SATBAYEV UNIVERSITY

Computing & Engineering

Volume 1 (2023), Issue 4, 12-18

https://doi.org/10.51301/ce.2023.i4.03

Research on ROS-oriented UAV virtual simulation control

Cao Yuning*, Fang Ding

Institute of Automation and Information Technologies, Satbayev University, Almaty, Kazakhstan

**Corresponding author: Cao.Y@stud.satbayev.university*

Abstract. With the rapid advancement of robotics technology, quadcopters have been widely used in military, commercial and other fields. Their ability to perform high-risk missions, keep personnel safe and significantly reduce operating costs makes them an indispensable tool. Although quadcopters perform well in terms of performance, they are expensive to manufacture and their vital electronic components are prone to damage. In this case, the application of simulation technology is particularly important. Virtual simulation technology can not only speed up the development cycle of UAV systems and reduce costs, but also help us discover and solve potential problems in advance, which is very important for students in related majors in universities.

Keywords: *UAV development, ROS robots, virtual simulation, drone simulation, educational application*.

1. Introduction

This article mainly studies the virtual simulation control of UAV in ROS (Robot Operating System) environment. This study first constructed a three-dimensional physical simulation model of a quad-rotor UAV in the Gazebo simulation environment. Then, through the MAVLink communication protocol, a wireless connection is established between the Pixhawk flight control and the QGroundControl ground station to achieve remote control and data transmission. In addition, this article also uses the XTDrone platform to verify and run the HectorSLAM algorithm to further enhance the positioning and navigation capabilities of the drone. Finally, this study tested key functions such as UAV flight control and route planning through simulation, trying to verify the effectiveness and practicability of the proposed method.

1.1. Exploration of quadcopter coordinate system and flight principles

1.1.1 Establishment of coordinate system and reference system

A coordinate system is a mathematical tool referenced to describe the position of an object relative to a reference system. A reference system is a reference object selected to describe the motion of an object. This reference object is generally called a reference system. In the dynamics and control theory of quadcopter UAVs, two coordinate systems are usually used: the ground coordinate system (or global coordinate system, inertial coordinate system) and the body coordinate system.

1.1.2 Ground coordinate system

The ground coordinate system is a fixed reference system, usually with the earth's surface or the take-off point as the reference origin. In the ground coordinate system, we define three mutually perpendicular axes: X-axis, Y-axis and Z-axis. The X and Y axes are usually in the plane of the ground, with the Z axis pointing upward perpendicular to the ground. In this coordinate system, the position of the drone can be represented by its coordinates on three axes, providing a global position reference (Figure 1).

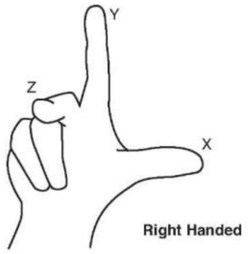


Figure 1. Right-hand coordinate system

1.1.3 Body coordinate system

The body coordinate system is a reference system that moves with the drone, and its origin is located at the center of mass of the drone. In the body coordinate system, three main axes are defined: the forward axis (usually consistent with the forward direction of the UAV), the roll axis (perpendicular to the forward axis, usually located in the wingspan direction of the UAV) and the pitch axis (perpendicular to the UAV's wingspan direction). planes of forward and roll axes). The body coordinate system is used to describe the rotational movements of the drone relative to its own center of mass, such as yaw, pitch and roll (Figure 2).

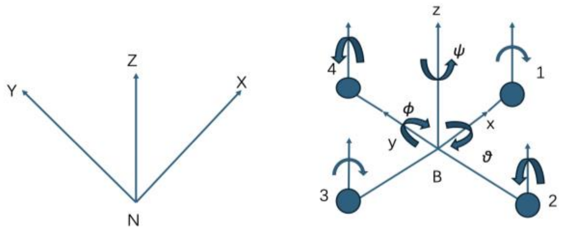


Figure 2. Two coordinate systems

1.2 Body structure and flight principles

A quadcopter drone has four rotors, which are symmetrical to each other and are distributed in the front, rear, left and right directions of the body. As shown in Figure 3 below:

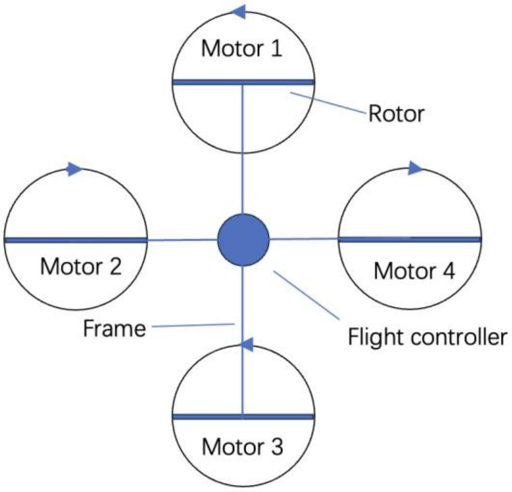


Figure 3. Quadcopter UAV body structure

The real-time attitude and real-time position transformation of the quadcopter UAV is achieved by adjusting the speed of the four motors. When the speed of the motor [5] changes, the speed of the quadcopter drone's rotor changes. Since the rotational speed of each rotor is different, the lift generated is also different. Through the transformation of different lift forces, the attitude and position of the body change.

Vertical movement: When the quad-rotor drone increases the [5] same speed of the four motors at the same time, the four rotors increase the same lift. If the total lift generated is greater than the gravity on the body, the drone will rise vertically. On the contrary, when it reduces the same speed of the four motors at the same time, the total lift generated is less than the gravity on the body, and it will fall vertically. If the total lift generated by the UAV [5] is equal to the gravity experienced when there is no external interference, then the quad-rotor UAV is hovering horizontally [5].

Yaw motion: As mentioned earlier, the quadcopter UAV uses 2 forward propellers and 2 reverse propellers to offset the torque generated by the rotor during rotation and allow the UAV to maintain a stable flight. Adjacent rotor propellers are different, and the motor rotation directions on the diagonal are also different. The torque generated by the rotor is related to the rotation speed of the rotor itself. When the four motor speeds are the same, the torques generated by the four rotors cancel each other out, and the quad-rotor drone does not rotate; when the four motor speeds are not exactly the same, the unbalanced Torque causes the quadcopter to turn. In Figure 2, when the speeds of motor 2 and motor 4 increase and the speeds of motor 1 and motor 3 decrease, the reaction torque of rotor 2 and rotor 4 on the fuselage is greater than the reaction torque of rotor 1 and rotor 3 on the fuselage, and the machine The body rotates around the z-axis under the action of the excess reaction torque, realizing the yaw movement of the aircraft, and the steering is opposite to the steering of motor 2 and motor 4.

Pitching motion: The speed of motor 1 decreases (increases), the speed of motor 3 increases (decreases), and the speeds of motor 2 and motor 4 remain unchanged [5]. As the lift of rotor 3 increases (decreases) and the lift of rotor 1 decreases (increases), the resulting unbalanced moment causes the fuselage to tilt, generating a forward (backward) component and flying forward.

Rolling motion: The principles of rolling motion and pitching motion are the same. The speed of motor 2 decreases (increases), the speed of motor 4 increases (decreases), and the speeds of motor 1 and motor 3 remain unchanged. As the lift of rotor 4 increases (decreases) and the lift of rotor 2 decreases (increases), the resulting unbalanced moment causes the fuselage to tilt, creating a lateral component and causing sideways flight.

The four flight motion modes of the quadcopter drone can be superimposed on each other to complete the specified flight mission and desired position through complex flight motions [5].

2. UAV virtual simulation platform and method

The purpose of this section is to show how to use these tools and platforms to build a simulation environment, and how to verify the design, control strategy and performance of the UAV system through virtual simulation.

2.1 Construction of simulation system

The operating system mainly used in this article is based on Ubuntu20.04, and the ROS version is Noetic.

2.1.1 Gazebo quadcopter UAV 3D modeling

Gazebo is an open source robot simulation software widely used in robot research, development and testing. It provides a distinctive 3D simulation environment where robots can interact in complex indoor and outdoor environments. Gazebo can simulate accurate physical environments, including lighting, shadows, terrain, and physical properties of objects (such as mass, friction, collision, etc.). Through Gazebo, we can easily build suitable three-dimensional scenes and simulated objects (Figure 4).

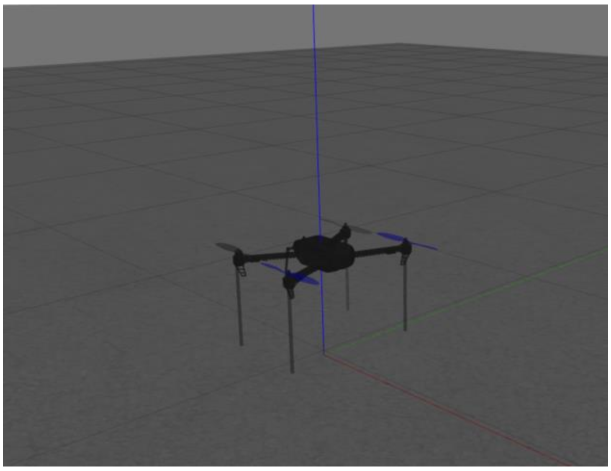


Figure 4. Gazebo simulated drone

2.1.2 PX4 autopilot

PX4 is a professional autopilot [1]. Developed by world-class developers from industry and academia and supported by an active global community, it powers a variety of vehicles from racing and cargo drones to ground vehicles and submersibles.

Figure 5. PX4 flight control software

PX4 (Figure 5) is an independent open source project dedicated to providing low-cost, high-performance flight control for academia, amateurs and industrial communities. Gazebo is a dynamics simulation software with a built-in physics engine, directly integrated into ROS (Robot Operating System) development environment, both are commonly used tools by smart [2] drone developers. In the research process related to the reinforcement learning training of quad-rotor drones based on PX4 flight control in the Gazebo simulation environment, an interface program is needed to enable interaction with the simulation environment during the reinforcement learning training process.

2.1.3 XTDrone Platform

XTDrone provides a comprehensive drone simulation environment based on PX4, ROS and Gazebo. The XTDrone platform is compatible with a variety of drone types, including multi-rotor aircraft (four-axis or six-axis), fixed-wing aircraft, etc. At the same time, the XTDrone platform also supports other types of unmanned systems, including unmanned vehicles, unmanned ships and robotic arms. In the article, I mainly used the XTDrone platform to test and verify the algorithms to ensure that these algorithms can be correctly transferred to actual operating drones.

2.1.4 MAVLink & MAVROS

The MAVLink protocol supports two-way communication through wireless channels, which can realize wireless two-way communication between the UAV and the UAV ground control station QGC, so as to realize the control of the UAV by the QGC and the return of UAV data. The data frame sent through the MAVLink protocol is a data frame with a frame header of FE and a frame tail of 16-bit CRC check code. Each frame has a sending packet sequence from 0 to 255. Repeat counting is used to monitor packet loss during the transmission process.

The main function of MAVRO is to provide information exchange between the external computer and the PX4 flight control. MAVROS It is a software package used in the ROS environment, which allows the ROS system to communicate with the drone through the MAVLink protocol. MAVROS usually acts as a bridge between ROS and MAVLink[3], converting messages in the MAVLink protocol into message types that ROS can understand. The transmission of message content under the MAVLink protocol is too complex. Data messages written in C++ can be directly communicated with the Pixhawk flight controller through MAVROS. MAVROS will complete the conversion of the messages under C++ into the form of the MAVLink protocol and send them to the flight controller. And because it is under the large framework of the ROS system, it also brings convenience to other messages that need to access PX4. You only need to subscribe to the message name of the required message to access the data in PX4. Data is also published through this message mechanism.

```
roscd /opt/ros/noetic/share/mavros/launch
sudo vim px4.launch
```

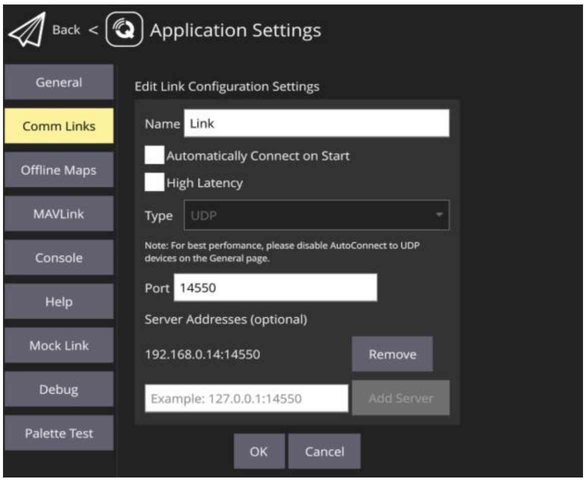
Through the above settings, you can modify the MAVROS configuration file of ROS based on the actual network IP.

```
<launch>
<!-- vim: set ft=xml noet : -->
<!-- example launch script for PX4 based FCU's -->
<arg name="fcu_url" default="/dev/ttyACM0:57600" />
<arg name="gcs_url" default="udp://:14550@" />
……
```

The master node allows disabling the GCS agent by setting an empty URL. Running example (autopilot connected via USB at 57600 baud, GCS running on host, port 14550).

2.1.5 QGC ground station communicates with PX4 and Gazebo

QGroundControl provides complete flight control and vehicle setup for PX4 or ArduPilot powered vehicles (Figure 6).



Edit Link Configuration Settings

Name Link

☐ Automatically Connect on Start

☐ High Latency

Type UDP

Note: For best perfomance, please disable AutoConnect to UDP devices on the General page.

Port 14550

Server Addresses (optional)

192.168.0.14:14550

Example: 127.0.0.1:14550

Figure 6. QGC ground station UDP communication settings

This article uses QGC to provide aircraft parameter settings and flight environment for the PX4 powered aircraft.

After modifying the PX4.launch file in MAVROS above, QGC can communicate with PX4 and Gazebo through UDP.

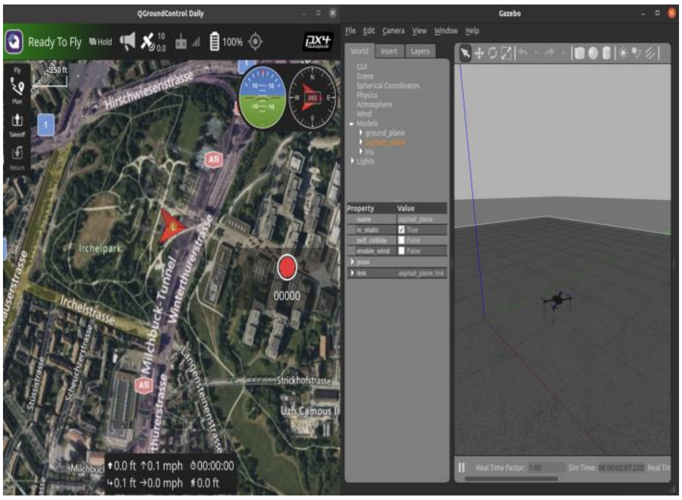


Figure 7. UAV flight simulation in QGC and Gazebo

2.2 The interaction between PX4 flight control and external programs and simulation environment

In software-in-the-loop simulation (SITL), the PX4 can interact with external programs through MAVLink messages. At the same time, PX4 also interacts with simulation environments (such as Gazebo) through MAVLink messages. When interacting with Gazebo, usually, PX4 issues control instructions to drive the drone's propeller motor to Gazebo, and Gazebo issues to PX4 the information sensed by various sensors on the drone model (simulator) in the simulation environment.

When using the ROS development environment for programming, since each ROS node (external control program) interacts through ROS topics and services, it is also necessary to use the MavROS function package to convert between MAVLink messages and ROS topics and services. The message flow is shown in Figure 8.

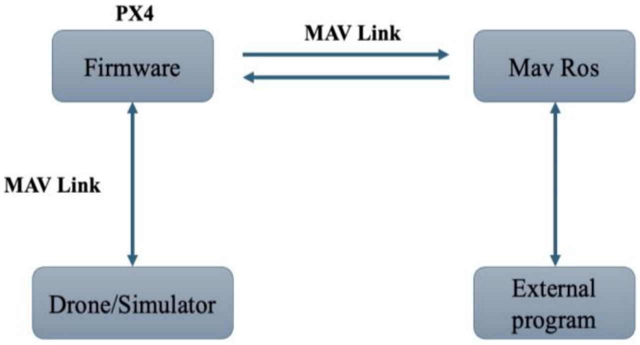


Figure 8. Message flow diagram from external program to simulator

2.3. Implementation of HectorSLAM algorithm

In this chapter, I mainly use the HectorSLAM algorithm to implement the SLAM (Simultaneous Localization and Mapping) process of two-dimensional laser scanning in the UAV virtual simulation system. HectorSLAM is able to provide accurate SLAM solutions because it does not rely on odometry data, and it performs well in applications in two-dimensional space. This chapter first outlines the working principle and characteristics of the HectorSLAM algorithm, and then details its integration and implementation process in the Gazebo simulation environment.

2.3.1 Overview of HectorSLAM algorithm

HectorSLAM is a SLAM technology based on lidar data. It can create an environment map and locate its own position in real time. It is especially suitable for robots with no or very limited odometry information. This algorithm mainly consists of three core components.

2.3.2. Scan matching

The robot's position is estimated using matching information between the current laser scan data and the established map. Use the current frame and existing map data to construct the [4] error function, and use the Gauss Newton method to obtain the optimal solution and deviation. Its job is to convert laser points into raster maps. All laser points at time t can be transformed into raster maps, which means the matching is successful.

2.3.3. Map update

In HectorSLAM, map updates are based on the latest estimated position of the robot. This process ensures that the map is continuously updated as the robot moves, reflecting the latest state of the environment.

The process is mainly divided into two parts. The data integration part mainly integrates the latest scanning data obtained from lidar into the current map. This step typically involves converting the lidar data from polar to Cartesian coordinates and adjusting the position of these data based on the robot's current estimated position. Next, the grid needs to be updated. For each grid in the map, the status of the grid is updated based on the position of the laser scanning point. Generally, if a scan point falls within a grid, the probability of occupancy of that grid increases; if a grid on the scan path is not hit, its probability of occupancy decreases.

2.3.4. Posture optimization

Improve the accuracy of position estimation and the quality of the map through pose optimization algorithms (such as Gauss-Newton method). Gauss Newton is one of the simplest methods in the optimization algorithm. The goal is to find Δx so that $f(x+\Delta x)$ is as small as possible. Its idea is to perform a first-order Taylor expansion of the objective function $f(x)$.

$$f(x+\Delta x) \approx f(x) + J(x)\Delta x \tag{1}$$

Here $J(x)$ is the derivative of $f(x)$ with respect to x, which is actually an $m \times n$ matrix and a Jacobian matrix. Our goal is to minimize $\left| f(x+\Delta x) \right|$. To find Δx we need to solve a linear least squares problem:

$$\Delta x^{*} = \arg\min_{\Delta x} \frac{1}{2} \| f(x) + J(x)\Delta x \|^{2} \tag{2}$$

What is the difference between this equation and the previous one? According to the extreme value condition, the above objective function is derived with respect to Δx, and the derivative is zero. Since what is considered here is the derivative of Δx (rather than x), we will finally get a linear equation. To do this, first expand the square term of the objective function:

$$\begin{aligned} &\frac{1}{2} \| f(x) + J(x)\Delta x \|^{2} = \frac{1}{2} (f(x) + J(x)\Delta x^{T} (f(x) + J(x)\Delta x) = \\ &\frac{1}{2} (\| f(x) \|_{2}^{2} + 2 f(x)^{T} J(x)\Delta x + \Delta x^{T} J(x)^{T} J(x)\Delta x) \end{aligned} \tag{3}$$

Find the derivative of the above equation with respect to Δx and make it zero:

$$2J(x^{T}) f(x) + 2J(x)^{T} J(x)\Delta x = 0 \tag{4}$$

The following system of equations can be obtained:

$$J(x^{T}) J(x)\Delta x = -J(x)^{T} f(x) \tag{5}$$

Note that the variable we want to solve for is Δx, so this is a linear system of equations. We call it the incremental equation, which can also be called Gauss Newton equations or normal equations. We define the coefficient on the left as H and the right as g, then the above formula becomes:

$$H\Delta x = g \tag{6}$$

3. SLAM simulation process and result analysis

In this chapter, I will explain how to implement two-dimensional laser SLAM and save maps through virtual simulation of drones.

```
roslaunch px4 indoor3.launch
roslaunch hector_slam_launch hector_slam_xtdrone.launch
```

Start the simulation environment first, and then start HectorSLAM.

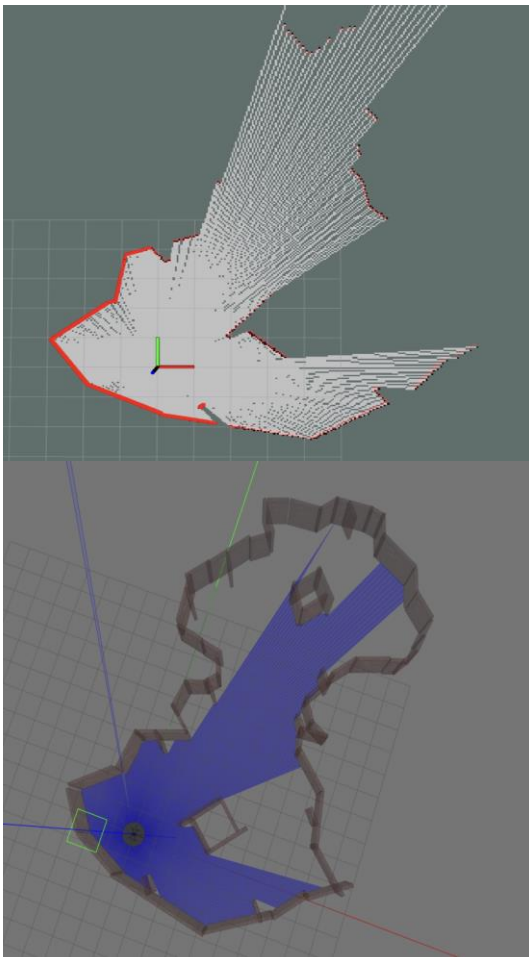


Figure 9. Two-dimensional laser SLAM scanning mapping

The figure below is the tf tree of the entire program, describing the transformation between various coordinate systems.

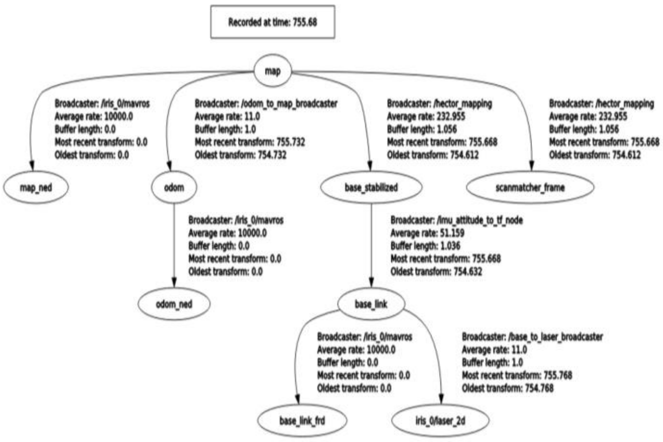
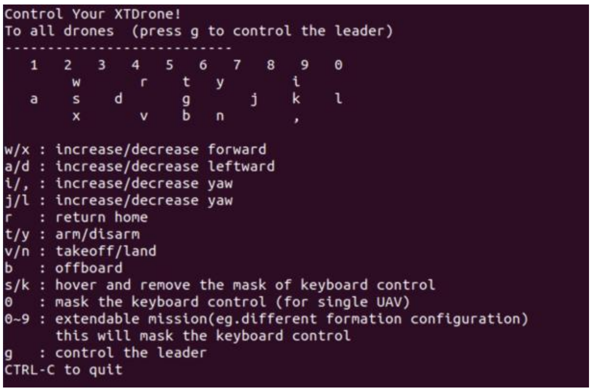


Figure 10. Transformation between various coordinate systems

During virtual simulation, a keyboard is needed to control the flight of the drone. Why does the XTDrone platform integrate everything? You only need to start the keyboard control script to realize virtual simulation and control drone flight. As shown in Figure 10 below, the drone control script. By controlling the movement of the drone, two-dimensional SLAM scanning is achieved.



```
Control Your XTDrone!
To all drones  (press g to control the leader)
---------------------------
   1   2   3   4   5   6   7   8   9   0
        w       r   t   y       i
   a    s   d       g       j   k   l
        x       v   b   n       ,

w/x : increase/decrease forward
a/d : increase/decrease leftward
i/, : increase/decrease yaw
j/l : increase/decrease yaw
r   : return home
t/y : arm/disarm
v/n : takeoff/land
b   : offboard
s/k : hover and remove the mask of keyboard control
0   : mask the keyboard control (for single UAV)
0~9 : extendable mission(eg.different formation configuration)
      this will mask the keyboard control
g   : control the leader
CTRL-C to quit
```

Figure 11. Keyboard control of drone flight

As can be seen from Figure 11, after setting up the UAV to take off, the UAV can control the flight according to the planned path and successfully realize the two-dimensional laser SLAM simulation.

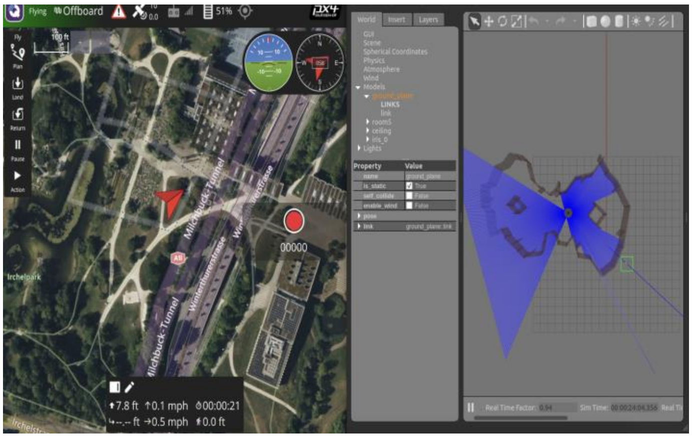


Figure 12. Two-dimensional laser SLAM simulation

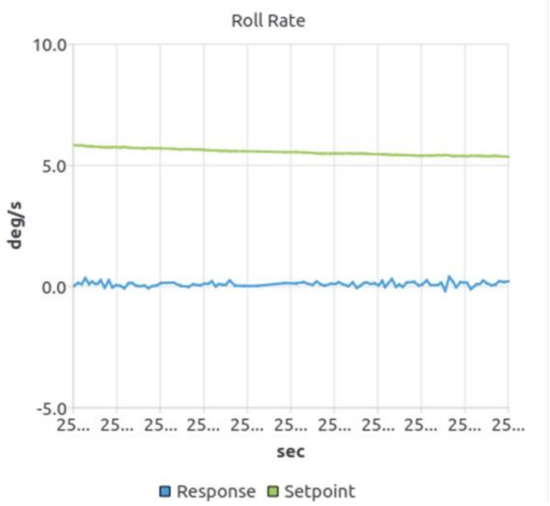


Figure 13. Roll rate response when the drone simulation test is initiated

The situation after the drone simulation test is started. The response curve begins to show fluctuations, the system is responding to changes in the set point, and the response curve fluctuates around the zero point.

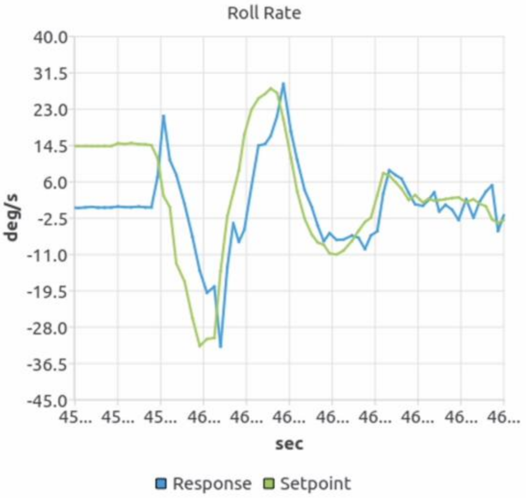


Figure 14. Roll rate response after the actual takeoff of the drone

Figure 14 shows the roll rate response of the UAV after actual takeoff. It can be seen that there is a significant difference between the response curve and the set point curve during the take-off phase, and it remains flat after take-off is stable.

4. Conclusions

This paper mainly describes the process and results of using the Gazebo simulation platform to control and navigate a quadrotor UAV in the ROS environment. By establishing an accurate three-dimensional physical simulation model and realizing wireless connections supported by the MAVLink communication protocol, this article not only achieves efficient data transmission between the Pixhawk flight control and the QGroundControl ground station, but also ensures the real-time and reliability of remote control. In addition, by applying the HectorSLAM algorithm, this research further enhances the positioning and navigation functions of UAVs, enabling UAVs to perform accurate map construction and path planning in a virtual simulation environment. Simulation test results show that the UAV demonstrates a high degree of accuracy and stability when performing flight control and route planning tasks. The research results of this article provide an effective methodology for the virtual simulation control of quad-rotor UAVs. Through the comprehensive use of modern sensor fusion technology and advanced SLAM algorithms, the operational performance and performance of the UAV system in the virtual simulation environment are introduced. safety. Future work will focus on optimizing the control algorithm, expanding application scenarios, and further verifying the robustness and adaptability of the virtual simulation system in complex environments.

References

[1] Huang Rongyue. (2021). Design of intelligent kit drone based on OpenCV. *Information Technology and Informatization, (6), 228-231*

[2] Sun Xianglong & Liang Yangang. (2022). Design and implementation of reinforcement learning training simulation environment interface based on Gazebo and PX4. *Electronic Technology and Software Engineering, (5), 68-71*

[3] Lin Binwei, Zhang Wenjuan & Li Qinglin. (2022). Intelligent cluster design for industrial training competition based on ROS system. *Information Technology and Informatization, (10), 200-204*

[4] Liu Zhicheng, Wang Hualong & Ma Xinglu. (2024). Review of laser real-time positioning and mapping algorithms. *Computer Measurement and Control, (3), 1-8*

[5] Zhou, L., Pljonkin, A. & Singh, P. (2022). Modeling and PID control of quadrotor UAV based on machine learning. *Journal of Intelligent Systems, 31(1), 1112-1122*. https://doi.org/10.1515/jisys-2021-0213

ROS-бағдарланған UAV виртуалды симуляциясын басқару бойынша зерттеулер

Цао Юнин[*], Дин Фан

Автоматика және ақпараттық технологиялар институты, Satbayev University, Алматы, Қазақстан

**Корреспонденция үшін автор:* Cao.Y@stud.satbayev.university

Аңдатпа. Робототехниканың қарқынды дамуының арқасында квадрокоптер әскери, коммерциялық және басқа салаларда кеңінен қолданыла бастады. Олардың қауіптілігі жоғары тапсырмаларды орындау, персоналдың қауіпсіздігін қамтамасыз ету және операциялық шығындарды айтарлықтай азайту қабілеті оларды таптырмас құрал етеді. Квадрокоптер өнімділік тұрғысынан жақсы жұмыс істегенімен, оларды өндіру қымбатқа түседі және олардың маңызды электрондық компоненттері зақымдануға бейім. Бұл жағдайда модельдеу технологияларын пайдалану ерекше маңызды. Виртуалды имитациялық технология UAV жүйелерінің даму циклін жылдамдатуға және шығындарды азайтуға ғана емес, сонымен қатар ықтимал проблемаларды алдын ала анықтауға және шешуге көмектеседі, бұл колледждер мен университеттердегі сабақтас салалардың студенттері үшін өте маңызды.

Негізгі сөздер: *UAV әзірлеу, ROS робот, виртуалды модельдеу, UAV симуляциясы, білім беру қолданбасы.*

Исследование управления виртуальной симуляцией БПЛА, ориентированного на ROS

Цао Юнин*, Дин Фан

Институт автоматики и информационных технологий, Satbayev University, Алматы, Казахстан

**Автор для корреспонденции: Cao.Y@stud.satbayev.university*

Аннотация. Благодаря быстрому развитию робототехники квадрокоптеры стали широко использоваться в военной, коммерческой и других областях. Их способность выполнять миссии повышенного риска, обеспечивать безопасность персонала и значительно снижать эксплуатационные расходы делает их незаменимым инструментом. Хотя квадрокоптеры хорошо работают с точки зрения производительности, их производство дорого, а их жизненно важные электронные компоненты подвержены повреждениям. В этом случае применение технологий моделирования особенно важно. Технология виртуального моделирования может не только ускорить цикл разработки систем БПЛА и снизить затраты, но также помочь нам заранее обнаружить и решить потенциальные проблемы, что очень важно для студентов смежных специальностей в колледжах и университетах.

Ключевые слова: *разработка БПЛА, робот ROS, виртуальное моделирование, моделирование БПЛА, образовательное приложение.*

Received: 23 August 2023
Accepted: 15 December 2023
Available online: 31 December 2023