Robotic Mining Competition – Milestone 2

Liam Sapper (lsapper2020@my.fit.edu)

Faculty Advisor: Dr. Marius Silaghi (msilaghi@fit.edu)
Client: Robotic Drilling team (Previously the Robotic Mining Competition team), NASA
Meeting Times: Wednesdays, 4:00pm - 5:00pm; Fridays, 3:00pm - 3:30pm

Milestone 2 Progress:

Task	Completion %	To Do
1. Implement a simulator	100%	none
2. Design test vectors for main requirements to be verified by simulator	80%	Try to create numeric measurements to be. checked
3. Look up documentation of involved hardware	30%	Still need to look up documentation for chosen encoder, motor, and other sensors
4. Research relevant algorithms for autonomous tasks	100%	none
5. Develop navigation system	30%	Move past pseudocode phase, start building/testing base software

Having been rejected from NASA's Lunabotics competition, Electrical, Software and Navigation have been pushed into a single subsystem called "Controls". My tasks remain generally the same but new requirements are to be set for 2, 3, and 4. Task 4 somewhat merges with 5 because algorithm research will be done mostly for the nav systems now, although I will still be helping the team to create autonomous drilling (this likely will not involve much algorithm work).

Except on Oct 25, RMC's class told them they had to continue working on RMC and stop with the drill. Their whole process had already been stalled for several weeks, and this has only set the team back further. While I could work on some of the logic, it felt lackluster when the team still hadn't confirmed any real plans/designs for the hardware that would be communicating with the software.

Task 1:

This was completed during milestone 1.

Task 2:

We have a general idea of how the waypoint system will be working, and have loose conditions set for some basic testing until there's a better understanding of what's being communicated by their choice of sensors and motors.

The requirements that need to be passed for the software to be successful are as follows:

- Successfully set and store waypoint information
- Be able to navigate to each waypoint
    - Acknowledge it has reached a waypoint
    - Turn in the proper direction of next waypoint
    - Start moving forward towards next waypoint

Those are the requirements for our current navigation plan, to mark certain waypoints on the first trip manually, then have the robot be able to follow the waypoint path on its own. This is the minimum we want with the navigation as of now. If possible, I would like to add to this so that the robot can navigate and create the waypoints on its own, and be able to avoid added obstacles that were unaccounted for.

I've tried my best to focus more on the navigation aspect, but the shaky planning and lack of communication from certain teammates due to illness or family emergencies has set the whole project back.

Task 3:

I have been slowly looking through the documentation for the computer that will be used, Raspberry pi. I'm still trying to obtain one for myself so I may start getting familiar with it. Although the necessary components for navigation are not fully decided, I have done some research to better understand encoders. They work by counting the number of rotations of the motor as a certain component is moving. Calculating the distance would be something like this: distance = (motor rotations) * (wheel perimeter / gear ratio). For our team the encoder will be working in tandem with the IMU (inertial measurement unit) which will measure the relative acceleration of the bot, and well as the angle of the acceleration.

I will look up documentation on the specific motor, encoder and other sensors to be used when that decision has been made.

Task 4:

Once again, focus has been more on navigation. Our team's plan is to mark waypoints for the bot to follow ahead of time and, should there be any unexpected obstacles encountered by it, adjust those waypoints as it goes along. Again, this second part will only be sought after once the original plan of manual waypoint marking is successful. I've found several resources that may prove useful and developing both an autonomous waypoint system and a manual system of navigation to solve this problem.

One specific source is a document outline autonomous waypoint navigation from 2008, by Yang Wang, David Mulvaney, and Ian Sillitoe from Loughborough University, and Erick Swere of the Scottish Association for Marine Science. Their system essentially defines a waypoint as a point where a robot begins to change its trajectory. This means the robot relies on the external behavior output instead of an internal state that may not always match that specific behavior.

They've split their strategy into two units; reactive and deliberative. The deliberate unit creates a path out of waypoints, and the reactive unit navigates the robot along that path, keeping track of each waypoint it reaches and the next one it is aiming for. The reactive unit allows the robot to create new waypoints as it encounters new obstacles, and the deliberate unit allows for keeping records of these waypoints and even deciding on if it should cut certain waypoints out based on prior and newly given knowledge.

Task 5:

Currently, there is only python pseudocode for the navigation system. Some of it is usable in our Webots simulation with adjustments, but I've been hesitant to go farther with testing as the team still hasn't finalized the kind of sensors and the number of sensors that will be used for this, as well as other components that contribute to the math, like the wheels and gears.

There was also the possibility of the team being disbanded, at least according to their professor at the time. If I develop the code based on assumptions, it could get complicated making adjustments to functions where there aren't enough variables or the wrong variables are being accounted for. Unfortunately our team member that was mainly responsible for navigation has been dealing with health issues and a family emergency. Upon reflecting, I still should have done more with developing the actual software, as there was no real point to some of these anxieties.

Milestone 3 Plan:

Task
1. Implement code in simulator that passes test vectors
2. Implement unit tests for verifying simulated code
3. Continue researching algorithms for autonomous tasks, look up libraries for selected algorithms.
4. Implement/adjust any missing/existing techniques and tasks.

Task 1: I will be working on developing the actual code and testing it according to set requirements from the previous milestone.

Task 2: Parts of the code will be tested for bugs before more of it is assembled together.

Task 3: I will be continuing to do research on autonomous tasks, especially with navigation. While the documents I have already found are extremely helpful, I want to see if there's a way to improve these waypoint methods that have already been done.

Task 4: Looking up more hardware documentation, continuing to. Develop the navigation system, confirm that the test vectors for simulation are enough.

Meetings with Client:

- 10/04/2023
- 10/06/2023
- 10/11/2023
- 10/18/2023
- 10/20/2023
- 10/23/2023
- 10/25/2023
- 10/27/2023

Feedback – Milestone 2

There has been some better communication. The whole team has been slow with setbacks keeping many of the members from doing work, or erasing the work that was previously done. While he still seems to be making progress with the logic and simulation, we're hoping he can do a little more to help the team understand the logic side of programming so, in the case he or anyone else is sick or has an emergency, someone else will be able to properly pick up that work. But for now, we are happy with the plan for navigation.

Meetings with Faculty Advisor:

- 10/27/2023

Feedback – Milestone 2

Focusing too much on the lower levels of the project, need to do more abstracting. Don’t worry so much about the details of the variables and just try to work on developing code that runs and can be put into the simulator.

- Faculty Advisor Signature: ______________________________ Date: ________

Faculty Advisor Evaluation

Liam	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

- Faculty Advisor Signature: ________________________________ Date: ________