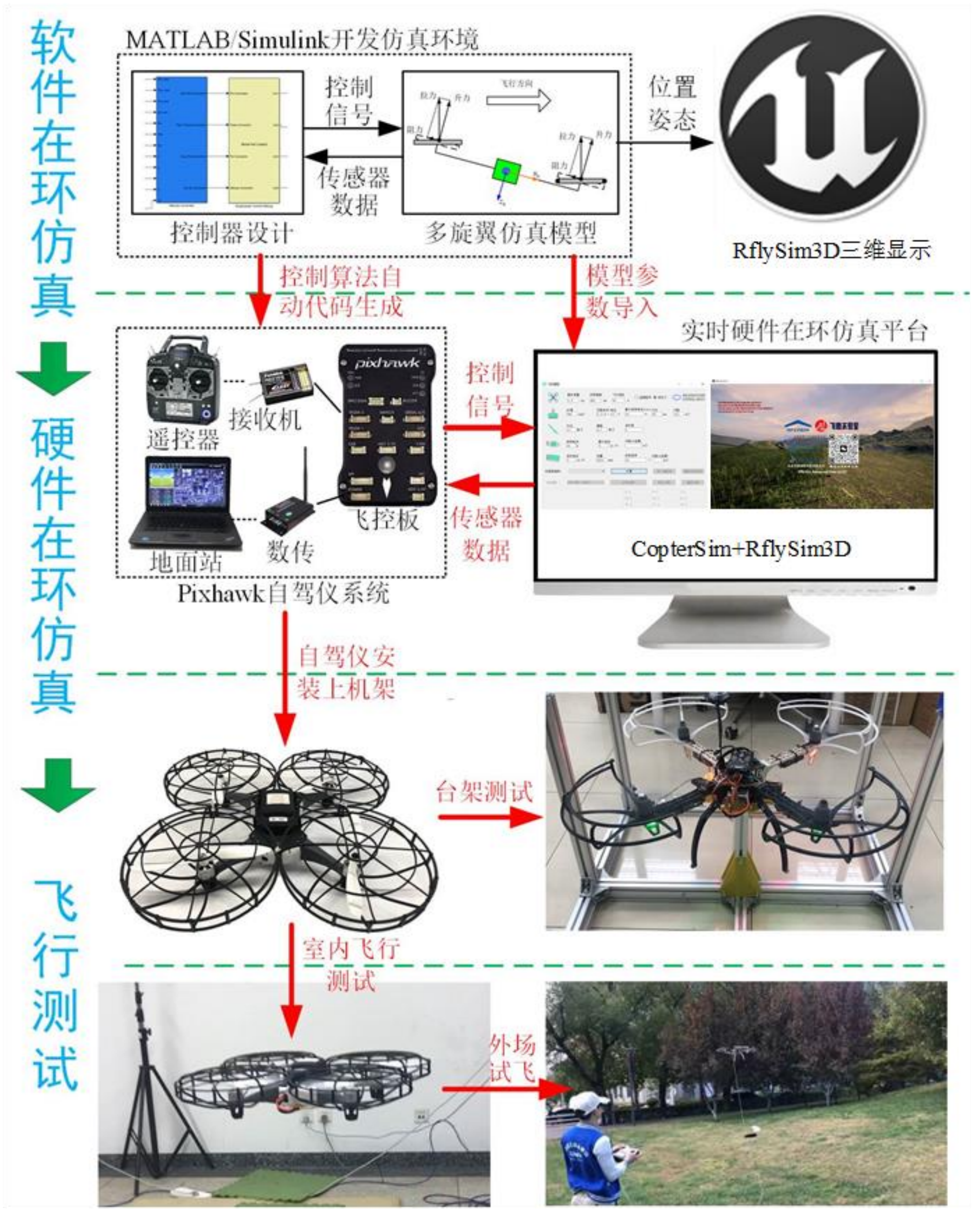
At present, most universities and research institutes in the field of multi-agent cooperative control research often have the following difficulties:

- The whole platform design tool chain is complex and huge, which is time-consuming and laborious to build from scratch.
- Due to the lack of systematic platform construction strength, personnel energy was spent on non-core research responsibilities in the early stage of the research.
- The existing decentralized software and hardware have inconsistent usage standards, software interfaces, communication protocols, and related source codes. It is difficult to learn and master and secondary development.
- Some open source platforms have insufficient service support capabilities and cannot meet the needs of local scientific research.

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

2. What is RflySim?

RflySim is an ecosystem or toolchain released by Beihang Reliable Flight Control Group. It i s directed by Professor Dai Xunhua and developed by Dr. Dai Xunhua. After that, the Fesi Laboratory under Zhuoyi Intelligence takes over and promotes the development of advanced functions. It is a set of high credibility unmanned control system development, test and evaluation platform specially developed for the frontier research fields such as unmanned platform control system development, large-scale cluster collaboration, artificial intelligence vision and so on. The platform adopts the design concept of model-based (MBD), based on Pixhawk/PX4, MATLAB/Simulink and ROS, as well as shelf intelligent hardware, etc., and can carry out (not limited to) : Simulation and real flight/motion of unmanned agent control, simulation and real flight/motion of unmanned agent clusters, and simulation and real flight/motion based on unmanned agent vision. In order to solve the above problems, we can carry out unmanned system modeling, controller design, Software-In-the-Loop simulation (SIL), Hardware-In-the-Loop simulation (HIL), and so on. 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, and the Sim2Real can be realized.



3. RflySim platform version division

RflySim platform is currently divided into three versions: **free edition, personal edition, Collection edition**, **full edition**, **Enterprise edition**, please consult service@rflysim.com).

Free version: As an experimental platform, it corresponds to the book "Design and Control Practice of Multi-rotor Aircraft". The installation package is small in size and only contains the functions developed by PX4 underlying algorithm Simulink.

Personal version: Based on the free version, the number of UAV swarm simulation is less than 15, and more routine resources are supported for optional purchase.

Collective version: Based on individual version, the number of aircraft in cluster simulation is unlimited (Note: The final number of aircraft that can be simulated depends on the performance of the computer, usually the SITL of the high-end computer is less than 15, HITL is less than 20, Si mulink&DLL supports more), aircraft dynamics model development, UE4 3D scene development, visual control development and cluster algorithm development, but the number of clusters and distributed simulation are limited.

Full version: retain all functions of RflySim, add the latest UE5 engine, global large scene simulation, distributed LAN cluster visual simulation and other functions.

Enterprise Edition: Based on the full edition, The new CopterSim and RflySim3D support hidden or custom LOGO, support large-scale cluster simulation of multi-computer distributed networking architecture, support Redis communication protocol (for large-scale distributed cluster simulation), with customized large-scale advanced routines (helicopter, tilt rotor, multi-machine cluster experiment, etc.), support Windows high High-performance computers, or Linux servers for deployment (RflySimCloud cloud platform), support for FPGA-based ultra-high real-time hardware-in-loop simulation platform (support Ardupilot and other flight control) and so on

Release difference: see http://doc.rflysim.com/1/RflySimIntro/RflysimVersions.pdf.

4. RflySim platform features

The advanced and customized versions of RflySim platform have the following features:

4.1. Unity

The whole research framework extends to all unmanned control systems, forming a standard automatic development, test and evaluation framework system.

4.2. High-fidelity uav models

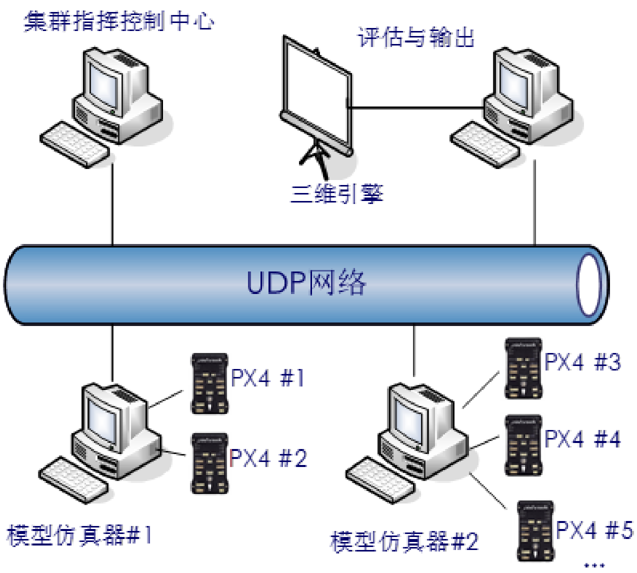
Its developers are all UAV research teams with rich research experience in the field of UAV.

4.3. Ease of use

It is very convenient and easy to use for one-click installation, one-click code generation, one-click firmware deployment, one-click software and hardware in the loop simulation and fast real flight under the Windows platform. Users do not need to know the flight control source code, Linux programming, C/C++ programming, network communication, aircraft assembly and other low-level knowledge, only need to have the basic Simulink (or Python) knowledge, they can quickly verify their own algorithms and apply them to the real plane, which is helpful to focus on the development and testing of algorithms.

4.4. Fully Distributed architecture

All the application software can be open on one or more computers, and each application can send and receive messages through UDP network. This distributed architecture is very suitable for large-scale UAV swarm simulation test with vision.



4.5. A variety of aircraft simulation

Support car, fixed wing, vertical take-off and landing aircraft (VTOL) and other models. Users can build the frame model according to the standard interface in Simulink, and then automatically generate DLL files for HIL simulation. Furthermore, the experimental platform can be extended to any unmanned system.

4.5.1. SIL/HIL simulation of UAV swarms

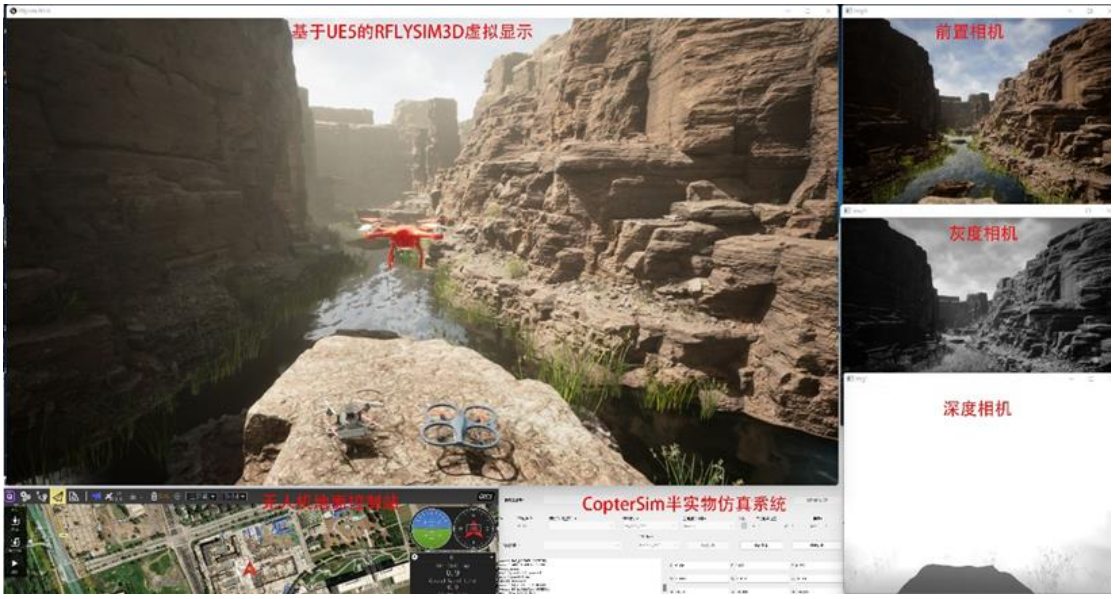
In the same local area network, developers can use CopterSim to connect multiple Pixhawks for hardware or software in-the-loop simulation. At the same time, Simulink or C++ programs can also be used to control the aircraft, and the control instructions will be sent to Pixhawk by the Mavlink protocol through the serial port (data transmission) or network (WIFI).

4.6. Highly realistic 3D viewing

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 development; The scene supports physical collision engine, global terrain and map, OSGB+Cesium tilt camera scene map import, custom GPS coordinates, arbitrary multi-window switch observation, RGB, depth, gray, IMU, lidar and other sensor data output, support shared memory or UDP picture directly sent to a specified IP address. It can be used for hardware-in-the-loop SLAM simulation of airborne computer.

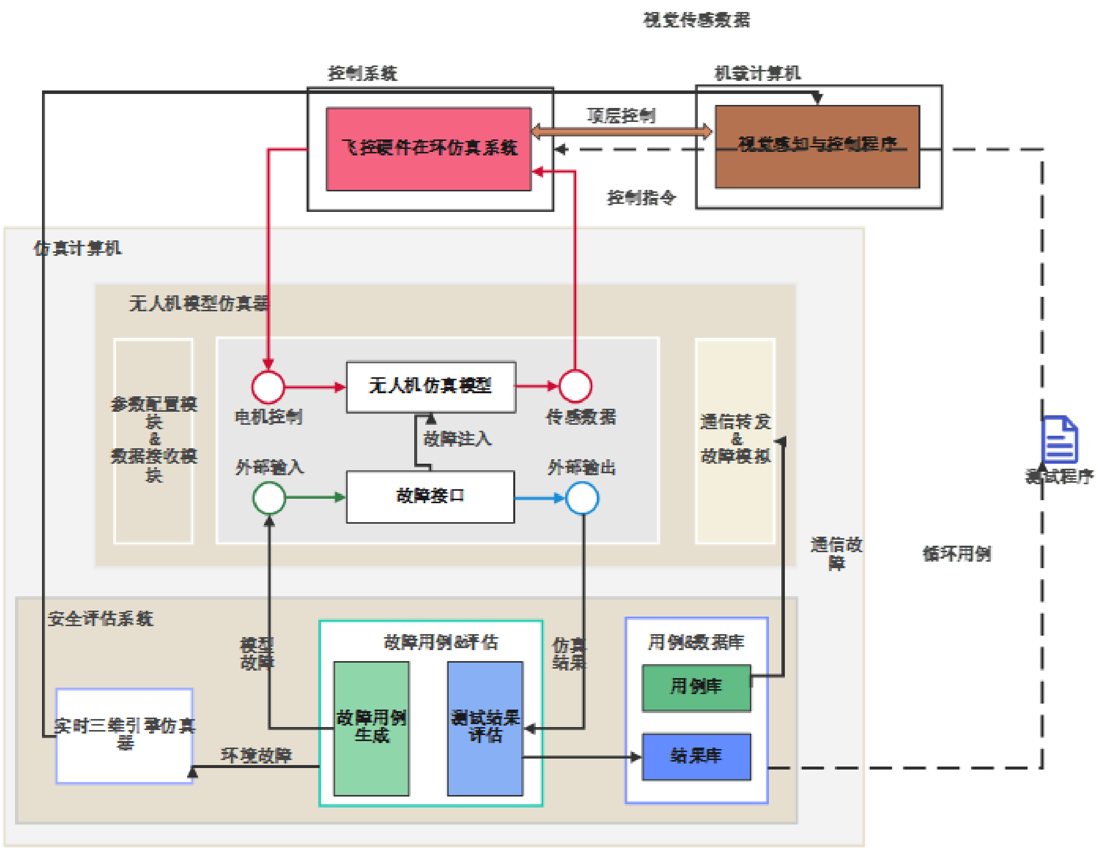
4.7. It supports vision-based control

The Ue-based 3D scene platform also supports the view switching function, which can easily obtain image data from multiple views. It also supports real-time image d ata acquisition and processing in Simulink, Python, C/C++ and other code platforms b y means of shared memory. The visual data processed can be returned to CopterSim or Simulink control through UDP, forming a hardware-in-the-loop simulation closed l oop with vision.



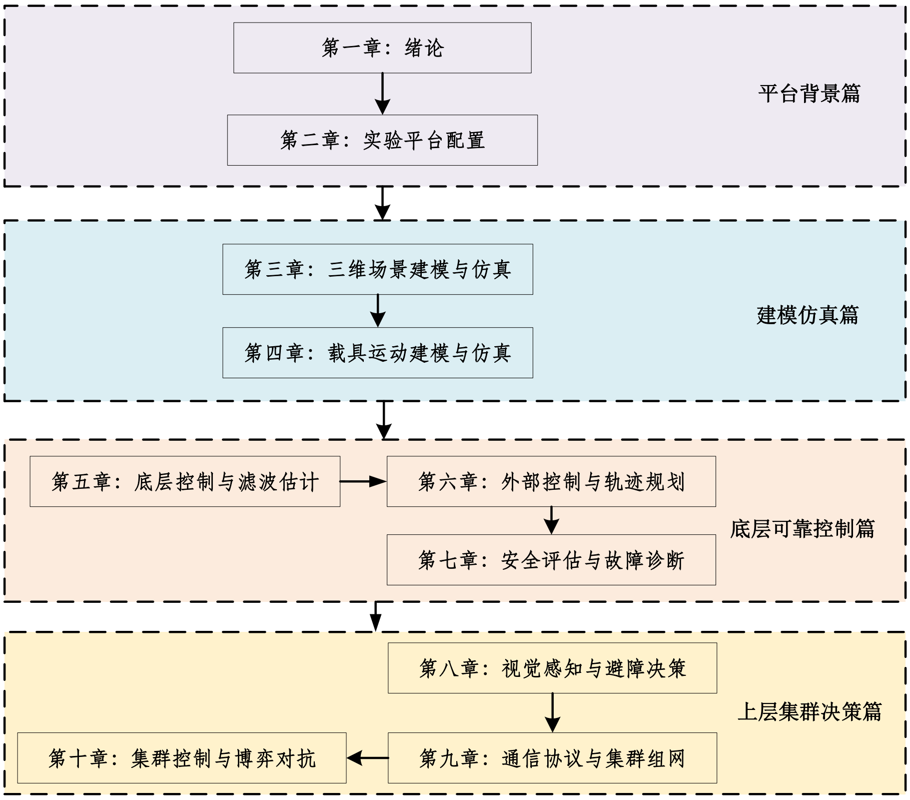
4.8. Fault injection

The types of faults that can be implemented include model faults, communicatio n faults, and environmental faults.



5. RflySim platform learning materials

This platform is mainly divided into 10 different chapters according to the stage of the unmanned system development process. Its contents include scene construction using RflySim, dynamic model establishment of unmanned system, underlying control, intelligent perception, health assessment, network simulation, cluster control and so on. The second chapter is about using the RflySim platform. Through this chapter, the installation method of the software platform and the hardware platform configuration can be completed. Through the first two chapters, readers can have an overall understanding of the research object and the use of tools, which lays a foundation for the subsequent development and design. In the third and fourth chapters, the development of 3D model and scene is related to the working environment of the unmanned system, and the development of unmanned vehicle system modeling is related to the mathematical model of the unmanned system. Based on the practice of Chapter 3 and 4, the fifth and sixth chapters include the low-level control development practice related to how to control the power unit of the unmanned system, and the remote control development practice related to how to control the speed of the unmanned system and other high-level instructions. The unmanned system involved in the latter is a composite system of "unmanned system + autopilot", in which the autopilot can be used to control the speed of the unmanned system. Remote control development practices are mostly for high-level decision-making. In the seventh chapter, the practice of fault injection and safety test development mainly involves how t o design the unmanned system model with fault injection, and how to carry out fault injection. This part of the work can be used to test the reliability and fault tolerance of the system, etc. In Chapter 8, it was explained that many intelligent systems depend on vision as an informative sensor. The practice of developing vision algorithms mainly involves how to access vision sensors and complete visual input to control. The tenth chapter includes cluster communication networking development practice and cluster control development practice. Cluster related work can also be combined with stand-alone related work to build different scenarios. The overall framework is as follows:



6. Platform Version Differences

RflySim platform is currently divided into three versions: **free version, full version, enterprise customized version**, please consult service@rflysim.com).

- **Free version**: supports up to 8 aircraft (hardware and software in the loop simulation of 8 aircraft); Visual board card in the loop simulation for 1 aircraft. Note: when the CopterID of CopterSim is 1, the LAN communication mode can be turned on, and the virtual machine or NX board can be supported to realize VIation-in-loop simulation. With helicopter, vertical aircraft, underwater vehicle and other DLL dynamic models, it can carry out task-level software and hardware in the loop simulation, but does not provide model source code. Does not support online button, can not send messages to the LAN. Only single computer simulation, does not support distributed multi-computer network, the formation of large-s cale cluster simulation. It does not support advanced simulation modes such as HITL_NET. It cannot connect Pixhawk with network port in LAN (e.g., 6x) or third-party flight control for hardwarein-the-loop simulation.

- **Full version:** RflySim3D supports receiving LAN data (default closed, selective open) and generating infrared images. CopterSim supports online mode (UDP mode, small-scale distributed simulation) and does not support Redis communication protocol. Does not support RflySimCloud large-scale cluster framework (for large-scale distributed cluster simulation) does not support LOGO replacement or masking (for enterprise customization) with UDP-based distributed visual cluster simulation routines, digital twin routines, etc.
- **Enterprise custom edition:** CopterSim and RflySim3D support hiding or customizable LOGO; Support large scale cluster simulation of multi-computer distributed network architecture; It supports Redis communication protocol (for large-scale distributed cluster simulation). With customized large advanced routines (helicopter, tilt-rotor, multi-aircraft cluster experiment, etc.); Support for Windows high performance computers, or Linux server deployment (RflySimCloud); Fpga-based ultra-high real-time hardware-in-the-loop simulation platform (supporting Ardupilot and other flight control).

The more difference between each version, please see: https://rflysim.com/doc/zh/RflySimAPIs/1.RflySimIntro/RflysimVersions.pdf

For information retrieval of each chapter and introduction of each software of RflySim platform, please see: Information Retrieval of each chapter and Introduction of each software of RflySim platform.