IMFlySim: A High-Fidelity Simulation Platform for UAV Swarms

Shi Chen, Cheng Zhang, Chengwei Yang, Juan Li, Chang Liu, Zhenbei Wang, Jie Li,
and Yu Yang(✉)

School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China
`7520210073@bit.edu.cn`

Abstract. The development of unmanned aerial vehicles (UAVs) shows the development trend of high intelligence and UAV Swarms, which makes the UAV Swarms flight trails in the real environment expensive and time-consuming. The current simulation platforms that can serve UAV swarms have problems such as insufficient number of nodes, low visual rendering fidelity, lack of intelligent interfaces, and poor real environment reproduction. This paper proposes a high-fidelity simulation platform for UAV swarms, which is a multi-node simulation platform based on Client/Server (C/S) architecture with Unreal Engine as the rendering engine and JSBSim as the physical engine. JSBSim can realize multi-model, high-frequency hardware-in-the-loop data simulation and Unreal Engine provides data interface for intelligent sensing and decision algorithms. The C/S architecture can increase the maximum number of hardware-in-the-loop simulation nodes to meet the UAV swarms flight requirements. In this paper, simulation experiments of fixed-wing are conducted and compared with the flight test, the error is between -2.851017 and 3.108375, with a mean of -0.034536 and a variance of 0.476047. The error always fluctuates around 0, and this result is not bad. Finally, this paper successfully realized formation flight simulation.

Keywords: UAV swarm · High-fidelity simulation platform/system · UE4 · JSBSim

1 Introduction

In recent years, with the development of technologies such as reinforcement learning, intelligent perception, migratory learning, and swarm decision-making, the intelligence and embedded computing capabilities of unmanned aerial vehicle(UAV) swarm have increased significantly. However, while artificial intelligence techniques have brought high intelligence and autonomy, they have also significantly enhanced the quality and quantity of high-fidelity dataset requirements, which poses several levels of challenges for research and progress in UAV swarm flight technology [1]. Firstly, the frequent data

---

This work was supported in part by the National Natural Science Foundation of China under Grant 62003043 and the Beijing Institute of Technology Research Fund Program for Young Scholars.

Z. Ren et al. (Eds.): *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, LNEE 934, pp. 209–220, 2023.
https://doi.org/10.1007/978-981-19-3998-3_21

acquisition experiments are very costly. Secondly, in order to serve the UAV swarms algorithms, the vehicle needs to employ multiple high-performance sensors for different environments, which also adds significantly to the cost and time spent on UAV swarm experiments. Finally, UAV swarm flight experiments of fixed-wing are inherently challenging in terms of number, complexity, wear and tear [2]. These issues significantly enhance the need and difficulty of researching intelligent simulation platforms for UAV swarm flight (Fig. 1).

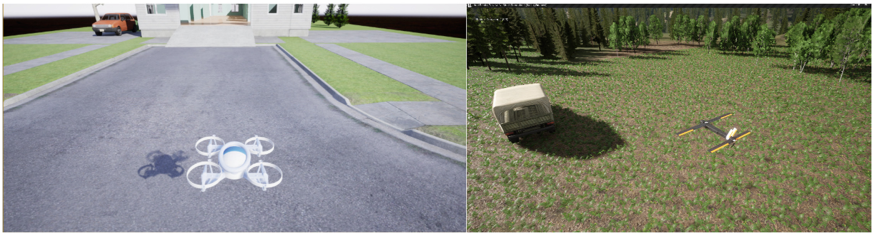

Fig. 1. Rendering of a rotor (left) and a fixed wing (right) of the simulation system.

Therefore, researchers have designed many simulation platforms to solve these problems, such as the XPLANE series [3, 4], JSBSim [5], FlightGear [6], GAZEBO [7], RFLYSim [8], Airsim [9]. FlightGear and XPLANE have multiple types of fixed-wing UAV aerodynamic models and can realize the flight simulation of fixed-wing UAV. but their rendering engines use mapping mode when importing map models, there is a large gap between the real environment. GAZEBO works most closely with the ROS [10] framework and allows the use of different physics engines, sensor models and vehicle models. AirSim and RFlySim are both simulation platforms based on the Unreal Engine and are both designed to provide intelligent interfaces and simulation platforms for fixed-wing vehicles or air vehicles. However, AirSim uses Unreal Engine's own physics engine, which requires modification and addition of dynamics and aerodynamic models when adding fixed-wing flight models, making it too complex and cumbersome. The number of nodes is limited by the transmission effect in RFlySim, and there is a certain impact on data validity, making it difficult to ensure real-time data and synchronization.

Combining the features and advantages of the above simulation platforms, we propose a high-fidelity simulation platform for UAV swarms(IMFlySim), which is a multi-node simulation platform based on Client/Server(C/S) architecture [11] with Unreal Engine as the rendering engine and JSBSim as the physical engine. JSBSim can realize multi-model, high-frequency hardware-in-the-loop data simulation and Unreal Engine [12] provides data interface for intelligent sensing and decision algorithms. The C/S architecture can increase the maximum number of hardware-in-the-loop simulation nodes to meet the UAV swarms flight requirements. Each node consists of JSBSim, Unreal Engine, PX4 firmware and NVIDIA NX. IMFlySim aims to solve the common problems of poor rendering, insufficient number of nodes and lack of intelligent algorithm interfaces in the current fixed-wing vehicle simulation, and provides highly reducible data sets and simulation validation for artificial intelligence algorithms based on deep learning, reinforcement learning and so on.

IMFlySim will provide an open source platform for UAV swarm simulation that can be developed twice, aiming to reduce the gap between the simulation environment and the real environment in the development of UAV swarm intelligence technology, and to reduce the cost of UAV swarm intelligence technology development. IMFlySim also can provide effective data sets and simulation scenarios for artificial intelligence algorithms such as reinforcement learning, deep learning, etc. IMFlySim draws on several previous simulation platform, and has developed a design and research based on them, hoping to provide a more convenient and advanced research platform for the field of UAV swarm intelligence.

The main work of this paper is as follows:

- A high-fidelity hardware-in-the-loop simulation platform based on Unreal Engine and JSBSim, providing an intelligent algorithmic interface and platform for fixed- and rotary-wing vehicles.
- A multi-node simulation platform is proposed for swarm intelligence requirements, and a multi-node initialization, communication and control framework is designed and built to provide a verification platform for swarm intelligence algorithms in a real environment.
- Designed and built a secondary developable simulation platform data interface and artificial intelligence algorithm interface to facilitate the porting and testing of intelligent sensing or swarm decision-making algorithms.
- Comparing the data parameters of the simulation platform with those of the real environment to verify the authenticity and validity of the simulation platform.

2 Architecture of IMFlySim

IMFlySim is dedicated to solving the current problems of low intelligence, distorted graphics rendering, difficulties in multi-node control, the existence of upper limits on nodes, and the lack of support for multi-class UAV models when it comes to UAV swarms simulation, with emphasis on high fidelity, data alignment and scalability of the simulation platform. IMFlySim can carry out both single-node simulation and multi-node simulation. Multi-node simulation is based on single node, adding communication and control, etc., which can achieve the effect of UAV swarms.

2.1 Single-Node Structure

Single-node simulation includes hardware-in-the-loop simulation and software-in-the-loop simulation, the main differences being the platform on which the algorithm is run and the way in which the data is communicated. Software-in-the-loop simulation is convenient for experimental verification, low cost of simulation and easy to use, but poor consistency with realistic environment. The biggest gap lies in the platform on which the algorithm runs and the realism of the flight parameters. In the hardware-in-the-loop simulation, NVIDIA NX and Pixhawk2 are used as the embedded processing board and self-pilot, and AI top-level controller and PX4 flight controller are equipped on them to simulate and emulate the vehicle respectively.

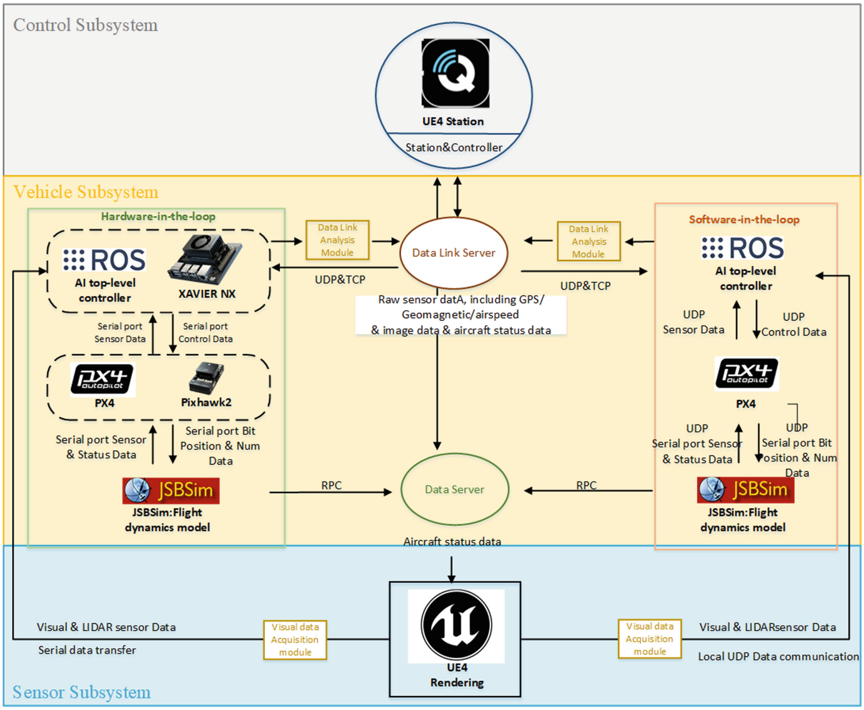

Fig. 2. Single-node simulation structure.

The structure of the single-node simulation is shown in Fig. 2. The single-node simulation platform can be split into three parts: the vehicle subsystem, the sensor subsystem and the control subsystem. The next three sections are described in detail.

Vehicle Subsystem. The vehicle subsystem consists of two main parts: the UAV model and the physics engine.

1) UAV models.
The flight vehicle model needs to contain structural parameters, mass characteristics, inertial characteristics, aerodynamics, flight control systems and other parameters. As IMFlySim uses JSBSim as the physics engine for the vehicle, the vehicle model and aerodynamic parameters can be easily modified and optimized. We allow the rigid body dynamics to be operationalized by setting the mass, inertia, ground feedback characteristics, linear and angular drag coefficients, and aerodynamically mapped forces and moments of the vehicle model.
The aerodynamic model of a quadrotor is relatively simple, which can be expressed by the following formula.

$$F_i = C_T \rho \omega_{max}^2 D^4 u_i \tag{1}$$

$$\tau_i = \frac{1}{2\pi} C_{pow} \rho \omega_{max}^2 D^5 u_i \tag{2}$$

where C_T and C_T denote the thrust and power coefficients, ρ is the air density, ω_{max} is the maximum angular rate, D is the propeller diameter and u_i is the input control.

The flight parameters of a fixed-wing vehicle can be split into three force and moment equations for aerodynamic forces, thrust, gravity and their resulting momentum moments. The vector expressions of their flight dynamics:

$$\boldsymbol{F} = \frac{d\boldsymbol{v}_c}{dt} + m(\boldsymbol{\omega} \times \boldsymbol{v_c}) \tag{3}$$

$$\boldsymbol{M} = \frac{d\boldsymbol{H}}{dt} + \boldsymbol{\omega} \times \boldsymbol{H} \tag{4}$$

where **F** is the combined force of the fixed-wing vehicle, **M** is the combined moment, $\boldsymbol{v}_c$ is the vehicle velocity, m is the mass, $\boldsymbol{\omega}$ is the angular velocity of rotation and $\boldsymbol{H}$ is the momentum moment. After splitting $\boldsymbol{v}_c$ and $\boldsymbol{\omega}$ in the x, y, z directions into $\boldsymbol{v_c} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k}$ and $\boldsymbol{\omega} = p\boldsymbol{i} + y\boldsymbol{j} + r\boldsymbol{k}$, the scalar equations:

$$\boldsymbol{F_x} = m(\dot{u} + qw - rv), \boldsymbol{F_y} = m(\dot{v} + ru - pw), \boldsymbol{F_z} = m(\dot{w} + pv - qu) \tag{5}$$

$$\boldsymbol{L} = \dot{H}_x + qH_z - rH_y, \boldsymbol{M} = \dot{H}_y + rH_x - pH_z, \boldsymbol{N} = \dot{H}_z + pH_y - qH_x \tag{6}$$

where $\boldsymbol{F_x}$, $\boldsymbol{F_y}$, $\boldsymbol{F_z}$, **L**, **M**, **N** are the components of aerodynamic, gravitational, and inferred contributions respectively. Aerodynamic modelling of a fixed-wing flight model can be achieved by using equations (5-6).

2) Environment and physics engine

The environmental and physical engines are also important indicators of the realism of the simulation platform when performing vehicle simulation. The environmental engine includes gravity, air pressure, geomagnetism, air density, etc. The physics engine uses environmental data, sensor data, aerodynamics, etc. to calculate the expected aerodynamic states of the vehicle, which include position, attitude, linear velocity, linear acceleration, angular velocity, angular acceleration, etc. Small deviations in the input data of the physics engine may have a significant impact on the effectiveness and realism of the simulation results (Fig. 3).

Sensor Subsystem. The sensor subsystem obtains sensor data from two main sources. One part is the common sensor parameters of the vehicle from the physics engine, including GPS, IMU, accelerometer, barometer, gyroscope and other common sensor data. The other part is the sensor data obtained from the rendering engine, such as vision, radar, infrared and other sensors that serve the intelligence algorithm.

1) General Sensor Parameters

For the GPS data, the real environment GPS data is set up and simulates the latency (200 ms+) and packet loss (e.g. in urban environments) that exists in real environments. Also, horizontal and vertical errors are added to the GPS data. A low-pass filter is used to add error attenuation to the horizontal and vertical GPS errors so that the errors are corrected or augmented with the operating environment and operating time, improving the realism of the positioning data.

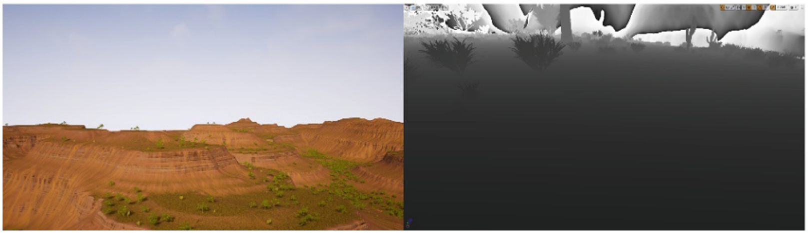

Fig. 3. Visible light simulation image (left), simulated infrared imaging effect image (right).

For geomagnetic data [13], JSBSim uses the World Geomagnetic Model WMM2000, which may be biased compared to the current geomagnetism and therefore needs to be optimized; With the release of the International Geomagnetic Reference Field (IGRF13) and the WMM2020 geomagnetic model proposed by NOAA, this simulation system optimizes the geomagnetic model used in JSBSim using the current spherical harmonic function used in IGRF13 [13].

2) Visual Parameters

In order to provide raw data for the intelligent algorithm, we add infrared material to the original rendering effect of Unreal Engine, making it possible for the simulation system to modify the target material under the rendering effect to achieve the effect of simulating infrared image, and to perform the acquisition and rendering of infrared data.

Control Subsystem. The control subsystem is divided into ground station control and airborne control according to the control mode of the UAV. Ground control can preset the flight path, flight parameters and rendering engine parameters of each UAV to control and collect data from the UAV, realizing flight control and aerodynamic model simulation of the UAV. With the development of intelligent algorithms and the improvement of hardware platform performance, the simulation system also proposes multi-language support, GPU computing, data synchronization and UAV swarms communication. Therefore, the system is designed on the XAVIER NX embedded development board with an intelligent detection, tracking, positioning, obstacle avoidance and decision-making algorithm framework based on the ROS framework. Data interaction is completed through MAVLink messages, enabling flight behavior such as target tracking and guidance to be completed with target detection and tracking implemented. The strongly distributed architecture of ROS2 is able to achieve data interoperability between UAV, distributed information interaction and UAV swarms behavior decision-making when the UAV swarms are in good communication status, and distributed UAV swarms simulation.

2.2 Multi-node Simulation

The highlight and focus of IMFlySim is on multi-formation UAV swarm countermeasures and decision-making simulation. XPlane provides simulation for fixed-wing UAV,

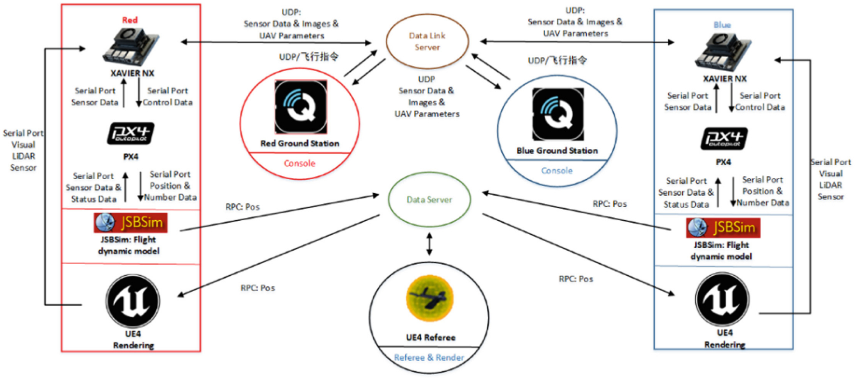

Fig. 4. Multi-node simulation structure. The red and blue camps are shown as an example, showing the information transfer logic and communication methods of the components of multi-node simulation during normal operation.

is limited by its P2P structure and has a connection limit of 20 nodes. And ground control platforms, QGC and Mission Planner, do not have the capability of real-time multi-UAV multi-formation manoeuvres and UAV swarm simulations. But, IMFlySim is an infinitely scalable distributed multi-node hardware-in-the-loop simulation platform based on the C/S framework. The structure of multi-node simulation is shown in Fig. 4.

The single-node simulator in Sect 3.1 is used for the individual nodes in this platform. The node modules are identical and their built-in parameters are completed by the initialization phase of the simulation platform, including data such as the node's array allocation, UAV type, flight scenario, initial position and initial attitude. The simulation platform designs the judgement and the controller of each formation, and secondarily develops previous ground station. The multi-node simulation structure mainly consists of four parts: referee, console, server and simulation nodes. In this paper, we will introduce the referee, the console and the data relay server respectively.

Referee and Console. The referee and the console have different functions in the multi-simulation node. We want the simulation to have as much effect and functionality as possible for multi-formation confrontations, so the global display function is split into two parts, the referee and the console.

The referee is a simple interactive interface that is used to monitor the connection status of each hardware in the data server during the initialization phase. When the available nodes are confirmed, parameters such as scenario map, climate, time, UAV models, number of camps, hardware or software emulation can be set. During the normal working phase, the referee is able to obtain the simulation data of all camps and all nodes and render and display them, enabling the monitoring function for all consoles. In addition, the referee has the right to start, pause and restart the global simulation platform (Fig. 5 and Fig. 6).

The console is a secondary development of the ground station software, and its main function is to provide the ground station with more convenient multi-UAV control capability and multi-UAV parameter configuration capability. Considering control and

Fig. 5. Referee parameter setting and display.

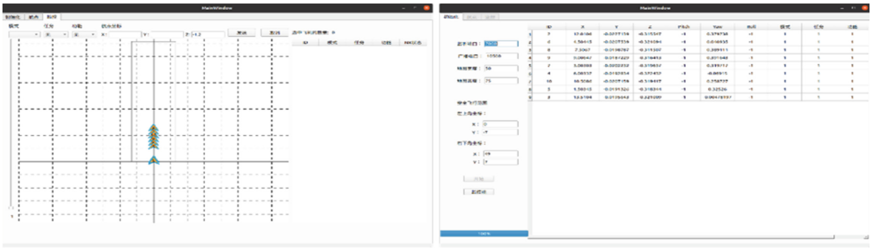

Fig. 6. Console ground station.

adjust individual node of real simulation, as well as the overall control of the UAV swarms simulation, this console flight control mode is mainly divided into two categories, one is the console to each node direct control, manually add waypoints or read the waypoint configuration file to generate the flight path and waypoints of each node, and then transmitted to each UAV; the other category is to provide the simulation function of the autonomous formation of the UAV swarm, placing the flight path algorithm in the onboard processor and performing UAV swarm intelligence algorithm simulation. In both cases, UAV swarms control of multiple UAVs can be achieved. In addition, as the simulation platform focuses on hardware-in-the-loop simulation, it is important that the flight controller parameters are configured and the intelligent algorithms are unified. To solve this issue, we develop the ground station secondarily, so that it has the ability to synchronize the configuration of multiple UAV parameters, reducing errors and workload caused by inconsistent start-up simulation platform and parameters.

Data Transfer Servers. In order to realize the functions of the referee and the console and to provide a more realistic simulation of the communication effects, two types of data transit servers are designed for data transmission and communication simulation, namely the data server and the data link server.

The main function of the data server is to communicate the data between the physics engine and the rendering engine. The dynamics of the nodes are transferred via RPC to the rendering engine and to the referee. These two parts will render the poses and states of all nodes so that they can be displayed in the same scene and complete the acquisition of visual information of the UAV swarms flight, thus realizing the simulation of vision-based algorithms such as final guidance and UAV swarm formation.

The data link server, on the other hand, takes on more of a communication simulation function. Through a large number of flight tests, it can be found that packet loss, data interruption and interference in the real environment is one of the difficulties and key points of UAV swarm flight. Communication loss and interference cannot be ignored when performing a trunked flight. Therefore, this platform is designed as a data link server, which is mainly used to carry out the simulation of self-assembled network data link between UAV. Considering that the UAV data designed for this simulation platform is sent out via the embedded board, it is mainly divided into two types of data: broadcast-based inter-UAV network communication data and raw MAVLINK data transmitted to the ground station at the control end, both of which are subject to interference such as communication distance and environment. Therefore, the data link server distributes and forwards the messages that each UAV needs to receive based on the UAV location and environmental factors after receiving the inter-UAV messages, enabling the simulation of the communication function.

3 Experiments

3.1 Flight Experiments

In order to verify whether the flight characteristics of the simulation system and the real environment are consistent, this paper conducted a set of waypoint flight tests using a self-researched fixed-wing flight platform, collected rudder volume, speed and other control quantities, input them into the simulation platform for verification, and compared and evaluated the data from the simulation and tests to verify the realism of the simulation. An autonomous formation flight UAV swarm mission test was also conducted.

Hardware Platform: For realistic flights, the Pixhawk v2 is used as the flight controller on the self-developed fixed-wing vehicle used, along with external sensors such as barometer, GPS, airspeed meter and built-in sensors such as geomagnetic and IMU. In addition to this, the UAV is equipped with a Xavier NX as an embedded processor to capture flight images. The sensor data and detected physical parameters are stored inside the vehicle and the control variables from the real environment are then fed directly into IMFlySim for simulation to obtain the attitude and position of the vehicle, which is then verified for simulation realism.

Trajectory Evaluation: This experiment was conducted by comparing the flight test with the infield simulation to observe whether the actual flight trajectory was highly compatible with the simulated trajectory. The flight test experiments were conducted in Shou County of Huainan City, and the experimental design is flying seven waypoint several circles and then hovering. The waypoints as shown in Fig. 7(a), using a self-researched fixed wing to fly and collect flight data; The exact same model and the corresponding dynamics model were used in simulation to collect flight data.

Fig. 7(b) and (c) show the three-dimensional trajectory diagram of the aircraft in flight test and simulation by WGS-84 coordinate system. We analyzed and compared the pitch angle error to expected value for the same control algorithm. The results are

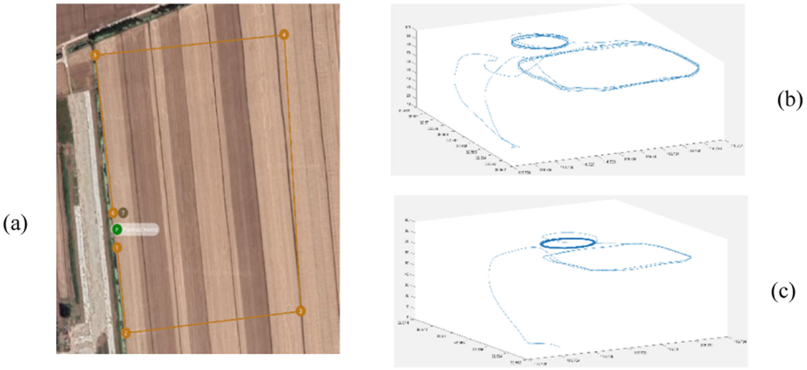

Fig. 7. Flight test and simulation using the same waypoints (a), path in flight test (b), path in simulation (c).

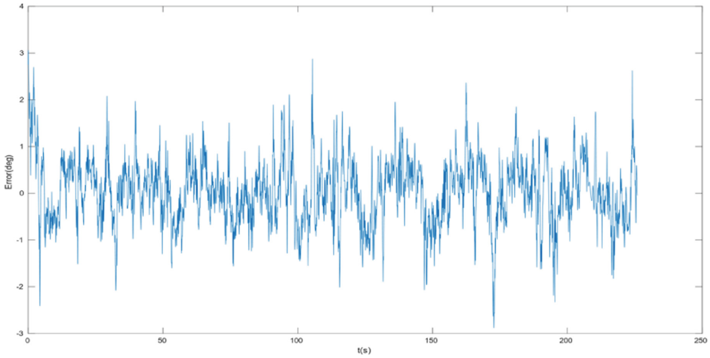

Fig. 8. The pitch error of flight test and simulation.

shown in Fig. 8. The pitch error is between −2.851017 and 3.108375, with a mean of −0.034536 and a variance of 0.476047. The error always fluctuates around 0, and this result is not bad.

3.2 Formation Experiments

Autonomous formation flight is the basic form of UAV swarm execution for combat missions and the simplest way to verify the simulation platform's support for UAV swarms simulation. During formation flight each node needs to adjust its own flight status based on the status information provided by other nodes. The formation process requires the nodes to transmit their own data and receive data from other nodes in real time. This requires the simulation platform to leave a UAV swarms simulation interface. To test whether the data interaction function of each node in the simulation platform is normal under the existing software architecture, fourteen simulation nodes are used for the simulation test of autonomous formation flight. In this test it is assumed that the 14 nodes are under ideal communication conditions, i.e. the data link simulation server is only responsible for data forwarding.

Fig. 9. The result of formation in simulation.

Before the simulation started, the 14 nodes were timed through a time server to ensure time consistency and facilitate subsequent data analysis. The simulation uses 14 nodes to fly in diamond and diagonal formation, the effect of formation flight simulation is shown in Fig. 9. The formation status of the formation is shown in the ground station at the control end. The flight status of the other vehicles can be seen in the rendering engine of referee.

The 14-node formation flight simulation shows that the software and hardware architecture of the simulation platform supports large-scale UAV swarm behavior simulation. Simulation validation of commonly used intelligent UAV swarms algorithms can be performed using IMFlySim.

4 Conclusion and Future Work

IMFlySim is able to provide a high fidelity physical and visual simulation for both fixed and rotary wings, allowing very cheap and easy generation of large amounts of training data for intelligent UAV swarm studies. The core components of the simulation platform, JSBSim and UE4 rendering, enable multi-UAV and multi-scenario simulation and IMFlySim is based on a C/S architecture, effectively increasing the hardware in the upper limit of the number of loops to meet UAV swarm flight requirements. Simulating the real world in a simulation system is a challenging task and there are many areas that could be improved. In addition, in the future we will be increasing the type of sensors used to simulate realistic flight processes and obtain realistic data.

References

1. Zhang, J., Wu, X.: Analysis of UAV clusters and their power requirements. Aerodynamics **04**, 32–34 (2021)
2. Wang, X., Liu, Z., Cong, Y., Li, J., Chen, H.: Review and future development of small fixed-wing UAV clusters. Journal of Aeronautics **41**(04), 20-45 (2020)
3. He, G., Yu, L., Jia, S., Wang, X.: Simulation verification of Flight Control of a tilt tri-rotor UAV Using X-plane. Technical Committee on Control Theory, Chinese Association of Automation, Chinese Association of Automation, Systems Engineering Society of China, p. 6 (2020)

4. Zhou, Y., Hou, C.X., Wang, B., Shan, S.: Aileron failure diagnosis and control surface reconstruction simulation of small remotely piloted aircraft in X-Plane environment. Science and Technology Bulletin **30**(11), 194-197 (2014)
5. Jon, S.: Berndt and the JSBSim Development Team. JSBSim Reference Manual (2011)
6. Wood, A., et al.: GymFG: A Framework with a Gym Interface for FlightGear. arXiv preprint arXiv:2004.12481 (2020)
7. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IROS (2004)
8. Dai, X., et al.: RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. Aerospace Science and Technology, 106727 (2021)
9. Shah, S., et al.: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles (2018)
10. Jackson, J., Ellingson, G., McLain, T.: Rosflight: A lightweight, inexpensive mav research and development tool. In: ICUAS, pp. 758–762 (2016). https://doi.org/10.1109/ICUAS.2016.7502584
11. Peng, Z.: Research and design of distributed management system architecture. Info. Technol. Informati. **07**, 72–74 (2019)
12. Karis, B., Games, E.: Real shading in unreal engine 4. In: Proc. Physically Based Shading Theory Practice (2013)
13. Fei, P., Ling, X., Yang, C.: Simulation of geomagnetic model construction and selection of matching feature quantities. Automation and Instrumentation **35**(8), 87–92 (2020). https://doi.org/10.19557/j.cnki.1001-9944.2020.08.018