Northeastern University
Autonomous Aerial Robotics Team

Yuchao Hu
Northeastern University, Shenyang, China
Jianyu Yang
Northeastern University, Shenyang, China
Bo Shang
Northeastern University, Shenyang, China
Siyang, Liu
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 by assembling hardware and software. The aircraft carries an APM autopilot, an onboard computer, a camera, a lidar and a device to replace the USB flash drive. This quadcopter can send video, 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 aircraft.

INTRODUCTION

Our design is divided into two parts to achieve the desired flight navigation and replacement of USB flash drive: the hardware construction and software development.

In terms of hardware, first of all, we set up a commercial flight control system to remote control the quadcopter. On that basis, We replace the commercial flight control with APM autopilot. Gradually, the whole aircraft can be controlled to fly remotely. Later, we implemented SLAM by adding lidar and the AscTec Atomboard. Next, we realized hovering and wappoint navigation by combining the whole system. Additionally, we used OpenCV to recognize objects. Last, we designed a device to replace the USB flash drive.

On software, we combined and adapted ArduCopter and ROS. We used MAVLink protocol to complete the communication between them. Our tasks mainly solved the following inherent problems of ArduCopter: 1. Dependence on GPS; 2. Dependence on the earth's magnetic field; 3. Accuracy of altitude measurement is low; 4. Accuracy of horizontal control is low. Last, we used the ROS platform to add task planning for the system.

HARDWARE

Our autonomous indoor quadcopter hardware consists of power system, the sensors, the computers, the picking and placing mechanism of USB flash drive and the chassis.

Power System

As Fig.1 shown, the quadcopter is powered by a 12V Li-Po battery. To make sure it is safety at any times, there is a kill switch which can cut off the battery in case of emergency. The kill switch also divides the power into four motor drivers. Those drivers change DC into AC to run the motors and provide 5V for flight control at the same time.

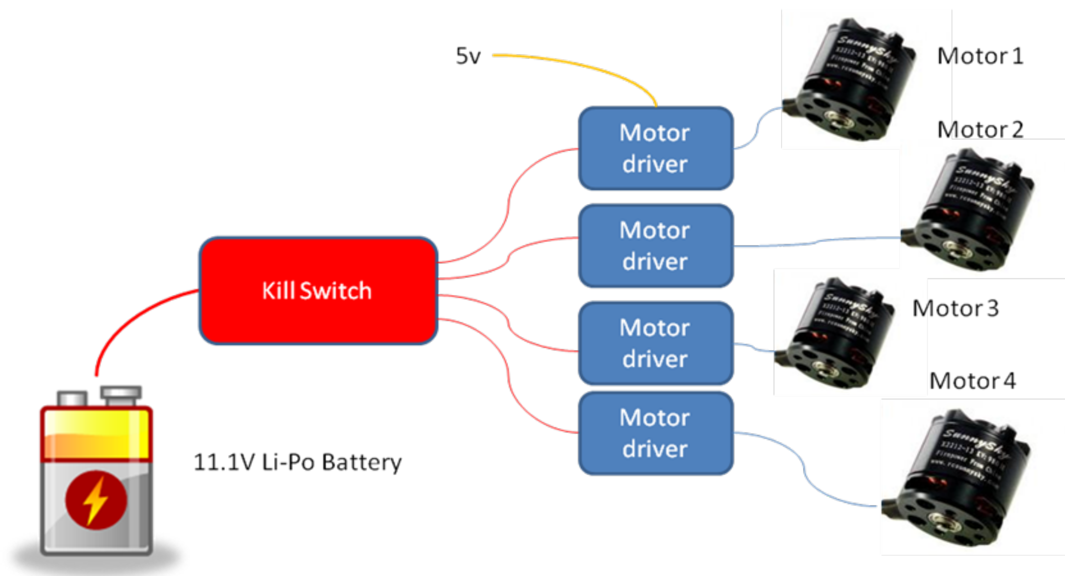

Figure 1. Power system

Sensors

There are three kinds of sensor onboard.

UTM-30 Laser Scanner

As shown in figure 2, it was used for detecting the environments around. Also provide data for SLAM (Simultaneous Localization and Mapping).

Figure 2. UTM-30 Laser scanner

MPU6050 Attitude Sensor

It provides attitude data for stabilization.

CMOS Camera
The camera is to detect targets.

Computers
There are two computers on board: an X86 and an ArduPilot Mega (APM) autopilot.

The X86-based computers is responsible for the communication with the ground station as well as positioning and path planning. To minimize weight, we chose the AscTec atom board as onboard computer, which weighs about 90g.

ArduPilot Mega is an autopilot controller, which enables autonomous flight.

Picking and Placing Mechanism of Flash Drive
Our picking and placing mechanism of flash drive is very simple. There is no server on it. When the aircraft flies over the USB flash drive, the ingenious mechanism would automatically pick up the USB flash drive and place a fake one.

Chassis
The chassis is made of carbon fiber and aluminum to minimize the weight. It provides protection for important devices.

SOFTWARE

Software Architecture
Our quadcopter aircraft software system uses APM combined with ROS. Both of them are open source software. Therefore, we modified codes on the basis of the original ones to make the system suitable for indoor environment.

ArduPilot Mega
APM is responsible for the PWM signals of the four servos of our unmanned aerial vehicle (UAV) immediately as the bottom part of the software architecture.The bottom control algorithms which need to run at a high frequency run in the APM, such as stability control of unmanned aerial vehicle (UAV), altitude control and throttle control.

Altitude Control
Altitude data come from two sensors: Sonar and lidar.

The cause of abandoning using the pressure sensor is the airflow the motor of unmanned aerial vehicle (UAV) generated makes the error of the air pressure sensor goes between -0.5m to +0.5m.

Data Processing of Lidar
As shown in figure 3, if we set the measured altitude is h', the true altitude is:

$$h = OC * \sin\alpha$$

From the figure:

$$OC = OB + BC = h' + d * \cot\alpha$$

Therefore, true altitude can be calculated by the following formula:

$$h = (h' + d * \cot\alpha) * \sin\alpha$$

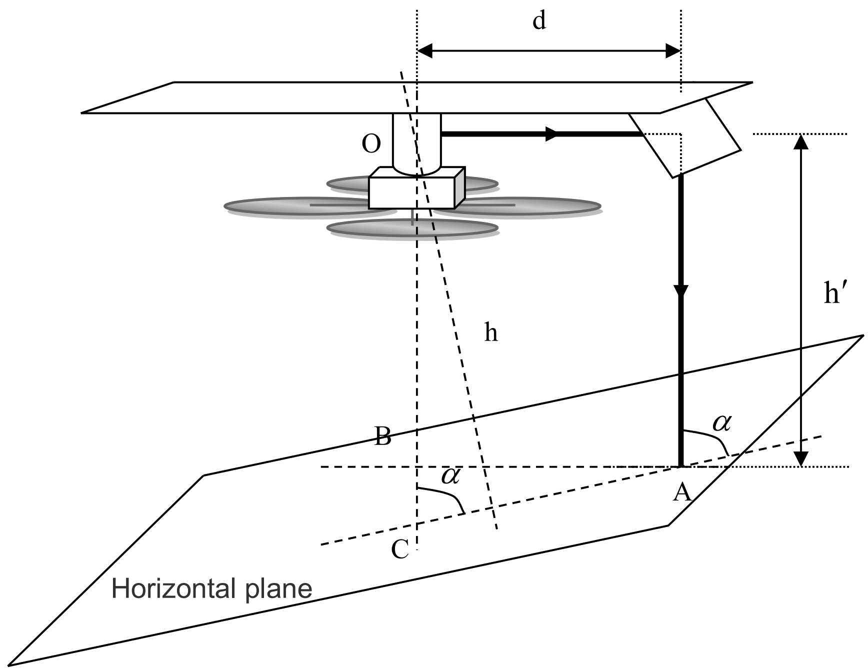

Figure 3. Calculation of altitude when the aircraft is not straight

Sonar

Since there are many obstacles in room, the surface of the ground is uneven. Using a single sensor is easy to cause the misjudgment of unmanned aerial vehicle (UAV) when UAV is flying over obstacles such as tables, chairs and windows. Therefore, lidar and sonar are fixed at different positions of the UAV. When UAV is flying over obstacles, there's difference between the output of the two sensors. However, we can determine the height of the obstacles from the differences and revise the estimated altitude of the UAV.

NAVIGATION

ROS is a robot operating system platform, on which high-level control algorithms are executed, such as SLAM, path planning, virtual GPS, image processing and communicating with APM by using MAVLink protocol.

SLAM algorithm of this platform uses a method based on point cloud registration. Because the scanning frequency of Hokuyo UTM 30LX is very high (40Hz), this algorithm can calculate the optimal position and orientation of MAV based on each scan. Since the time between two scans is short (1/40Hz), in the case of normal flight, it can fully meet the simultaneous localization and mapping requirements.

Path planning algorithm builds a map by using costmap to ensure that the resulting path can fully consider the size of the MAV, so MAV can avoid obstacles. Since the paths generated by this algorithm tend to explore the unknown area of map, we need to control the scope of the generated paths. This algorithm is improved by building the virtual fence to ensure that the resulting path will not leave desired area.

Virtual GPS is to solve the GPS positioning failure in the indoor environment. Its working principle is that the SLAM algorithm employs lidar data to generate MAV location information, which is converted to UBX format to send to real GPS interface of APM board. Thus the function of real GPS is fully replaced by virtual GPS.

Virtual geomagnetic sensor is used because geomagnetic sensor is instable in the indoor environment, as a result, MAV cannot determine the orientation of itself only based on the geomagnetic sensor. High-level algorithm is needed to calculate the orientation of MAV, which is sent to APM. The detail process is similar to Virtual GPS. SLAM algorithm generates the UAV orientation information, which is sent to APM through MAVLink protocol.

Image processing is used when the UAV is performing a specific task. For example, searching for specific objects according to the known image information in the scene.

Figure 4 shows the core part of the software.

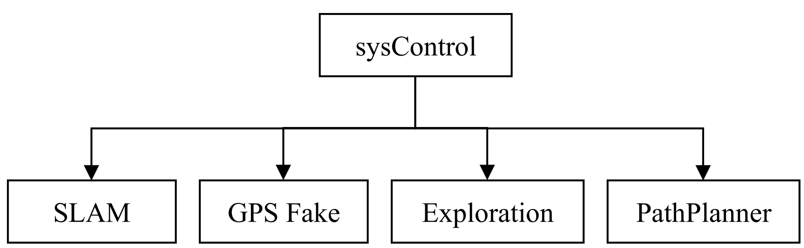

Figure 4. Top diagram of software design

SLAM module is responsible for receiving data from the lidar and creating two-dimensional map. This module can also estimate vehicle position and orientation relative to the map. Virtual GPS module converts location information from SLAM module to UBX binary format, which is sent directly to the APM board. It can deal with the problem that valid GPS signals cannot be received. Virtual compass extracts orientation information from the output pose generated by SLAM to construct a two-dimensional vector, and then send it to APM via MAVLink. It can solve the problem that MAV may not get the correct geomagnetic information in the indoor environment. Path Planner module determines the path based on the SLAM map and specific searching needs.