

Individual Lab Report #07

Progress Review #8

October 1st, 2020

Alex Withers

Pit Navigator

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

Individual Progress	3
Pit Navigator	3
Pit Edge Detection in Simulation	3
New Simulation Environment	3
Inputs and Outputs for RASM	4
Challenges	4
Pit Navigator	4
Inputs and Outputs for RASM	4
Teamwork	5
Pit Navigator	5
Plans	5
Pit Navigator	5
Inputs and Outputs for RASM	5

Individual Progress

Pit Navigator

Pit Edge Detection in Simulation

We had existing code for detecting the pit edge from last semester. The goal for this PR was to port it into the simulation and add it to the planning and simulation pipeline. I was working from the state machine side while Awadhut was working from the detection code side. I set up the place for the code to begin running within the state machine, and made sure all the hooks were in place. Once the hooks on each side aligned, I ran the code on my computer, and it worked the second time. Not only did it work the second time, the rover did not fall into the pit the nine times it approached the pit. On the first try, an info publisher had the wrong name and crashed the state machine node causing it to fall into the pit after sending the first twist message. The point cloud started out being visualized as white, which made it hard to see. It was changed to red for the video. The video of the simulation is available here:

<https://www.youtube.com/watch?v=3QJY7qDFKY0>

New Simulation Environment

Awadhut had a significant amount of trouble running the simulation on his laptop and on the nuc. While I was able to tell him about some of the settings that he could turn down, like turning off shadows or using a different global planner, or turning visualization off completely, he could only achieve a real time speed multiplier of .35x. This means that one second of simulation time took 2.85 seconds to calculate. It makes testing extremely difficult and time consuming. I am able to achieve a real time speed multiplier of 1.5x which means that I can run the simulation in real time or slightly faster. I have an additional graphics card that I am able to use for the simulation which allows me to achieve these higher speeds. But always using my system for testing is not ideal as we both have our own schedules and would cause delays if he cannot test. I worked to make a simplified version of the simulation that should be significantly easier to run on a laptop for testing different edges. In Solidworks I drew up a model that is essentially a tower with 4 pits on each side. Each side of the tower has a different type of edge that the rover could encounter and the geometry is much simpler so that the computer can do less work to render it every second, only 1,000 triangles vs the 3 million on the West desert sinkhole model. Figure 1 shows the new simulation environment.

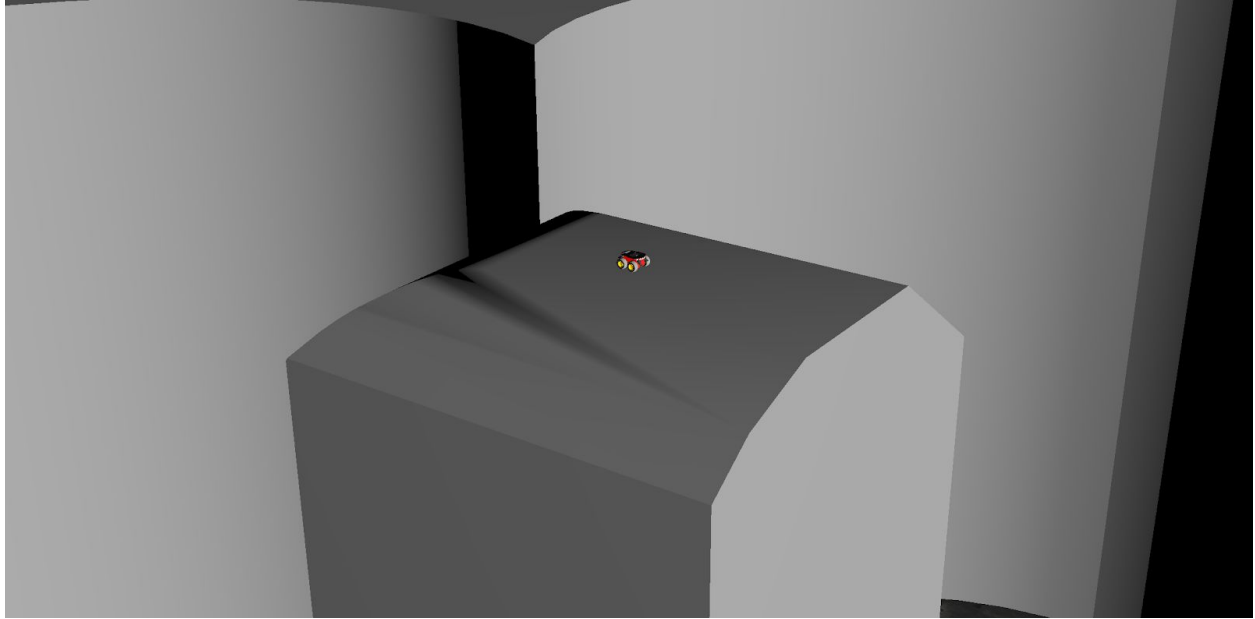


Figure 1. Different pit edges on each side of the center tower to showcase the simplified Pit environment for testing on laptops.

Inputs and Outputs for RASM

In my spare time I was working on the next PR deliverable, which is making the hooks to connect to the RASM code developed by MoonRanger. We were primarily interested in the mapping and localization part of their code. In particular, they seem to be making a persistent triangular mesh of the terrain that they have traveled on previously, and we would like to adopt that and use it in our brinksmanship code. By creating a triangle mesh of the terrain in front of us we can have a greater understanding of the slopes and what is pit and what is not. I have identified the code responsible for creating the map and triangulating the terrain, and can confirm that it takes in a point cloud that we are already sending. Other than that its difficult to read and does not have very good documentation, so I will need to get in touch with Pat, and possibly have weekly meetings with him.

Challenges

Pit Navigator

Inputs and Outputs for RASM

As I described above, the code is difficult to read and there is not a guide book for how to navigate it. There is also a significant amount of files and file structure that we may be interested in that needs to be combed through. If we had a guide for navigating the code, it would be very helpful, but Pat has made it clear that he does not want to handhold us through

what exists. We will need to strike a balance between talking with Pat and combing through the material itself.

Teamwork

Pit Navigator

Individual	Main	Sub	Description
Alex W.	Simulation Env, Assist Pit edge detection in sim	Begin looking at inputs and outputs for RASM	I was primarily assisting on getting the pit edge navigation working in the simulation then developing a new smaller environment for Awadhut to be able to do the simulation on his computer.
Awadhut T.	Pit edge detection in sim	Assist with getting the panorama sequence to work on the robot	Awadhut focused on setting up the pit edge detection code in the simulation but had issues running it on his machine
Justin M.	Tilt angle capturing	Get access to Blue2	Justin focused on adding additional parameterization to the pit capturing sequence and set up the panorama sequence to work on the robot

Plans

Pit Navigator

Inputs and Outputs for RASM

In preparation for the next PR, I need to create solid hooks for adding a finished RASM code to our code. I'll need to figure out what RASM requires to run and where it starts and is launched in their code. I definitely know that the mapper class is important and necessary, but the planner seems to be redundant to what we have, but with additional proprioceptive sensing and limits. I will need to determine if this planner works on top of the 3D mesh or if it is translated into a 2D map. If it is translated to a 2D map, then it would be very helpful to hook that into our code and only use the translation function. There is a significant amount of mystery as to what is useful and what is not and we need to be prepared for any amount of useful code.