UAV Solution of Volcanic Exploration

Group 4, Project "*Icarus*"

Honghao Pan (lz21202@bristol.ac.uk), Lijing Zhou (qt21142@bristol.ac.uk), Yanfeng Zhu (gc21878@bristol.ac.uk), Yilin Duan(pk21489@bristol.ac.uk). Jiachen Yu (dz21232@bristol.ac.uk), Wenda Li (oy21104@bristol.ac.uk), Ye Tao (pa21876@bristol.ac.uk),

1 Abstract

Nowadays, with the rapid development of aerial robotics technology, Unmanned Aerial Vehicles (UAVs), as aerial robots, are widely used in both military and civilian fields. In this study, we will discuss the possibility of using UAVs for volcanic exploration using provided technologies. By writing programs for flight control, image recognition, path planning and etc., and designing User Interface (UI), the UAV can independently complete a series of tasks such as takeoff, navigation flight, landing and so on. By improving the design, challenges such as obstacle avoidance, flying around the no-fly zone and emergency control can be completed.

2 Table of Contents

3 Technology Overwiew

In this project, the following technologies are mainly used: *ROS2* for simulation, *A* Algorithm* for path planning, *OpenCV* library for Image recognition, *Foxglove Studio* for User Interface design. By using the technology mentioned above, this project can achieve: autonomous take-off, navigation flight, landing and other tasks of UAV, without modification of UAV. There is no need for personnel to participate in the control during the flight. Users only need to set the starting point, end point and flight altitude before the flight.

4 State Machine

State machine of UAV is as follow.

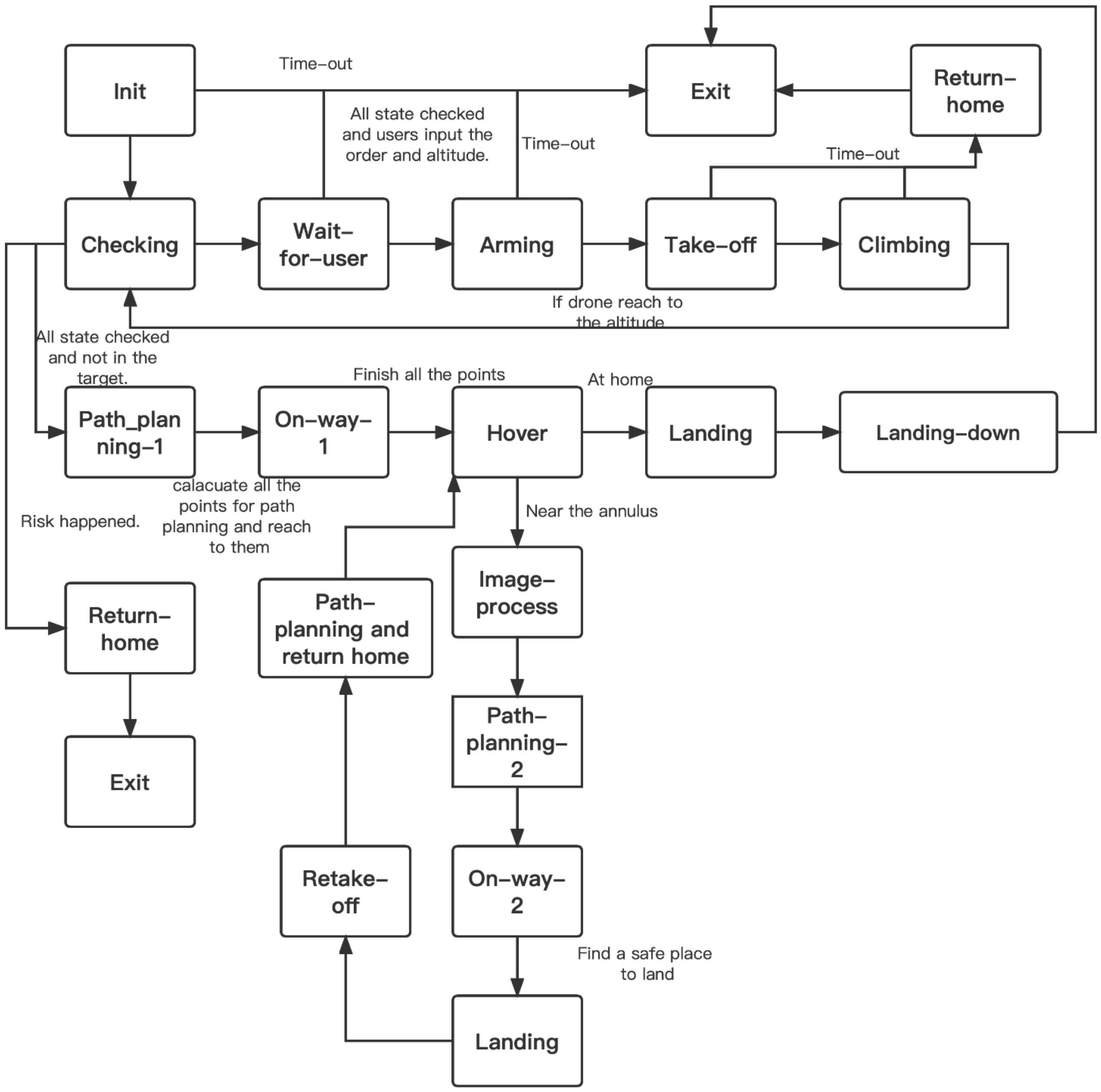


Figure 4.1 State machine of UAV flight.

Table 4.1 Actions for State machine

State	Action
Init	UAV will do the connection by the *Rosbridge* and update data to *Foxglove*.
Check	A class called *Risk Management* will be checked to test three main indexes of error: whether the drone fly out of the edge, whether the drone fly into No-fly zone, and whether the battery is less than 30 percent. This state will have several brunches.
Wait-for-user & Arming	At this state, users will need to enter a command called 'Arming Checked' to activate motors, and altitude for flying is asked to fill in. Then UAV will attempt to rotate its four wings. If UAV receives no command after a certain period of time, UAV will automatically lockdown for safety and enter 'Exit' state.
Take-off & Climbing & Re-take-off	At this state, UAV will take-off and gain altitude until reaches the altitude for flying.
Path_planning-1 & Path-planning and return home	UAV will do path planning for waypoints towards the area of interest.
On-way-1	UAV will fly by waypoints towards the vicinity of the area of interest (annulus).
Hover	At this state, UAV will hover and enter several states according to its latitude and longitude.
Land & Landing-down	At this state, UAV will land and stop rotating wings.
Image-process	At this state, UAV will run image recognition process.
Path-planning-2 & on-way-2	Based on the information from 'Image-process' state, UAV will reach the target position and land.
Exit	End of program. At this state, UAV will power down.

5 Path Planning

During the whole process of flight, from the take-off point to the vicinity of the target point, path planning avoiding no-fly zone is needed. According to the state machine, waypoints are calculated on board. The algorithm we use in this process is called *A* Algorithm* [1].

5.1 Advantage of *A* Algorithm*

A Algorithm* is a classic path planning algorithm on a given environment. Compared to other algorithms, *A* Algorithm* can create the path with the smallest cost of the length and time, which is thought as the basis for the heuristic determination of minimum cost path. Other algorithms of path planning like *VFH* [2], *VFH+* [3] and *BUG* [4] are focusing on the unknown map and need to do the new computation according to the changes of the environment, which means it requires high mathematical ability. These algorithms are built on a strange environment. In fact, in our map, as we know that the entire map is made up of a flight zone defined by 4 fixed points at given latitude and longitude, a no-fly zones also defined by fixed points, as well as departure points (51.4234178N, 2.6715506W) and flight terminations (51.4219206N, 2.6687700w), enough information of the map is collected. *A* Algorithm* doesn't need a complex computation and can create a smallest path with high efficiency.

5.2 Implementation for Path Planning

By using *A* Algorithm* and information given, UAV calculates the coordinates of waypoints by flight step and safe distance to avoid obstacles, both set by users. As Figure 5.1 shows, *A* Algorithm* will create several waypoints like O_1 to O_n, to guide UAV to fly points by points and reach the position near the area of interest. In fact, parameters can be adjusted to achieve the precise route and shorten the length of the path. In general, the smaller the flight step, the more points of path planning, the higher the accuracy of path planning, the shorter the flight path length of UAV, but it will also increase the amount of calculation.

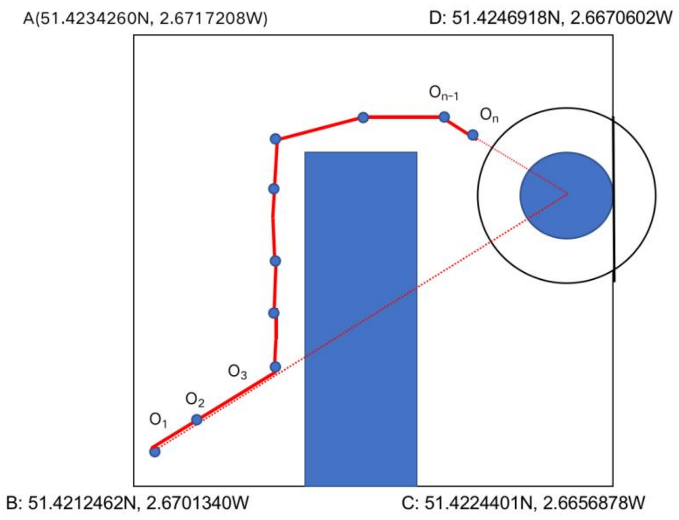

A(51.4234260N, 2.6717208W)
D: 51.4246918N, 2.6670602W
O_{n-1}
O_n
O_3
O_2
O_1
B: 51.4212462N, 2.6701340W
C: 51.4224401N, 2.6656878W

Figure 5.1 Example of diagram for path planning.

The waypoint coordinates are then imported into an array in the form of two coordinate vectors. And by the built-in fly-to function of the system, used to fly UAV from one point to another, UAV flies along the waypoints, and finally approach the vicinity of the area of interest. At this point, the position where UAV stops is near the area of interest, but does not enter the area of interest.

6 Image Processor

For UR7 in the requirements in this project, image recognition is selected to complete the task. Image recognition, refers to the use of computer image processing, analysis and understanding, in order to identify a variety of different patterns of the target and object technology. As this project only needs to identify the colour of the block, and there is only a single common camera, it is not possible to build an accurate map and distance calculation, the general plan is to use the *OpenCV* library to achieve colour recognition and then estimate the actual distance between the UAV and the target colour block according to the height, so that the UAV yaws and completes the task.

6.1 Advantage of *OpenCV*

OpenCV is a cross-platform computer vision and machine learning software library distributed under the *Apache 2.0* license (open source) [5]. It can implement many general algorithms in image processing and computer vision. Furthermore, Furthermore, considering the compatibility of the *ROS 2* platform, *OpenCV* has a mature communication method for cross-platform image format conversion with *ROS 2* called cv2_bridge [6]. The process is as Figure 6.1 shows.

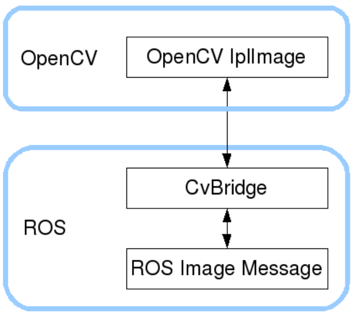


Figure 6.1 Data exchange between *OpenCV* and *ROS 2*

6.2 Implementation for Image Processing

The detailed implementation of image processing is as follow. When the path planning process finish, UAV flies towards the circle in the area of interest. At this point, the image processing system starts to work. First, in the image processor node, raw images from the camera are subscribed and transferred to *OpenCV* via cv_ bridge for image processing. Then, using the functions in the *OpenCV* library, the images are Gaussian filtered and eroded where red and yellow parts of the images are identified and marked. And these images and data for flight control are released to Flight Control Unit (FCU) and User Interface (UI) to complete the task.

7 User Interface (UI) & Simulation

7.1 UI Layout Instruction

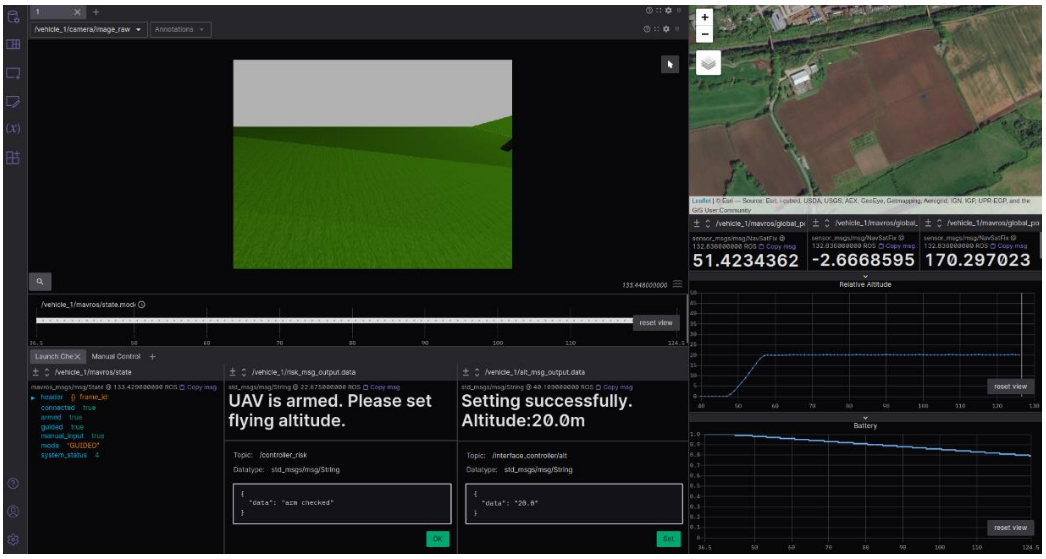
The layout of User Interface (UI) is shown as Figure 7.1. In general, UI is divided into three parts: *Real-time Image Display*, *UAV Control Center*, and *Data Display*.

7.1.1 Real-time Image Display

Real-time Image Display is located at the upper left of UI, which can display real-time image received from onboard camera. Through the image recognition algorithm, marks for volcano and other key areas of concern are superimposed on the real-time image, so that users can observe and focus to key areas at all times.

7.1.2 UAV Control Center

UAV Control Center in the lower left part of UI has tabs integrated according to operation, allowing users to operate UAV in different ways. Currently, there are 'Launch Check' tab and' Manual Control' tab. More tabs will be added in the following development.


/vehicle_1/camera/image_raw
Annotations
/vehicle_1/mavros/state.mode
reset view
Launch Che
Manual Control
/vehicle_1/mavros/state
/vehicle_1/risk_msg_output.data
/vehicle_1/alt_msg_output.data
/vehicle_1/mavros/global_po
Relative Altitude
Battery
OK
Set

```
mavros_msgs/msg/State @ 133.429000000 ROS
▸ header {} frame_id:
  connected true
  armed true
  guided true
  manual_input true
  mode "GUIDED"
  system_status 4
```

UAV is armed. Please set flying altitude.

Topic: /controller_risk
Datatype: std_msgs/msg/String

```
{
  "data": "arm checked"
}
```

Setting successfully. Altitude:20.0m

Topic: /interface_controller/alt
Datatype: std_msgs/msg/String

```
{
  "data": "20.0"
}
```

51.4234362 | -2.6668595 | 170.297023

Figure 7.1 The layout of User Interface

7.1.3 Data Display

On the right part of UI is *Data Display*, which shows important parameters of UAV, including GPS data, real-time map and relative location display. Moreover, *Data Display* also draws charts for UAV's terrain-clearance change during flight and the UAV's battery power change, helping users to monitor and predict UAV's flight status.

7.2 Simulation Details

The current simulation test is mainly based on some key crosslinking communication, such as initialization check, flight altitude setting, etc. Specifically, when the initialization of UAV is completed, UI will display information and a button for users to confirm the completion of startup check. By clicking the button, a valid flight altitude will be required to enter. Then the drone will fly automatically to the target at set altitude and land. By repeating the process of setting altitude, UAV will take off and return to its starting point.

8 Conclusion & Future Plans

8.1 Conclusion

This project has now completed the functions of UAV for autonomous take-off, path planning, navigation flight, image recognition, autonomous, landing and etc.

8.2 Future Plans

Our teams plan to focus on the simulation of flight test under various states in the following stage, as well as more in-depth interactive operation, so as to simulate a more realistic environment and better deal with more complex situations.

In the current path planning scheme, a potential improved field lies in the tune of finding a good parameter to balance the accuracy and the power consumption level, as the more accurate the output of waypoints are, the more overall flight time higher battery consumption there will be. In addition, adding a manual mode to achieve a higher degree of freedom of the operation is also planned. And Improvements will also be made to the UI's page layout and user interaction features.

Reference

[1] Hart, P., Nilsson, N. &amp; Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), pp.100–107.

[2] Ulrich, I. &amp; Borenstein, J., 1998. VFH+: Reliable obstacle avoidance for Fast Mobile Robots. Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), 2, pp.1572–1577.

[3] Borenstein, J. &amp; Koren, Y., 1991. The vector field histogram-fast obstacle avoidance for Mobile Robots. IEEE Transactions on Robotics and Automation, 7(3), pp.278–288.

[4] Xu, Q.-L., Yu, T. &amp; Bai, J., 2017. The mobile robot path planning with motion constraints based on Bug algorithm. 2017 Chinese Automation Congress (CAC), pp.2348–2352.

[5] Anon, Introduction - docs - foxglove studio. Foxglove. Available at: https://foxglove.dev/docs/studio [Accessed March 9, 2022].

[6] Anon, 2020. About. OpenCV. Available at: https://opencv.org/about/ [Accessed March 9, 2022].