The ARLISS Project
Winter Progress Report
CS 462

Zhaolong Wu
Capstone Group 27, Winter 2017

Abstract

This document briefly describes how Zhaolong's pieces of the ARLISS Project is developed and implemented throughout the Winter Term 2017. Including brief intro of the ARLISS project, protocols that Zhaolong is responsible for, and a brief group evaluation.

I. Introduction

ARLISS is a rocket launch for international student satellites. The primary research outcome of this competition is the feasibility of autonomous driving enabled satellites that can be fit into a standard soda can and weigh less than 350 grams.The competition protocol is each team design a satellite that is going to be deployed by the ARLISS rockets at 12000 feet AGL. The unmanned satellite has to capable of being landed safely and travels to a given target pole autonomously.

In the ARLISS OSU chapter, there are mechanical and electrical engineering, computer science students teamed up to make this project happen.

II. Individual Pieces

The computer science team has decided to divide this CanSat project into 12 pieces, which giving each member 3 pieces of technology to develop. I am primarily responsible for designing the navigation system algorithm, payload fairing and getting the rover unstuck when it lands on its side protocols.

The navigation system protocol loops through of itself to update the shortest path on a given time interval. The payload fairing protocol is done by wrapping the rover into a piece of plastic, and having the parachute strings attached onto it, when the rover has landed, the servo arm attached on the rover will cut the strings that frees up the rover. Getting the rover straight up when on its side protocol also manipulates the servo arm to push against the ground to straighten up the rover. During the second half of the winter term I also worked on finding the finishing pole protocol, specifically I wrote an angle detection program in C++ that uses OPENCV library. I also wrote a little program that will determine the center of any given shape, it will also be useful on locating the target pole.

III. Progress and problems

At the beginning of the winter term, I was mainly focusing on getting the navigation algorithm and program ready, which by the week 5, the navigation system algorithm is already finished, however, due to the mechanical team failed on the rover's stress test in week 5 and electrical team did not meet the deadline of getting the rover's electrical components ready by week 10, I decided to put the navigation program development on hold at the end of the week 7. This is mainly because of I want to have a functioning rover ready so I can understand how the rover behaves when it drives, will the rover drifting left or right when it moves forward? If so we need to make a program that compensates the coordinates constantly. How fast the rover can go? What's the rover's maximum turning speed and angle? How accurate the GPS is? These are all unknowns and I have strong feeling and big concerns on that the current navigation will have to be rewritten or optimized when we can perform tests on the physical rover. So at the beginning of week 8 I stopped working on the navigation function, which gave me more time to work on other protocols. The current navigation output and the result comparison with the online GIS source is shown below in **Fig. 1** and **Fig. 2**

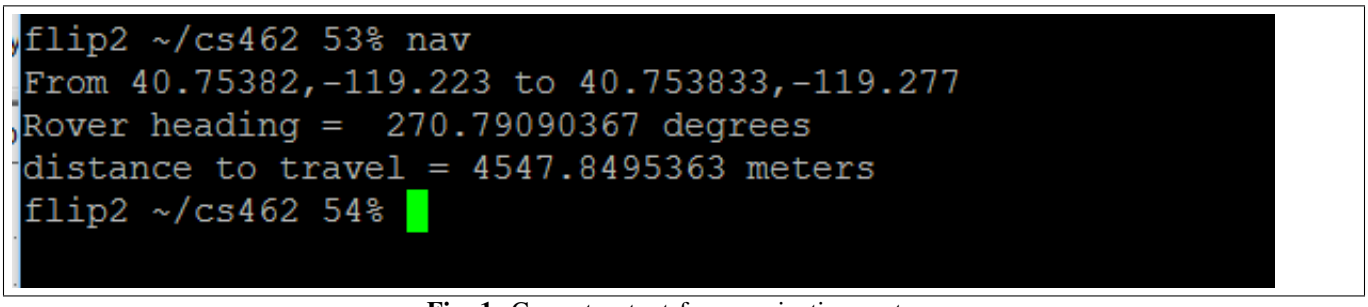

Fig. 1: Current output from navigation system.

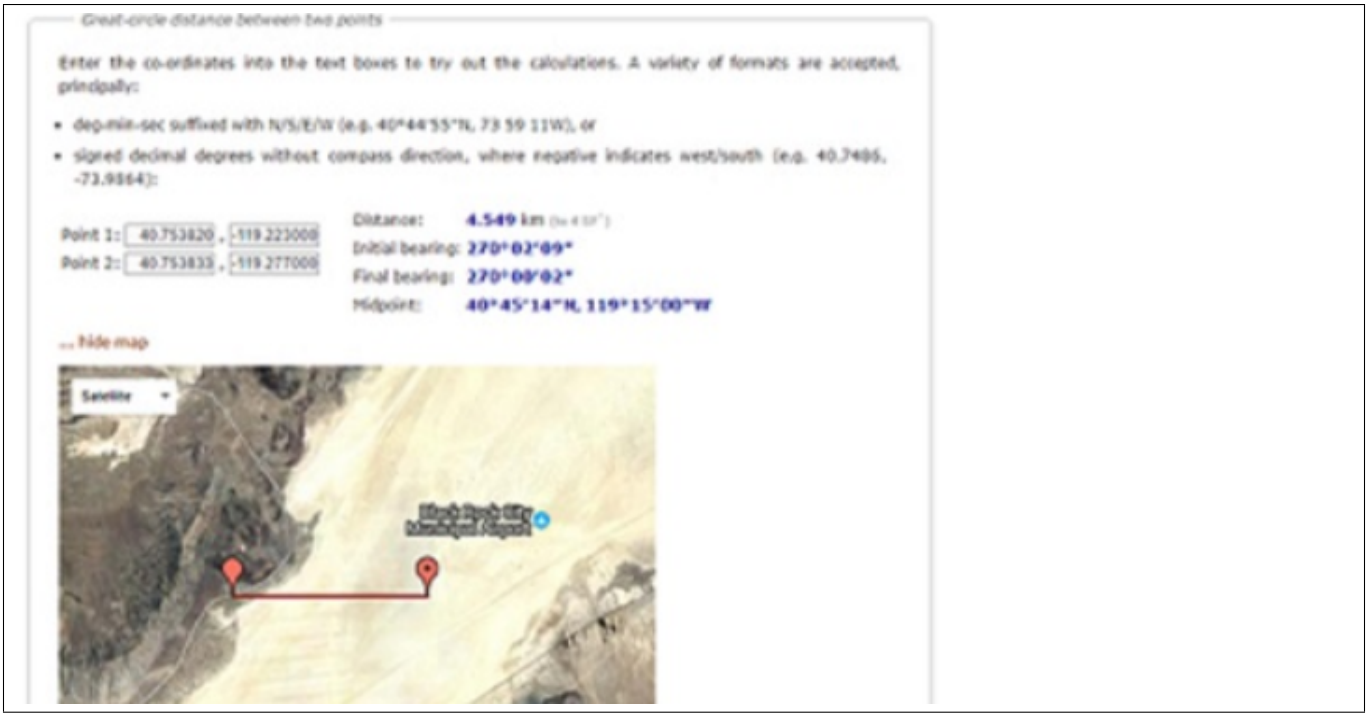

Fig. 2: Comparing current output from navigation system with online GIS source.

For the payload fairing protocol is currently finished and being tested for multiple times. The fairing itself is a piece or plastic that wraps around the rover. In the first test, we put the wrapped rover a stress tester to simulate the satellite launch stage, the rover stayed in one piece throughout the test. The second test is we drop the rover with the parachute attached on from certain height, to simulate the rover's landing stage, the rover hit the ground softly and stayed in one piece as well, we performed this test multiple times and conclude that this fairing protocol works. See **Fig. 3**

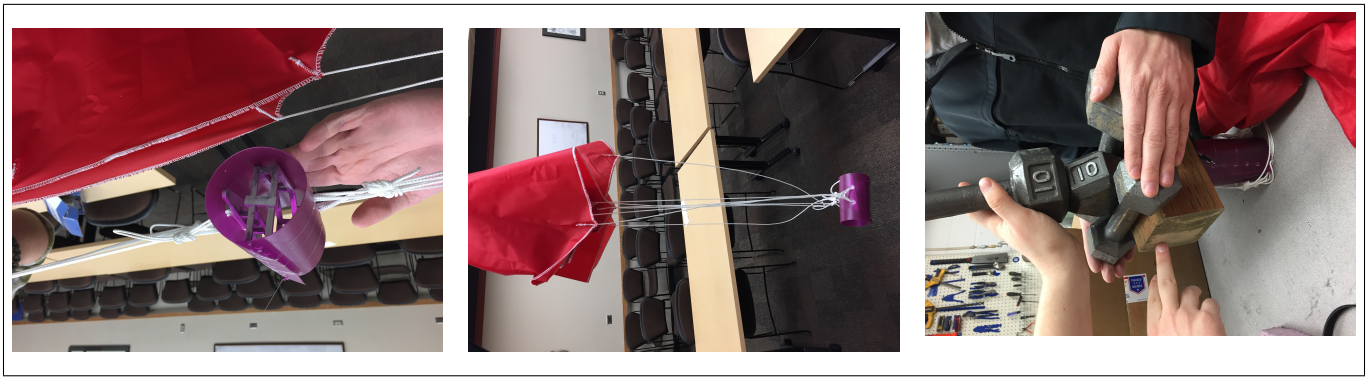

Fig. 3: Payload Fairing Protocol tests

The other piece I was working on during the second half of the winter term was the getting the rover unstuck when it on its side, this protocol is fairly easy to implement, and mostly the code is sharing with another group member Paul's getting the rover unstuck from obstacle. The only difference is that when the rover lands on its side, it will not moving and the accelerometer will capture a sudden change of speed and direction, which triggers the sub-module to be enabled. This is currently finished but unable to demo due to unknown of servo behavior and functions.

The last pieces I worked on in the term were the camera angle detection and center of a shape detection programs. They were both implemented by using C++ and OPENCV library. The angle detection program by preset a HSV color range to get only red color, which is as same as the cole of the finish pole. The program will detect the red object and calculates the corresponding angle of the object with the center of the camera. Further tests has to be performed on this protocol to do the angle detection, we don't know how many pixels the camera will have, the algorithm is to compare the X value with the center of the camera to calculate how many pixels there were to determine the angle, See **Fig. 4** but we don't know that yet, this will continue to be implemented next term.

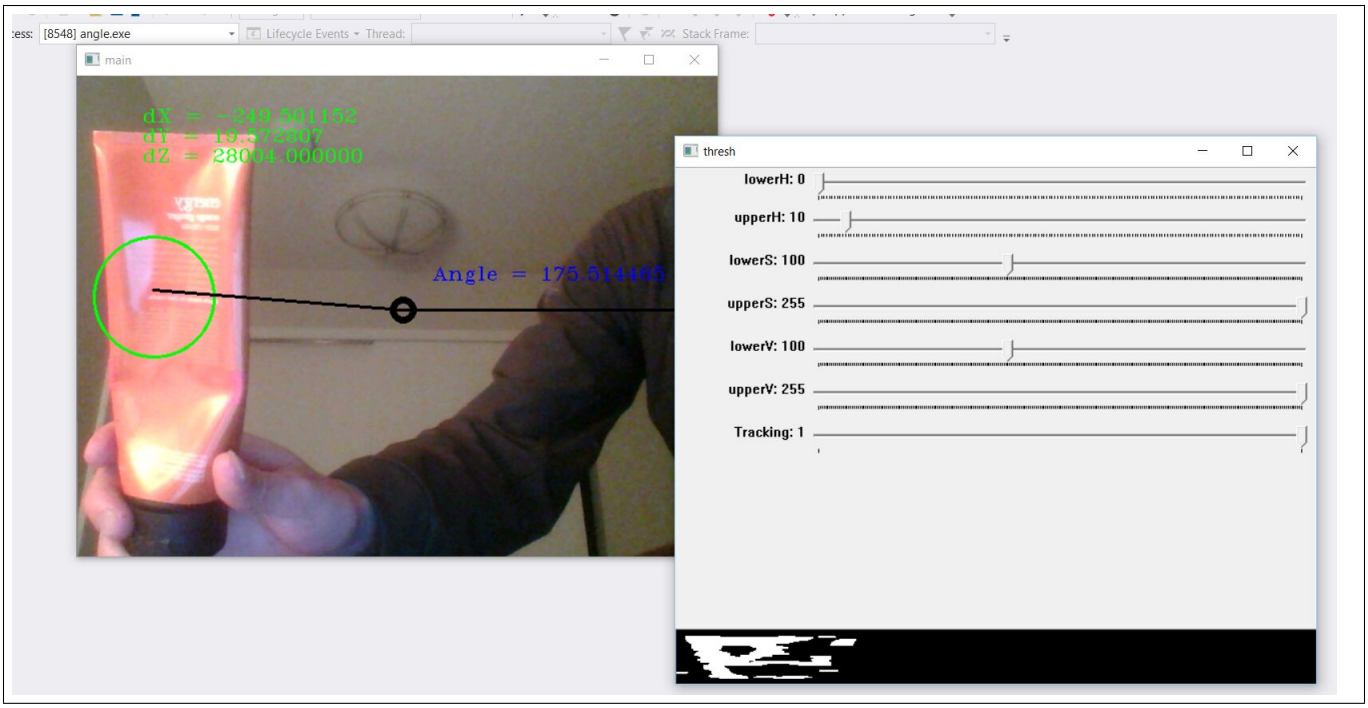

Fig. 4: Camera capturing the red/orange object and returns an angle

IV. Evaluation

In the ARLISS project cs chapter, I'm more focusing on the navigation of the rover, but I also worked on other parts like listed above. Paul is responsible for getting unstuck from obstacles, finding the finishing pole and parachute deployment protocols, he also largely contribute on putting every piece of programs together into a single module and put load into the Raspberry PI zero, because his talent on the Linux system. Steven is mainly focusing on the obstacle avoidance protocols and right now the obstacle avoidance function seems fully functional and ready to be tested, he also worked on putting everyone's presentation together, which should be respected. Zach's main focus is the Vision system of this rover, which where the autonomous driving function is highly dependent at, he works really hard to make this happen by researching the OPENCV library and image processing works, his module is finished as well and ready to be tested.

I would like to express my appreciation to everyone in the group worked hard and shared equal amount of work together. Because we have different class times we sometime won't able to meet together very often, and we only communicate through e-mail which give us some hard time to get prompt answer when we needed from each other, but overall everyone makes quick response on emails. Next term we will set up a CS team weekly session to give us time to implement things together more.

Though I have to address here is the most difficulty our team had during the winter term is we have no access to the rover, this is because of both mechanical and electrical team had delay on their progress, which leaves us no time to finish our functional rover project by this term, which was our goal from the beginning of the term. Because we have no rover to test, a lot of testing parameters we used were based on assumptions, which I'm sure that lots of them are not valid in the real world scenario, this may cause us having a real rough and busy term in the spring term.

V. A Reflection of Development Period

Retrospective		
Positives	Deltas	Actions
Everyone in the group shares equal amount of work. The whole program is now combined in one piece. The physical rover is nearly done, which gives us whole next term to perform tests.	Lack of communication with ECE team. No physical rover to perform test with, testing by making assumptions. Establish a weekly CS group meeting to work on the project together instead of everyone does on its own.	Make the ECE team to do better on meet the deadline. Establish communication with ECE team on regular basis. Finish the coding part, break the rover apart, make the rover better.

VI. Conclusion

Overall, I had some mixed feeling of this term's capstone in terms of progress, our CS team members all worked hard, but we didn't had to much to do, we had to assume a lot of things because of our project is highly dependent on other groups completion of their work. So in the future we should probably thing more on the project management for this type of projects. The good thing is most of our program are finished and they are ready to get tested. I have fairs on next term we will have loads of work to do to make this project happen, but after all, I'm very pumped for this and have very positive feeling on the future of our project.