



A UAV-Swarm Control Platform Architecture Based on Cloud

Li Zeng^(✉), Zesheng Zhu, Xuzhou Shi, and Yulei Liu

Nanjing University of Aeronautics and Astronautics, No. 29 Jiangjun Road, Nanjing, China
leey0410@nuaa.edu.cn

Abstract. In the recent 20 years, the research of UAVs has attracted the attention of researchers. At the same time, the drone swarm control system has also developed rapidly. However, the way of controlling a single drone through the drone controller cannot meet the control needs of the drone swarm, which significantly limits the interaction of the drone swarm, the sharing of its resources, and the ability of the drone to perform complex tasks. This paper proposes a centralized cloud platform-based drone cluster control platform architecture for collaborative control of drone swarm, which is combined with workflow technology to allow multiple users to control multiple drones to perform tasks and monitor in parallel via the Internet. The state of the man-machine is not restricted by time and distance and can be connected to the cloud platform anytime and anywhere. Besides, the architecture also provides cloud storage and cloud computing support for drones and improves the support of complex tasks for drones. We conducted experiments through the simulator to verify the effectiveness and feasibility of the architecture.

Keywords: UAV · IoD · Cloud platform · Control architecture · Workflow

1 Introduction

Unmanned aerial vehicles, as an up-and-coming industry and technology, have typical and extensive applications in the military, commercial, and civil fields, which have changed our production and lifestyle in a subtle way, such as reconnaissance detection, communication. There are huge potentials in many different fields such as communication relay, precision agriculture, atmospheric sampling, and aerial photography [1]. However, with the widespread application of drones in different fields, the changing task environment, and complex task structure have brought more challenges and higher requirements to drones. A single UAV is severely limited by its own size and weight. Its computing power and battery life are also severely affected. It cannot perform complex and computationally demanding tasks alone, such as image recognition. At the same time, most of the current drone control methods are to establish an end-to-end connection with a control device (such as a remote control) or a dedicated APP through WLAN to achieve the control. Its limitations are also apparent. The distance between the position of the UAV and the control device will be limited to a few hundred meters, and it does

not support multiple drones to be connected to one control terminal at the same time. Therefore, breaking through the above challenges and improving the ability of drones to solve complex tasks have become an urgent problem to be solved.

One way to improve the ability of the drone to perform complex tasks is to break through the size limitations of the drone, increase its volume, and the power consumption of the UAV platform. Still, this method is not suitable for some needs that are MAV tasks. Another feasible method is to form multiple drones into a drone swarm. The drone swarm fly in coordination and help each other to perform tasks, and make full use of the limited resources of the drone—huge advantages, and thus the enormous potential of drones in different fields.

This paper proposes a control platform architecture for cloud-based UAV swarms. The architecture is inspired by the conceptual models of IoT (Internet of Things) [2] and IoD (Internet of Drones), which deploys drones to the Internet. By using cloud computing, it provides non-airborne computing resources and data storage resources for the drone swarm; through Web technology, the drone can be connected to the Internet anytime, anywhere, so that the communication with the drone can be free from any restrictions and constraints. At the 2010 Humanoid Robots conference, Kuffner [3] proposed the concept of cloud drones. Robots do not need to store all data locally and perform intensive operations with complex calculations, but they cannot complete themselves. The tasks and parts of the data are handed over to the cloud server to solve the problem of insufficient computing and storage capabilities of the UAV itself. Based on this concept, in literature [4], Arumugam et al. Based on the Apache Hadoop platform to improve the computing power of the drone and the execution efficiency of the corresponding algorithm of the drone. However, the author has not proposed a mechanism to solve the reliability of the platform. Waibel et al. [5] proposed the establishment of a World Wide Web dedicated to robots and implemented a corresponding prototype system. Based on the idea of IoT, Gharibi et al. [6] proposed a conceptual model of IoD and planned a framework structure to deploy drones from a single aircraft to the network. In [7], the author proposed a multi-drone control platform, but the multi-drone control method is too cumbersome and does not consider process control.

In our paper, we abstract the UAV as a service, namely IaaS (Infrastructure-as-a-Service) [8]. The user does not need to understand how the drone communicates with the cloud platform but only needs to control and use the required drone resources through the interface provided by the cloud platform. At the same time, the cloud platform obtains the status of the drone and the data generated by its sensors through remote call (RPC) [9], and stores it on the cloud to achieve efficient storage or perform the necessary intensive calculation and processing through the data. Pass the result back to the target. Finally, through the use of workflow engines to achieve the customization of complex control commands for drones, the control methods and control processes of drones are optimized.

In the following content, we introduce the incoming research work of the UAV control system architecture and the proposed architecture in the second part. In the third part, we simulated and verified our proposed architecture. Finally, we summarize our work and future research directions in Sect. 4.

2 Platform Architecture

The UAV-swarm control platform architecture connects the UAV and the cloud through the Internet to achieve the expansion of the UAV communication distance and achieve the beyond-horizon communication and control of the UAV. The cloud platform serves as a relay between the drone and the control station, coordinating the flight of the drone and the task assignment of the ground station. On the one hand, the cloud platform collects and obtains the data generated by the drone for storage and visual display. On the other hand, according to the needs and commands of the ground station, the generated data is analyzed, and the generated control commands are packaged and transmitted to the drone. As shown in Fig. 1, this figure is the overall architecture of our entire UAV-swarm control platform. The control platform uses a modular, low-coupling design. The overall architecture can be divided into four parts, namely the UAV Client layer, UAV Service layer, cloud platform layer, and control station layer. Each module will be described in detail below.

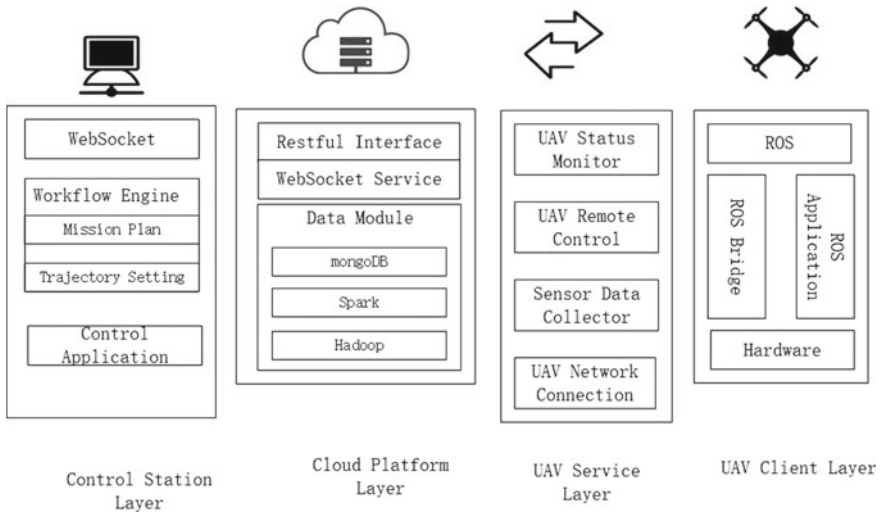


Fig. 1 Control platform architecture

2.1 UAV Client Layer

UAV can be regarded as a resource provided to users, providing similar services to external applications or users, and used to fulfill different requirements of external applications or users. To simplify the operation mode of the drone, the user and external applications deconstruct the operation of the drone and the hardware of the different drones. The introduction of the ROS robot operating system has played a crucial role. The ROS-based publish/subscribe mechanism, and the automatic boxing function provided by MAVROS can realize the conversion between ROS Message format messages and MAVLink format

messages. Secondly, because running the ROS system on a cloud platform is prone to crashing the Master node, the ROSBridge software package is used to realize the interactive communication between the non-ROS system and the ROS system. The cloud platform uses the WebSocket method and the corresponding interface provided by ROS-Bridge to establish the communication between the cloud platform and the drone, to achieve a cloud platform to drone data acquisition, drone to cloud platform data upload, and cloud platform control commands. The implementation of the uplink and downlink of the release. We have completed the abstraction of the drone client and the control decoupling of heterogeneous drones through the above methods.

2.2 UAV Service Layer

The purpose of the UAV Service layer design is to make different abstract types of drones into a universal cloud resource for drones, and provide users and external applications with an operation mode that is independent of the hardware structure and specific characteristics of the drone. As a relay, UAV Service acts as a bridge between the cloud and physical drones. First, the UAV Service layer forwards the status information of the drone and the sensor's perception data to the cloud for calculation and processing, and secondly sends the data processing results to the drone. This layer contains the following four components:

- UAV Status Monitor: It is used to monitor and manage the information of drones to facilitate the rapid maintenance of drones' resources.
- UAV Network Connection: The underlying communication module is used to handle the identity verification of UAV access to the cloud platform and the connection maintenance of WebSocket communication.
- Sensor Data Collector: Sensor data collection module, used to collect the data sensed by UAV sensors and upload it to the cloud platform.
- UAV Remote Control: An abstract package of unmanned aerial vehicle general actions, such as simple control commands such as takeoff, landing, and general flight, used to achieve a combination of complex mission processes.

2.3 Cloud Platform Layer

The cloud platform layer provides unparalleled computing and data storage capabilities for the drone fleet. According to different types of data structures and data application methods, different types of storage methods are provided. Traditional databases such as MongoDB are used to store UAV flight data, such as waypoints, UAV attitude, UAV equipment information, and UAV access authentication information. For the UAV command response data that requires high real-time performance, Storm's stream is used for real-time processing. For non-real-time sensitive big data, HDFS [10] is used for data writing and reading.

At the same time, the computing power provided by the cloud platform to the drone group can support intelligent algorithms. Through the parallel computing and processing capabilities of Map/Reduce provided by the Hadoop [11] cluster, the applicability of algorithms such as picture recognition and path planning in drone applications is improved, and the collaborative and intelligent operation of drone groups is supported.

2.4 Control Station Layer

The control station layer serves as the entire UAV-swarm control platform architecture, allowing users to obtain powerful control capabilities for the UAV group and visualization capabilities for the UAV status. Control station provides a workflow engine to simplify the user's collaborative flight control of the drone group. The control station layer establishes a connection by using WebSocket and the cloud platform, and simultaneously accesses and controls multiple drones and executes cooperative flight algorithms through the two interfaces of WebSocket and Restful.

At the same time, the workflow engine provides end-users with a simple visual UAV mission planning and trajectory control drawing method, which can be used to achieve remote collaborative flight control of multiple UAV terminals on the control side by dragging.

3 System Simulation and Experiments

In the evaluation and verification stage, by using the Gazebo simulation platform and the RotorS [12] extension package for simulation verification, we can design virtual rotor drones, fixed-wing drones, etc., without any hardware, and through the ROS system communicate with MAVLink to make up for the lack of the drones for simulation.

3.1 Simulation Environment

The simulation verification adopts SITL (Software In The Loop) simulation mode. RotorS is a ROS-based MAV drone simulator, which can provide us with AscTec Hummingbird, Pelican, Firefly, and other drone models, and allows different types of sensors to be mounted on the drone, such as IMU, VI-Sensor. The method to start RotorS and use it for SITL simulation is as follows:

```
roslaunch rotors_gazebo mav_hovering_example.launch mav_name:=
fireflyworld_name:=basic
```

After the simulation environment is started, we communicate with the topics exposed by the simulation environment through the message subscription mechanism provided by ROS to obtain the data of the drone in the simulation environment and control the drone accordingly. Figure 2 shows the simulation environment Interface.

The number of drones needs to be defined in the startup file in the XML format of the simulation environment. The definition is as follows. A group defines the attributes of a drone:

```
<group ns = "$ (arg mav_name) 1" >
...
< /group >
```

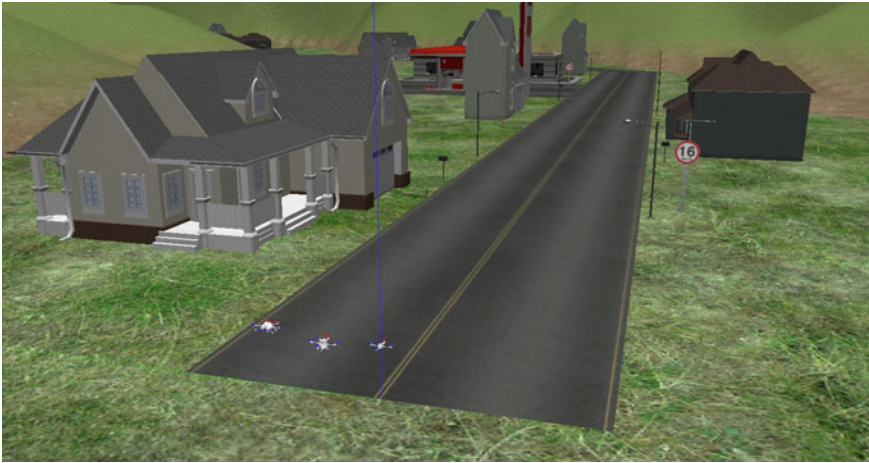


Fig. 2 Simulation environment

3.2 Control Station Interaction

The control station is based on Web technology and is developed using Java and Spring MVC. Figures 3 and 4 show that the control station is a single UAV flight control and flight data acquisition method.

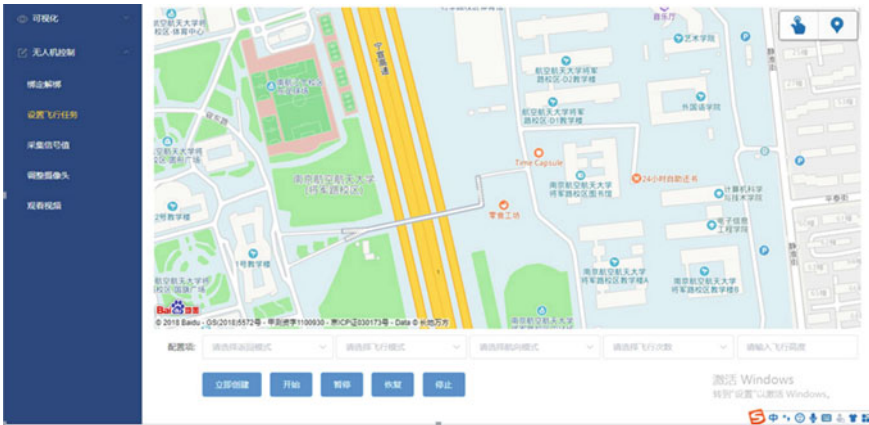


Fig. 3 UAV flight control

As shown in Fig. 3, we can set waypoints and missions for drones according to the map. However, after the drone is connected to the system, the status of the drone will be visually displayed in the control station.

As shown in Fig. 4, we can issue commands from the console to obtain the sensor data mounted by the drone and make a visual display.

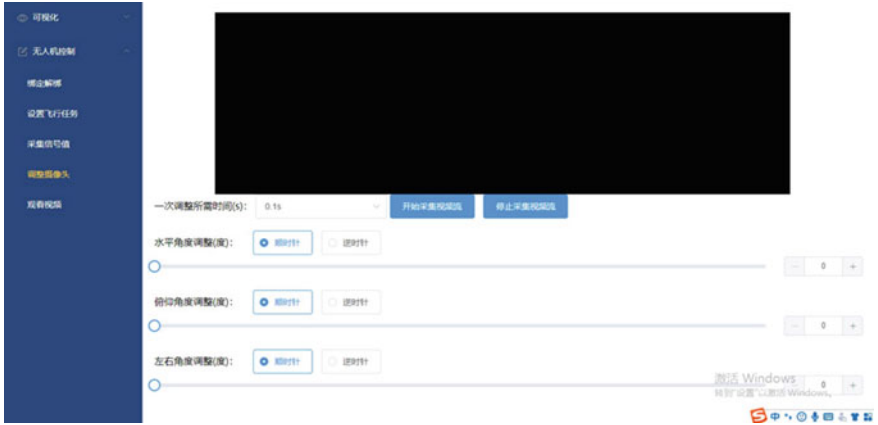


Fig. 4 UAV data obtain

For the control of multiple drones, we choose to use the workflow engine to control the drone swarm and use the drag and drop method to simplify the way operators control the drone group. Figure 5 shows how the drone cluster can be controlled by the workflow engine to construct a flowchart of the drone cluster.



Fig. 5 UAV-swarm workflow control

4 Conclusion

In this article, we propose an architecture that differs from previous UAV group control platforms. The architecture originated from the idea of the interconnection of all things in the Internet of Things, integrating drones into the Internet of Things and cloud robots, and allowing users to control multiple drones simultaneously through the workflow

engine. At the same time, we also designed a web-based console to verify the feasibility and effectiveness of the architecture.

However, due to the complexity of IoT platforms and unmanned systems, there are still some problems with the proposed architecture. First, the architecture uses a part of open-source software, which cannot ensure the security of the architecture. At the same time, although the process engine can efficiently control multiple drones, they have not designed algorithms for autonomous collaborative flight and collaborative mission completion, leaving only interfaces for future improvements and improvements.

Acknowledgements. This work is supported by the Aeronautical Science Foundation of China under Grant 20165515001, the National Natural Science Foundation of China under Grant No. 61402225, State Key Laboratory for smart grid protection and operation control Foundation, and the Science and Technology Funds from National State Grid Ltd. (The Research on Key Technologies of Distributed Parallel Database Storage and Processing based on Big Data).

References

1. Wei J, Hong-Li GE, Hua-Qiang DU et al (2009) A review on unmanned aerial vehicle remote sensing and its application. *Remote Sens Inf* 24(1):88–92
2. Atzori L, Iera A, Morabito G (2010) The Internet of Things: a survey. *Comput Netw* 54(15):2787–2805
3. Kuffner J (2010) Cloud-enabled humanoid robots. In: *Proceedings of the 2010 IEEE international conference on humanoid robots*. Nashville, USA, pp 55–62
4. Arumugam R, Enti VR, Liu B et al (2010) DAVinCi: a cloud computing framework for service robots. In: *Proceedings of the 2010 IEEE international conference on robotics and automation*. Alaska, USA, pp 3084–3089
5. Hunziker D, Gajamohan M, Waibel M et al (2013) Rapyuta: the RoboEarth cloud engine. In: *IEEE international conference on robotics and automation*. Karlsruhe, Germany, pp 436–445
6. Gharibi M, Boutaba R, Waslander SL (2016) Internet of drones. *IEEE Access* 2016(4):1145–1163
7. Hong C, Shi D (2018) A Cloud-based control system architecture for multi-UAV. In: *The 3rd international conference*
8. Vozmediano RM, Montero RS, Llorente IM (2012) IaaS cloud architecture: from virtualized datacenters to federated cloud infrastructures. *IEEE Computer Society Press*
9. Sun Microsystems (1988). RPC: remote procedure call protocol specification. RFC Editor
10. Borthakur D (2008) HDFS Architecture guide. Hadoop apache project, vol 53, issue 1–13, p 2
11. Shvachko K, Kuang H, Radia S (2010) The hadoop distributed file system. In: *Proceedings of the IEEE symposium on MASS storage systems and technologies*. Incline Village, USA, pp 1–10
12. Furrer F, Burri M, Achtelik M et al (2016) RotorS—a modular Gazebo MAV simulator framework. *Robot operating system (ROS): the complete reference*, vol 1. Springer (2016)