ILR 07 – Progress Review 8

Shivam Gautam

TEAM DAEDALUS

Members: Pranav Maheshwari, Richa Varma, Mohak Bhardwaj, and Dorothy Kirlew

February 11, 2016

1. Individual Progress

I undertook the following tasks:

1. Comparing Laser Range Finder (LRF) values with the ground truth.
2. Implementing the LRF as a ROS node.
3. Creating CAD mounts for the LRF Module.
4. Populating the PCB board with components.

1. Comparing LRF values with the ground truth.

This task is part of our test plan and is key in determining how accurate the LRF actually is. Table 1 lists the readings obtained from the LRF versus the ground truth

LRF Measurement (cm)	Ground Truth (cm)	Error (%)
15.4	15	2.67
20.6	20	3
25.2	25	0.8
30.2	30	0.68
40.3	40	0.75
50.8	50	1.6
60.8	60	1.33
71.1	70	1.571

Table 1- Observations from LRF range testing

This clearly shows that the error of the range measurements is within the error mentioned in the datasheet of the LRF module (+/- 3%). The error also lay within +/-2 cm- the benchmark that was set in the spring test plan. Since the test was successful, I decided to integrate it with the platform. The test bed for the measurements is depicted in Fig 1.

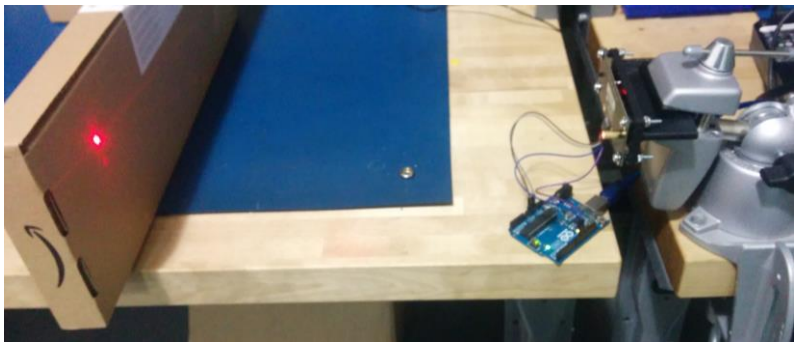

Figure 1- Test Bench Setup for measuring range values of the LRF

2. Implementing Laser Range Finder as a ROS Node

The LRF module consists of a laser, a CMOS camera and an onboard processor. The module works by emitting a beam of laser which is detected by the camera. The exact depth of the point is estimated by triangulation. If we sweep the laser incrementally between two angles, we would be detecting obstacles that would be half a meter in front of our platform.

This task involved writing code for the LRF based on the following outline-

1. Set the servo motor at a fixed angle.
2. Send a command via a software serial port to the LRF requesting range measurement.
3. Receive a string from the module and parse it to extract range in cm.
4. Check whether the current reading for range is an emergency or not.
5. Initialize a ROS node to publish the emergency message.
6. Rotate the servo by thirty degrees and get another reading. Repeat the process to get three readings at three orientations of the servo- 0 degrees (center), +30 degrees (right) and -30 degrees (left).

After writing the code, I conducted testing of the actual hardware and observed the flagging of emergencies when the obstacle was in close proximity. The ROS topic that would interface with the navigation node was observed and worked satisfactorily.

3. Creating CAD Mounts for LRF Module

Figure 2 depicts the calculations related to the rotation angle of the LRF. This shows that the servo needs to sweep an area of +/- 35 degrees from the center of the platform. To do this, mounts were created to attach the LRF to the servo.

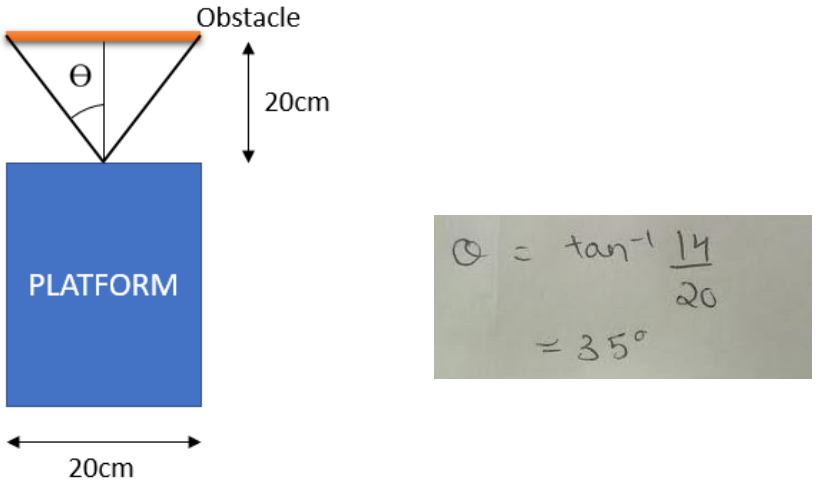

Figure 2- Calculating sweep angle for servo

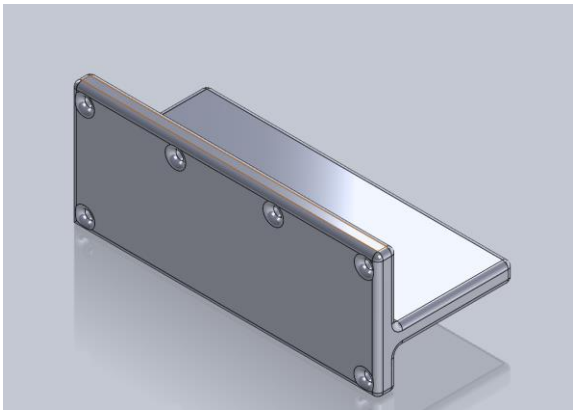

Figure 3- Isometric view of CAD mount (Cited by Pranav Maheshwari)

The mounts for integrating the LRF with the servo were sketched in CAD and then subsequently 3D printed. The designs are depicted in figures 3 and 4.

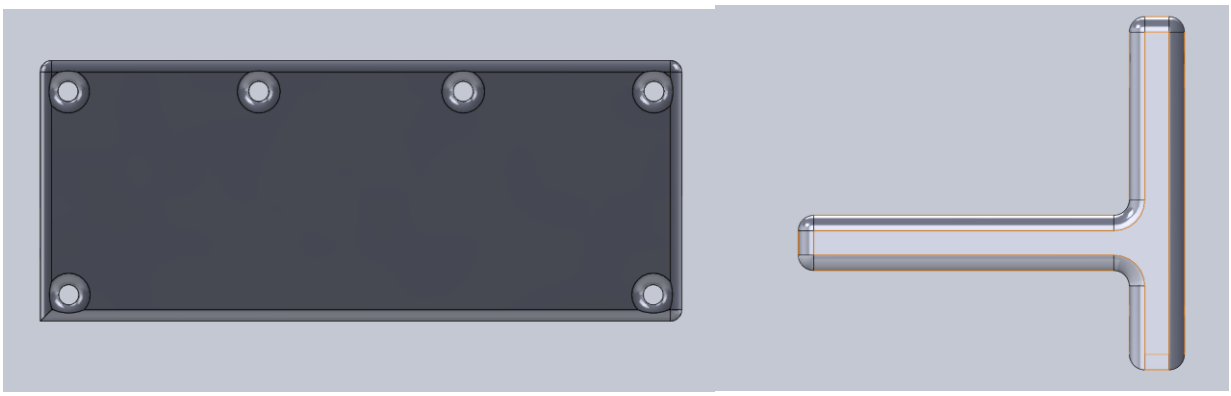

Figure 4- Front and Side view of CAD mount

The final printed mount and the integrated sensor is depicted in figure 5.

Figure 5- 3D printed mount with sensor

4. Populating the PCB Board

The manufactured PCB board was delivered this week along with a few components. The board has been populated partially and would be fully populated once the remaining components arrive next week.

2. Challenges

One of the key challenges I faced was re-calibrating the LRF module due to varying lighting. The sensor takes images of the emitted laser beam and the measurement was observed to be affected by different lighting conditions- indoors and outdoors. To mitigate this, I wrote a test script to calibrate the camera at different distances and obtain accurate range readings. Also, actual testing with the servo has revealed that rotating the sensor has an impact on the range readings. I would be investigating this further and would look to avoid this problem.

Another challenge arose when the power distribution board malfunctioned during testing of the locomotion subsystem. I diagnosed the problem as a short circuit which fried the voltage regulator. To avoid this in the near future, I tried to educate everyone about the risks of not doing the connections properly while working with the platform. The backup power distribution board was ordered and should arrive shortly.

3. Teamwork

I worked with Pranav to create CAD models of the mounts for the servo and the LRF module. I also collaborated with him to ensure that the emergency node published the right messages for the navigation node to pick up. Pranav also worked on the visualization tool and also tried running the navigation API of Oculus Prime. Mohak worked on completing the multi-agent planner and we discussed a few advanced scenarios that we could consider for the planner. Richa and Dorothy worked on testing the communication system with single-board computers to achieve communication between three platforms.

4. Plans

The team planned to finish all the major sub-systems by this progress review, but faced hardware related problems which did not allow the team to complete the mapping of the parking lot. We would be testing everything on a new platform which would house the exact hardware as is required for the Oculus Prime server to run on. Dorothy and Richa would be looking to integrate their communication routine with the commands received from the phone. Pranav would facilitate this by coordinating with them while writing the master node. Dorothy would also be constructing the parking lot once she receives inputs from the Mohak regarding the material needed for mapping. Mohak would be completing mapping using the Oculus Prime API. I would be working on integrating my sensors with the platform and adding the PCB and additional components to the platform.