Northeastern University
Autonomous Aerial Robotics Team

Yuchao Hu
Northeastern University, Shenyang, China
Jianyu Yang
Northeastern University, Shenyang, China
Yuanning Cao
Northeastern University, Shenyang, China
Fengshuai Yang
Northeastern University, Shenyang, China
Bo Shang
Northeastern University, Shenyang, China

ABSTRACT

In order to complete the IARC competition tasks, Northeastern University autonomous aerial robotics team has developed an indoor autonomous quadcopter system based on the famous AR.Drone system. The aircraft carries an APM autopilot to mitigate the geomagnetic problem of the original system. A lightweight laser scanner is deployed on the top of the airframe to avoid obstacles. The camera is modified to face down as an optical flow sensor. This quadcopter can send videos, lidar data and altitude data to ground control station (GCS). GCS is responsible for handling the data to generate path-planning data and sent them back to the aircraft.

INTRODUCTION

In the seventh generation of IARC competition, aircraft interacts with multi-ground mobile robots, which is called "shepherd action." The aerial robot is "shepherd" and ground mobile robots are "sheep". "Shepherd" rushes "sheep" into their "fold" through physical contact, while avoiding obstacles. Finally, the team who rushes into a maximum number within a time limit wins. This task is divided into two phases, the first phase is verifying that "shepherd" is able to automatically interfere the orientation and direction of ground robots, and the second phase is verifying that "shepherd" is able to compete against another "shepherd".

Our design is divided into two parts to achieve the desired flight navigation and influence the robots: the hardware construction and software development.

In terms of hardware, first of all, we modified the direction of the frontal camera. The frontal camera is configured to face down as an optical flow sensor. An ArduPilot Mega (APM) board is connected to the flight control board to make up for the magnetic problem of the original system.

On software, we combined Robot Operating System with the existing flight control system onboard. The videos captured onboard are transmitted to our ground station. The ground station is responsible for image processing, recognition and path planning.

HARDWARE

The hardware architecture is shown in Figure 1. The system contains an aircraft, a repeater, a ground station running Robot Operating System and several iRobots.

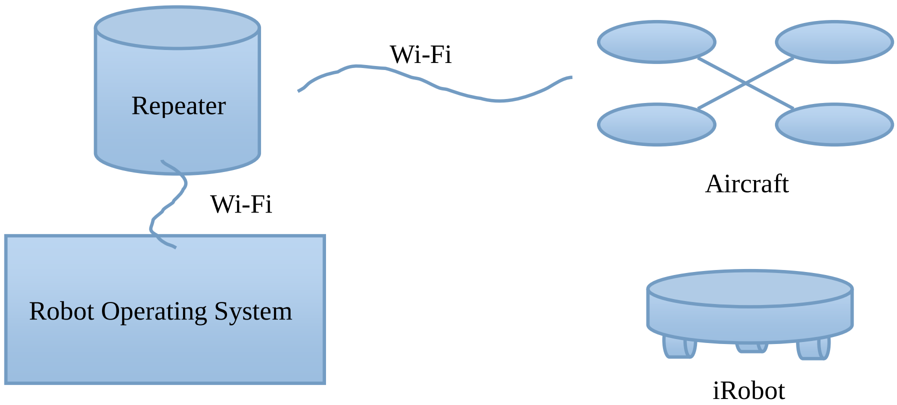

Figure 1. Hardware Architecture

AR.Drone

AR.Drone is employed in our system, which is shown in Figure 2. AR.Drone is a small four-rotor aircraft for commercial use. It is relatively economical, compared with Pelican or Hummingbird by Ascending Technology. Also, AR.Drone has such a high robustness that it can work properly after crashes. It has a elaborately designed shield to fly safely indoors.

Figure 2. AR.Drone

AR.Drone has a 3-axis gyroscope, a 3-axis accelerometer and two cameras. Frontal camera has a resolution of 320 * 240, which is responsible for capturing videos for ground station. The aircraft also has a vertical camera who’s resolution of 176 * 144, which is responsible for optical flow algorithm.

Power System

The aircraft is powered by a lithium battery. To ensure safety, power is connected to the aircraft through a kill switch. An ArduPilot Mega board is connected to the airframe with a USB interface to get power. The power system is shown in Figure 3.

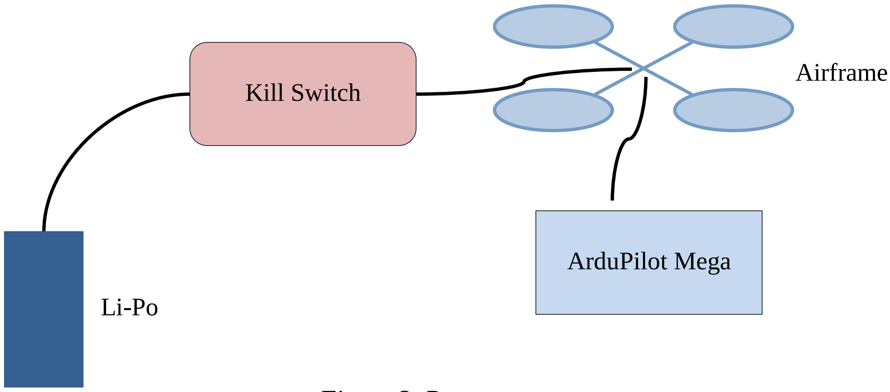

Figure 3. Power system

RoboPeak RPLIDAR

RoboPeak RPLIDAR is a lightweight 360-degree laser scanner. The laser scanner is placed on the top center of the airframe to avoid obstacles, which is shown in Figure 4.

Figure 4. Laser scanner

Camera

The frontal camera is configured to face down as an optical flow sensor, which is shown in Figure 5. A fisheye is placed in front of the camera to expand the field of view. An aluminum fitting is developed to fix the fisheye.

Figure 5. The frontal camera is configured to face down

ArduPilot Mega (APM)

A piece of ArduPilot Mega board is placed under the airframe to provide an alternative source of geomagnetic data. In this way, the direction of the airframe is ensured. The APM board is connected to the flight control board through USB interface, which is shown in Figure 6.

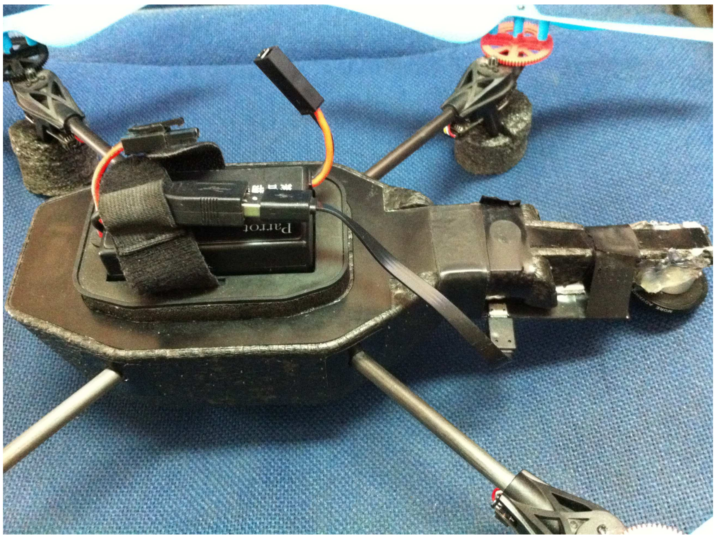

Figure 6. APM is connected to flight control board

Propeller protector

The official propeller protector is made of carbon fiber to minimize the weight, which is shown in Figure 7. It provides protection for important devices and protects people from being hurt by propellers. Two magnets are embedded in the protector in case the protector falls.

Figure 7. Propeller protector

SOFTWARE

Software Architecture

Key parts of software are in the ground station. They are state estimation, image processing and path planning. Figure 8 shows the software architecture of our system.

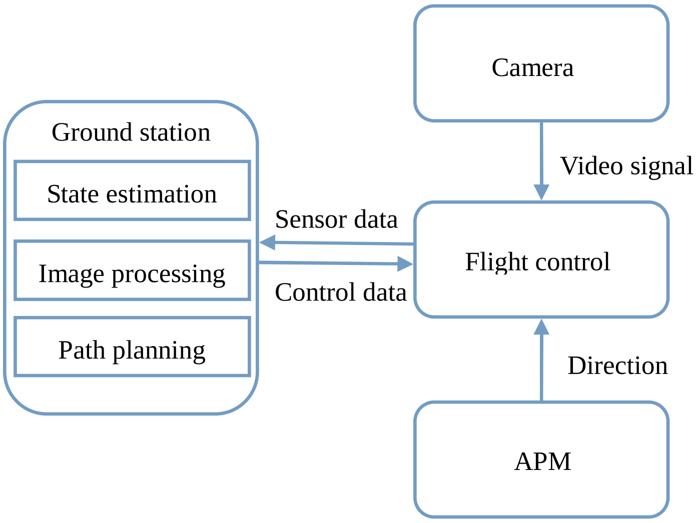

Figure 8. Software architecture

Simulation

Simulation of the competition environment is done by Gazebo. Gazebo is the physical simulation environment in ROS. It is a simulation software for robots based on ODE physical engine. It can simulate physical features of robots and its environment. Gazebo provides interfaces for connecting virtual robots and its control program. Also, maps in Gazebo can be defined by DAE files. DAE files can be modified in Google SketchUp and changes can be updated in Gazebo.

The simulated environment is shown in Figure 9. Several white points signify iRobots. The simulation environment can simulate the iRobot and aircraft moving and can provide the videos of the onboard camera.

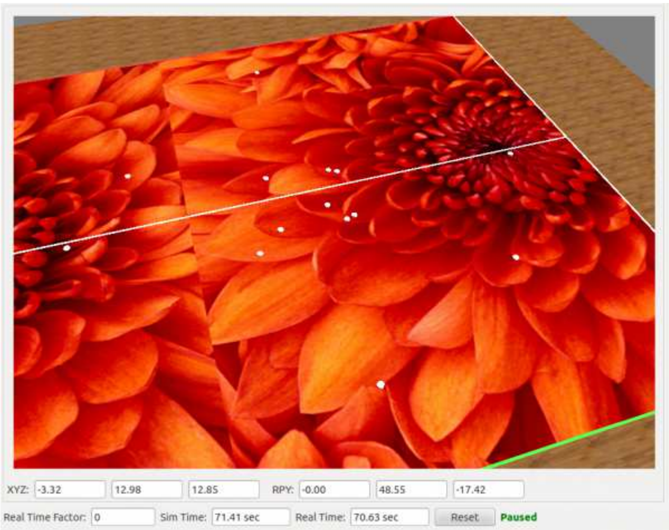

Figure 9. Simulation

State estimation

Data from inertial measurement unit are used to estimate the state of the aircraft. Figure 10 shows pose estimation model of the aircraft. The position estimated can reflect the actual position of the aircraft[1].

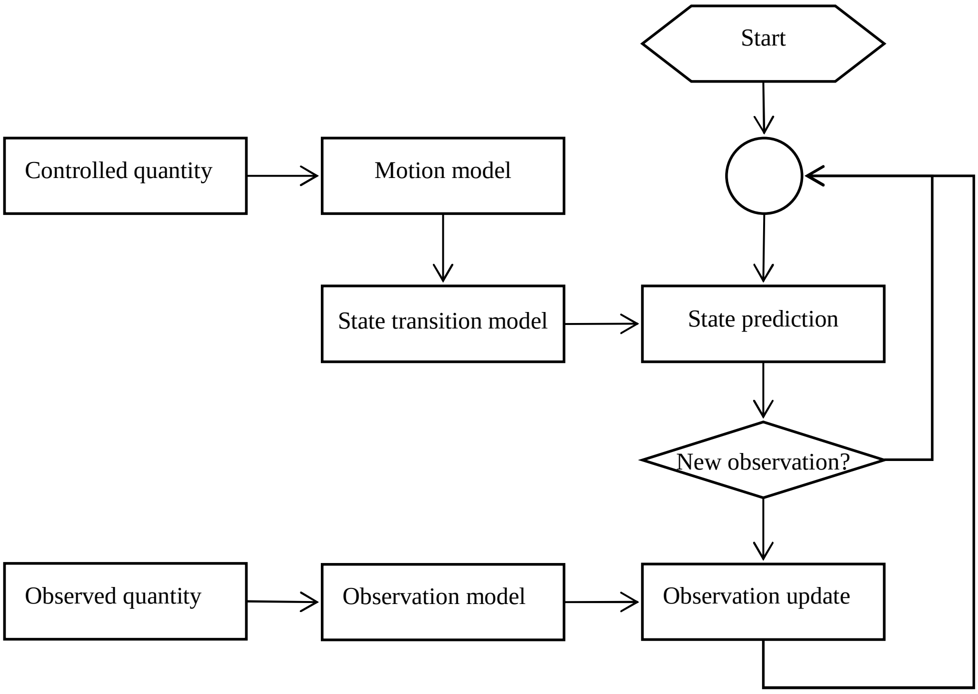

Figure 10. Unmanned aerial vehicle pose estimation model

In the figure, controlled quantities are converted to parameters such as linear velocity, linear acceleration, angular velocity and angular acceleration. The equation of transfer is obtained by the state transition model. State prediction is got from calculating the mean and covariance of data.

Similarly, observation quantities are obtained from parameterizing original data by observation model. State are updated by the observation updating progress. State prediction is trigger by control input, ignoring whether observation updates. The advantage is that the state is persistently updated when observation is not updated, which enhances the stability of the aircraft. However, if observation is not updated for a long time, estimation could drift due to control noise and inaccuracy of motion model.

Image processing

The image processing part mainly solves the identification for ground iRobot[2]. Based on the identification, the image processing module gives iRobots' coordination, ID and moving direction. In the process of recognition of iRobot, we use the HSV color space. Compared with RGB, HSV are affected less by light. According to the coordination and direction of the iRobot in the image, we will decide whether or not to chase the iRobot. iRobots are detected by the difference of the colors between the iRobots and the ground. Figure 11 shows two iRobots are detected by the aircraft. The outlines are drawn with lines in different colors.

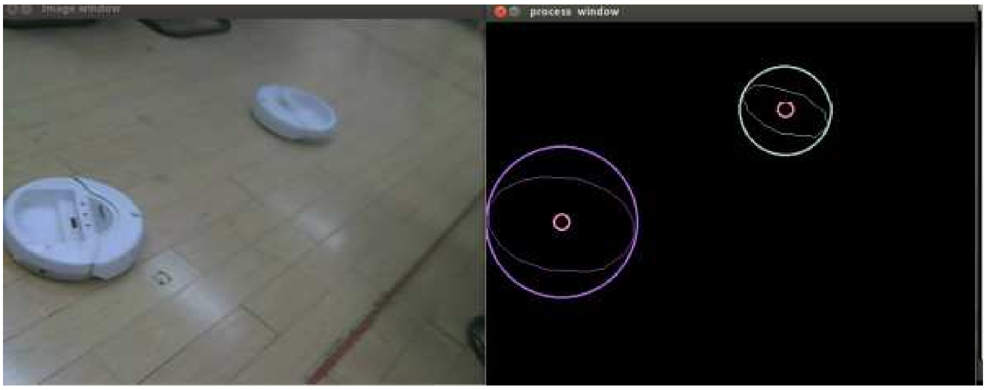

Figure 11. Image processing

Navigation

From simulation, we know the iRobots are not easy to go out within the time limit. Therefore, our navigation strategy is making the aircraft flying along the inner side of the green line (the double arrow) back and forth, which is shown in Figure 12.

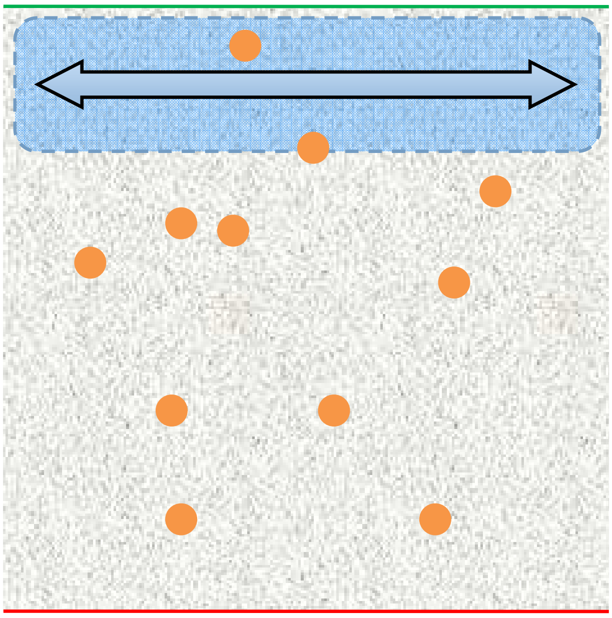

Figure 12. Navigation strategy

The light blue area represents the range that our quadrotor can cover by its camera. When the quadrotor noticed an iRobot running back towards the center of the field, it lands in front of the iRobot to make a collision. In this way, the iRobot turns its direction by 180 degrees. As a result, the iRobot gets out of the field from the green line.

CONCLUSION

We have designed and constructed an unmanned aircraft system that meets the requirements of the seventh mission of the IARC competition. Kill switch and propeller protectors and installed to ensure safety. A fisheye is placed in front of the camera to expand the range of the camera. State estimation and image processing algorithms are verified to work effectively. Navigation strategy is determined on the basis of simulation. Control performance has been verified by physical tests.

REFERENCES

[1] Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full

3d motion estimation[C]. Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on. IEEE, 2011: 155-160.
[2] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]. Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 2007: 225-234.