

# Individual Lab Report #09

Progress Review #10

October 29th, 2020

Alex Withers

Pit Navigator

Team G: The Pit Crew: Awadhut Thube, Justin Morris, Alex Withers

<b>Individual Progress</b>	<b>3</b>
Pit Navigator	3
Waypoint Navigation with Rover Hardware	3
<b>Challenges</b>	<b>4</b>
Pit Navigator	4
Waypoint Navigation with Rover Hardware	4
<b>Teamwork</b>	<b>5</b>
Pit Navigator	5
<b>Plans</b>	<b>6</b>
Pit Navigator	6
Testing at Gascola	6

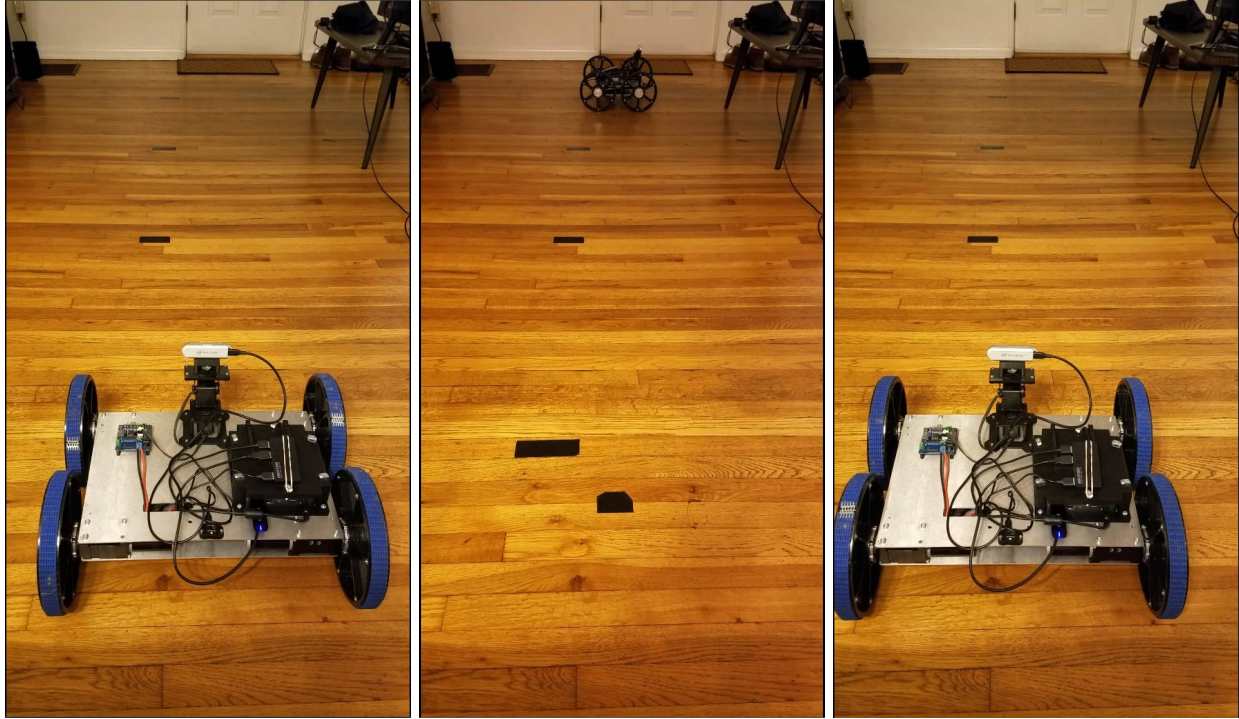
# Individual Progress

## Pit Navigator

### Waypoint Navigation with Rover Hardware

This week I took on the challenge of putting our waypoint navigation software onto our rover hardware to get the rover to navigate through a 3-meter square and end up where it started. This was a significant challenge and I encountered many problems along the way. In general, I was able to port the navigation code over from the simulation in less than a day, then worked on building the localization code that was inherent within the simulation, but not existent on the rover hardware. I first tried to run our navigation code with only wheel odometry, but the rover was too bouncy and did not reach its destination. This was evident in the video that I showed for PR 10. I began implementing robot localization. The usage looked very simple, but the online explanations were long and confusing. So I will summarize it here.

Largely the process to use the robot localization package is to make sure that your sensors are outputting the right format ROS message (Odometry or Sensor msg), a static transform to tell robot localization where your sensors are in relation to the center of the robot, and then use a true/false table to tell robot localization which values you want to use from each sensor. For example, on our skid steer robot, our wheel odometry message contained values for position in XYZ, orientation in RPY, velocities in XYZ, and angular velocity in RPY. We only considered the velocity in X as it was the only value that was truly accurate to our rover, so we gave that a true value and all the rest false values. Then for our IMU, it gave us values in all 15 states, XYZ, RPY, velocity XYZ, angular velocity RPY, and acceleration in XYZ. This IMU was really good, so it calculated all that for us with a compass, an accelerometer, and a gyroscope. But we found that the accelerometer would cause drift if it was on a slope, and the compass would force our orientation to not align with our map since we were ignoring the globe frame. By ignoring these values from the accelerometer and the compass, and only using the gyroscope we were able to produce the result shown in figure 1, an hour after our presentation for PR10. The rover travels 3 meters out to a position where it is rotated 90 degrees and then returns to the initial position. This test's scope was reduced due to rain and darkness.



<https://www.youtube.com/watch?v=paz57lqleX4>

Figure 1: Rover travels 3 meters out, turning 90 degrees, and then returns to the initial position.

## Challenges

### Pit Navigator

#### Waypoint Navigation with Rover Hardware

When working on the navigation for the rover, we came across a few issues with working with hardware, like batteries dying, and cables being cut, but also came across electrical and software issues. Firstly, when working on our rover, we let the battery's voltage run too low and were unable to charge the battery. Our replacement batteries had the wrong connector type to interface with our rover or the charging port, so we had to buy additional connectors on Amazon to get access to them very quickly. In all, we lost around 12 working hours to this fiasco. Additionally, when we were testing, the wire to the real sense got trapped between the wheel and the metal frame and became frayed to the point of not working. It was a USB 3.0 A to C cable that was surprisingly rare because it's on the higher performance end of cables. We tried using a USB2.1 cable of the same type but were unable to get it to run properly. After some extensive searching by the team in best-buys and online, I found one that I use to connect my Oculus Quest to my computer. We were able to use this cable to get the system running again. We lost another 12 working hours to this issue.

We also had issues that were unrelated to “hardware” in the strictest sense. The wheel encoders that we had do not have a datasheet online, and therefore we could not find the ticks per revolution value that is required to read the encoders properly. There was an initial estimation for the number of ticks, but it was overshooting our target distance, and we needed to do experimental tests ourselves to determine the best value. We settled on 6208 ticks per meter of travel. This includes our 10.5 cm wheel radius. Our code uses a ticks per meter value to inform the odometry how far it has gone, so we did not have to back-calculate a ticks per revolution. Lastly, the TEB local planner simply tries to be too smart. We would like the rover to make minor adjustments to follow a line, but make point turns whenever it is time to turn. TEB is always trying to optimize for car-like robots rather than skid steer, so it will deviate from the global plan to make a more optimal turn or make too many K turns. It has been an issue since we got the local planner and has become more obvious as we move over to the hardware. Additionally, TEB will send slower velocities when it is making a small movement. On our rover, the wheels won’t turn unless they surpass a specific velocity commanded by the twist message. So when we got to a place where TEB wanted to make a small movement, the wheels would freeze up and the rover would not move at all. We analyzed the data from the twist messages that TEB was sending and determined that, since there was no way for TEB to set a hard minimum velocity, we would add it to our motor controller. This leaves us open for localization issues in the future but has worked for us so far since we can sample our sensors fast enough.

## Teamwork

### Pit Navigator

Individual	Main	Sub	Description
Alex W.	Waypoint navigation with rover hardware	Set dates for Gascola testing, assist ideas for brinks in brinkmanship	I was responsible for getting the waypoint navigation code to run on the rover hardware. I also set up dates and times for testing at Gascola.
Awadhut T.	Detect sheer brinks in meshing code	Assist running meshing algorithms on rover hardware	Awadhut played a major role in detecting sheer brinks in rover code and developing the visualization to see it.
Justin M.	Run meshing algorithms on Rover hardware	Assist waypoint navigation with rover hardware	Justin worked on running our meshing algorithm on our rover hardware with the real sense stereo images.

# Plans

## Pit Navigator

### Testing at Gascola

In preparation for the dry run of our FVD, we would like to do system integration tests in the field. The only place we can really test the entire system is at Gascola. I have been the point of contact for making sure that the testing site is available on the dates that we need, and am responsible for making the test plan and making sure the tests run smoothly. In the coming time before the testing in the field, I will be making as many preparations as possible so that it can run smoothly. I will ensure that we have access, have the necessary equipment, extra batteries, vegetation clearing tools, lunch, PPE, rope, robot, portable monitor, hotspot, remote access to the rover, extra cables, chairs, and water. I will also make sure that we know which tests are useful and which we want to run so that we can most effectively prepare for the FVD.