

# Individual Lab Report #06

Progress Review #7

September 16th, 2020

Alex Withers

Pit Navigator

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

<b>Individual Progress</b>	<b>3</b>
PitRanger	3
Risk Assessment	3
Pit Navigator	4
Separation of Rover and Simulation Code	4
Stereo Vision	5
<b>Challenges</b>	<b>6</b>
PitRanger	6
Risk Assessment	6
<b>Teamwork</b>	<b>7</b>
Pit Navigator	7
<b>Plans</b>	<b>7</b>
Pit Navigator	7
New Simulation Environment	7
Separation of Rover and Simulation Code	7

# Individual Progress

## PitRanger

### Risk Assessment

During my internship over the summer, I was working with Red, our project sponsor, on PitRanger, the father of our project, Pit Navigator. Red had a different timeline and set of requirements for PitRanger, and I was brought on to use my simulation expertise to create a simulation of a rover going around a model of a terrestrial pit that they had captured in September. Then the rover should be shown to go around the pit in stages, autonomously deciding when to come back to drop off images. It was during this period that I was able to create the Risk Assessment algorithm that solved both their problem and satisfied our requirements for PR 10.

To begin accomplishing this risk assessment task, I needed to first determine the risk that I was assessing and managing. I decided to focus on two mission ending risks, running out of time due to lunar night, and falling into the pit. If the rover takes too long to complete its mission, then there will be incomplete results of the pit. If the rover tries to go too fast it might peer in the pit too many times before returning to the lander. The rover could fall in on the first trip to the pit and there might not be any results of the pit to show. The desirable solution is to return to the lander as much as possible without staying out too late.

To figure out what the right number of trips to the lander is, one will need to break down what the rover is doing into 3 categories of actions. The rover needs to travel between the pit and the lander (Pit-Lander action), the rover needs to advance to the next vantage point and take its pictures (Next-Vantage action), and the rover needs to travel around the pit at a safe distance to get to the vantage points from the pit area entrance (Pit-Vantage action). The Pit-Lander action is traveling along a long distance without interruption in two directions. I've assumed that the time it takes to complete this action is constant. If the vantage points are patterned every number of degrees, then the Next-Vantage action can also be assumed constant as the distance traveled and actions entering the pit will all take the same amount of time. Finally, the Pit-Vantage action is harder. The Pit-Vantage action counts the time it takes to reach any point around the pit from one point on the circumference of the pit. Assuming that the pit resembles a circle from above, the rover has to travel a distance of  $\pi * r$  to reach the furthest possible vantage point on the pit to safely travel around the pit. The rover could also travel 0 distance to reach the closest possible vantage point. This leads me to believe that the Pit-Vantage action time should be averaged to be the time to travel  $\pi * r / 2$  distance. Once an estimate for the time it takes to accomplish each of these actions is accomplished one can use the following formula to solve to find the right number of trips to the lander.

### Formula 1:

$$Trips = (Time\_Available - Next\_Vantage * Unvisited\_Vantage) / (Lander\_Pit * 2 + Pit\_Vantage * 2)$$

In Formula 1, Time\_Available is the time allotted to complete the mission, and Unvisited\_Vantage is the number of vantage points that the rover still needs to take pictures of the pit from. Any partial trips need to be rounded down so that the number of trips is a whole number. Next\_Vantage, Lander\_Pit, and Pit\_Vantage are the time estimates of how long each action takes. As the rover starts doing these actions it can update these estimates with historical data, to get more accurate predictions. During the update step, I recommend using a weighted average to trust the historical data greater than the estimate.

## Pit Navigator

### Separation of Rover and Simulation Code

In the first few weeks of the semester, I needed to transfer the simulation onto my teammate's machines. At this point, the GitHub holds 17GB of files that are not necessary and not useful to have on a future rover. I decided to solve both of these problems by finally separating the thinking algorithms and the simulator control software. By doing this I would also be able to easily interchange the simulation environments so that my teammates could run smaller versions on their laptops. Once I could switch between simulation environments, I would move onto switching over to the rover in the future. While this should be done at the start of the project, we adopted this code from the previous MRSD team and it contained no such functionality then. Last semester we also had a limited understanding of this code. I was always poking it and moving files to see when it would break or if the file didn't matter. Then after working heavily with the simulation over the summer I had gathered a great enough understanding of the file structure to determine what was minimally required to run the simulation.

In the following figure, I have the before and after folder structure that went into setting up the separation of the code. I have changed the file names to describe what was relevant in the folders.

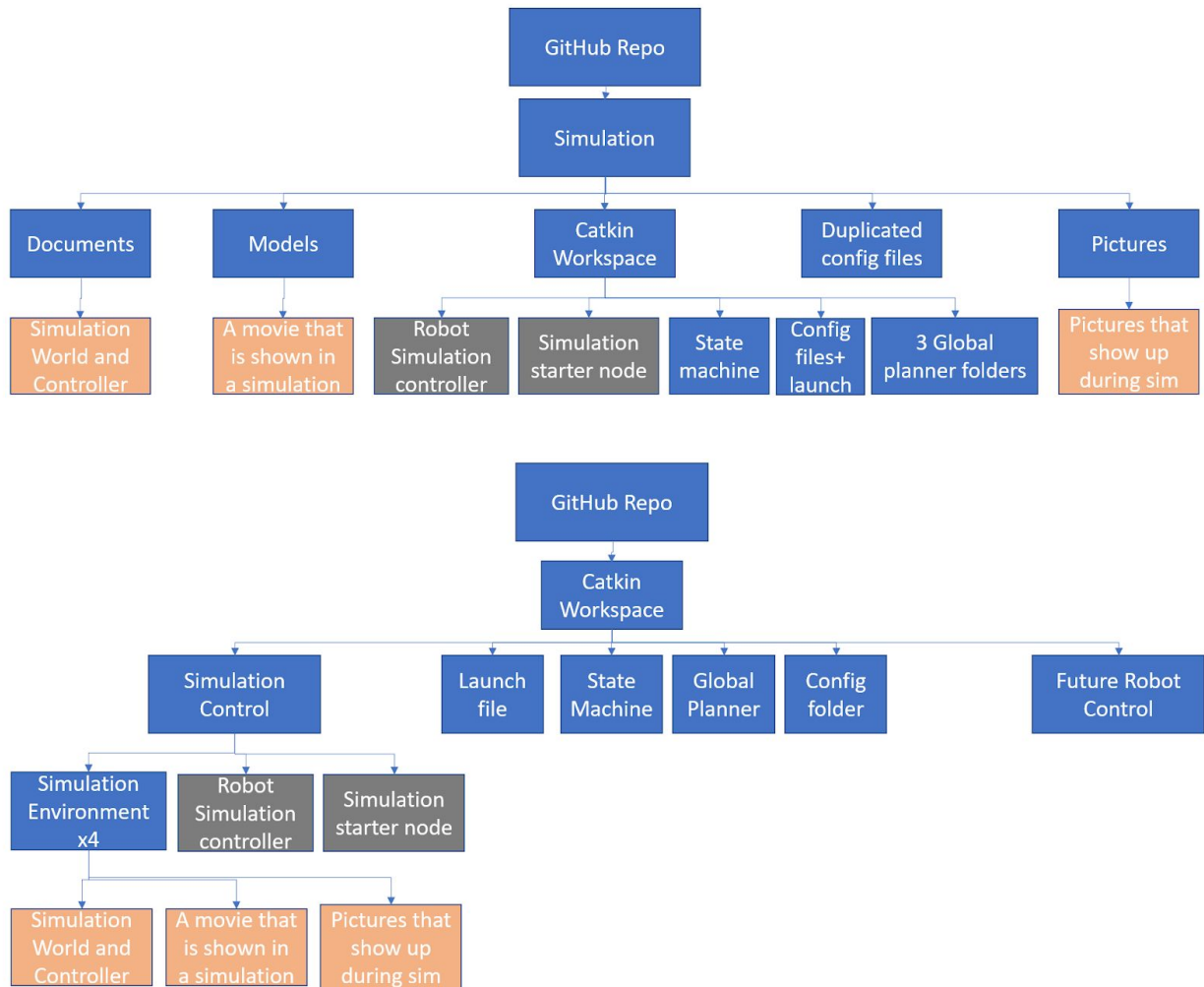


Figure 1: The file structures for the rover code before (top) and after (bottom) the separation. The file names have been changed for clarity.

After the change, the total size was reduced to 4GB, and the simulation environment that gets loaded is determined by a parameter in the launch file. Ideally, this same parameter will be able to change from simulation to robot controls, but additional work needs to be done in the launch file to not launch nodes that are only used in the simulation.

### Stereo Vision

Lastly, I altered the simulation rover's navigational stereo vision to match the Engineering model produced by the Pit Ranger team. This will make it easier to confirm that our perception algorithms will work on the new rover and in simulation. The process involved editing the simulation's robot controller to have additional cameras on the back of the rover and making sure that the cameras had the same transformation from the base of the rover. One challenge was that there was only one field of view value for both directions of the simulation camera. I

was able to create differing FOVs by aligning the resolution's ratio to correctly portray the horizontal and vertical field of view.

## Challenges

### PitRanger

#### Risk Assessment

There are some lingering challenges with the Risk Assessment process. The two that are noteworthy are, that the initial estimation must be reasonable and that the Pit-Vantage action should be overestimated. These two problems can cause the rover to go over time if they are not properly managed.

The initial estimations of how long the rover should travel should be reasonable. Due to the rolling average update method, the initial estimates are considered on every trip to the pit, even if the rover has historical data for each action. Underestimating the time it takes to do each action will cause the rover to take more trips than it should. The calculator does account for some of this inaccuracy but cannot guarantee that the rover comes back from the final trip within the time limit. An overestimation will cause the rover to visit additional vantage points per trip, which increases the risk of each trip. So this requires that the initial estimates be reasonably accurate. How much inaccuracy is reasonable is inversely proportional to the weight given to the initial estimates vs the historical data. The reason this proves to be difficult lies in the next problem.

The Pit-Vantage action should be overestimated. This problem lies in the implementation of the math rather than in the math itself. Let's walk through it. First, the rover travels to the pit using its Lander-Pit actions and records the time. The rover enters the Pit location and takes its Pit-Vantage action to get to the first vantage point and records its time. The rover solves the risk assessment problem and decides to take four trips back to the lander. The rover performs its Next-Vantage actions and records their times, then records the time of its Pit-Vantage action and Lander-Pit action after performing them in series. The problem lies in the very first Pit-Vantage action. This is when the rover travels from the entrance to the pit to the very first vantage point, which is likely immediately in front of the rover. This adds a near-zero historical time action to estimate how long Pit-Vantage actions will take. It considerably pulls down the average and can make the rover run overtime. Weighting historical data heavily can make this problem worse. The bandaid that I have placed on this action is to overestimate the Pit-Vantage action to account for the drop in average from the first data point. It works because we can run the simulation over and over and fine-tune this parameter, but will need a proper fix before the flight code is ready.

# Teamwork

## Pit Navigator

Individual	Main	Sub	Description
Alex W.	Stereo Vision Simulation, Separating Rover and Simulation Code	Helping set up the Simulator on teammates' computers	I primarily focused on getting the Stereo vision running in the first week, then moved to reorganize the code.
Awadhut T.	Develop pit edge distance heuristic	Set up simulation	Awadhut focused on getting the simulation running in the same OS, Ubuntu16, kinetic, and came up with the plan to find the heuristic with Justin
Justin M.	Set up simulation	Develop pit edge distance heuristic	Justin focused on getting the simulation running in an upgraded OS, Ubuntu18 with neotic, then helped Awadhut with the plan for the heuristic

## Plans

### Pit Navigator

#### New Simulation Environment

Next week I will create a new simulation environment to unit test the pit edge cases. It will have easy access to 4 different slope types and be very fast to run, so that Awadhut can begin testing his perception algorithm on the different pit edges, on his laptop.

#### Separation of Rover and Simulation Code

While the separation of different simulation environments has been completed, significant work needs to be done to be able to switch into rover code. Ideally, the launch file would be able to check if it is rover or simulation code and open the corresponding nodes, but a differing launch file for rover or simulation could also work. The folder paths to particular pieces of code will need to be checked again to ensure that there will not be confusion with the robot controllers instead of the simulation controllers. There is some rover code that exists for our rover, Blue2, that will need to be adapted to fit this structure as well.