MERA Team

Technical Report

Polytechnic University of Puerto Rico

San Juan, Puerto Rico

Team Members:

Gerardo M. Burgos
Elvin Flores
Irving Santana Vega
Samuel Nieves
Lucille Duprey
Yurimar López Cruz
Manuel Díaz Hernández
Fernando Salinas
Jeffrey J. Ríos Colón
Michell K. Ortiz
Brian Zaragoza
Brian Vazquez
Yamil Martinez
Pedro Burgos
Gerardo Ortiz
Brandon Baker

Revised by Jonathan Holguino – Mentor of the team

Abstract - This technical report documents the changes that the MERA team, participating in the NASA Swarmathon Virtual Competition 2017, made to the mobility.cpp, drop-off.cpp, pick-off.cpp, search-controller.cpp, obstacle.cpp packages to improve the rover's ability to navigate and collection targets in a virtually constructed space. The attention was principally given to the optimization the ways the rovers navigated the terrain available, the way they approached their goals, with rotational and linear velocities commands, remembering old and new goals locations, and obstacle avoidance calls. Changes were tested in the simulation and run multiple times to test for bugs and disorders. The things we thought at first would be most important like the optimum tag cubes pick-ups and drop-off timing. But later on, realize that there should be a balance between target collected and obstacle avoidance because that affects the time available for the competition.

I. INTRODUCTION

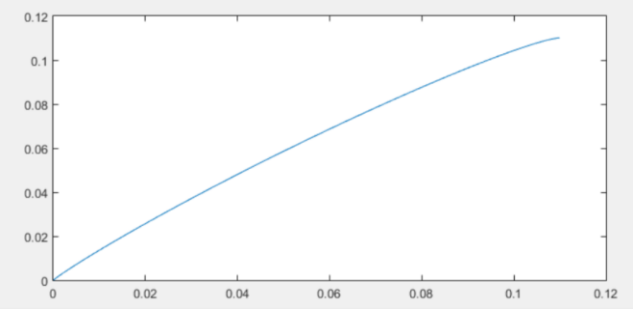
Our team's main goal was to understand the concept writing and interpretation of the code. Meanwhile making an optimization on the rover's search pattern and efficiency of the scouting area. The majority of the team of MERA were beginners to the collaborative team programming, but the time passing gave each of the team member the experience and feeling of being able to program a more challenging situation or environment.

In other to have a simulation, code, we needed to preload a USB flash keys with Ubuntu to start the GUI. Although Ubuntu was a new software for some of the members of the teams. The MERA team designated two regular meetings types, for example the regular meeting time was each Friday from 8:00am to 3:00pm and each time we ended the reunion there was an assignment given to each member for the next meeting. The other special case for meeting was at each ending of trimester we were given two weeks free, so we stayed from 8:00am to 3:00pm Mondays, Wednesday, and Friday.

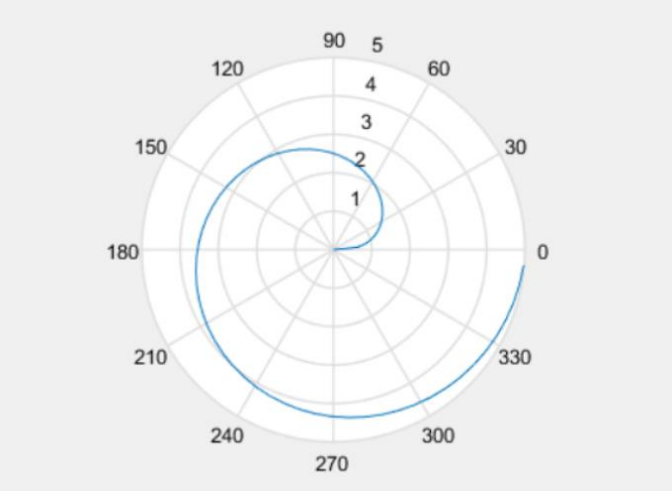
On each meeting, we had the main point was to examine the discovery of each member and circle the repetitive discovery if any, and after the brainstorming we would begin to program and run simulations, dividing ourselves in separate teams to examine the related performance, timing of the swarms, and discussed what can be perfected on the respective codes.

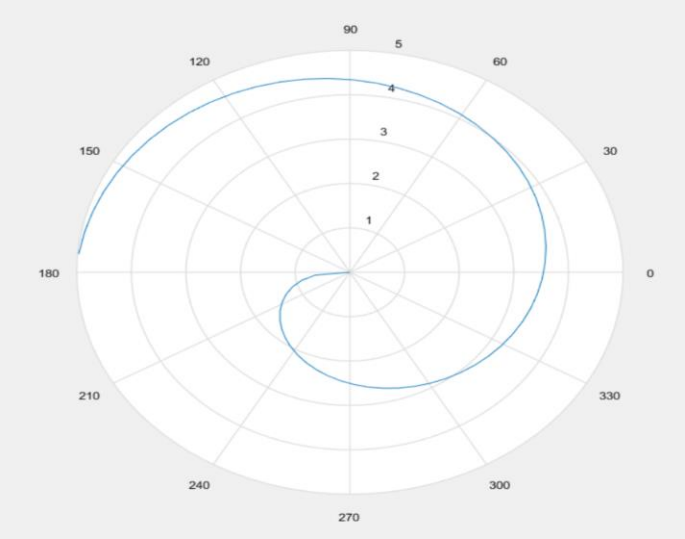
II. SEARCHING METHODS

The initial pattern that the searchcontroller.cpp has is the Gaussian formula, by plotting it looks like the next image:



After a few weeks of seeing the lost of time on search it would have we decided to implement different search pattern but in overall the best one was:





This is better known as the "Fermat Spiral". The main objective for this searching pattern is to minimize the time taken by the rover when searching for long distance targets. Besides this, the pattern will minimize the detection of farther targets, instead of ones that are relatively closer to the rover. With this

spiral, there will be better field sweeping of targets and overall efficiency per time interval.

III. RESULTS

The “Fermat Spiral” program was mainly program on MATLAB so there could be better simulation on graphs, and in addition we added the Gaussian distribution equation to calculate the possible random integer that the main program on the searchcontroller.cpp would establish. For the distribution of standard deviation, we took an average of 55 data point coordinates and made our calculation the result on pattern was previously shown. And with those result our team decided to apply all those polar like equation to every search type command on mobility and mainly on search controller.

After all the changes that were implemented on the rovers we saw that besides, the problems that the rover were having on locating closer tags, there were other serious complication like, the rovers lock base after avoidance calls, the rovers fighting for the same tag, and in some occasion the Swarmies going rogue all around the area bypassing all tags. The other three serious situations were that the rovers sometimes didn’t made the correct turns resulting on one of them tipping off the other one to the ground and falling on his back top. We often saw repeatedly the rovers getting some type of glitch once grabbing one tag and that tag would be embedded on his finger-claws, and the chain reaction was that the rover would not be able to secure another tag for the remaining time. The third situation were suspicious and random wheelie backs on the Swarmies.

We immediately commence on the new configurations on the program to adjust every error found on the mobility, drop-off, and pick-off packages by the team, relevant to the issues mentioned. We did discover ater that one of the main reasons for the lack of tag collections were mainly from the drop point of the rover on the base, for example once he placed the tags on the goal location he would make a 90 degree turn immediately causing itself to drag nearby tags to the outer perimeter of the center goal location. For the correction of the situation we made a backwards command velocity on the drop-off section, so that the rover wouldn’t extract more tags off the goal location.

During the last weeks of march, we began testing the resent updates and ran the program one time over another and began changing the distance of the sonars collision. In which thus gave more time efficiency, because the rover spent less time adjusting course because of the avoidance calls. All of these changes cause the rover to have an increase on collection rate and numbers.

IV. CONCLUSION

Our main changes were all targeted on the mobility package in which made a much more of an impact on the code itself and Swarmies. Apart of making good changes on the main code also one of the key turn point for this project is to make the rovers as efficient as possible, by improving movement, collection, driving, avoidance calls, targeting, and pick-up/drop-off all meanwhile having a big concern on the time usage. It was an important and inspiring to see how in such a short period compared to the immense practice of a software and program engineer have we could make a big change on the Swarmies performance everything with just the basics on programming skills. And over all that having an opportunity to work for a project of NASA itself, which make everything more exciting as students that have dreams at working closer to NASA.

These experiences will help our team grow, and at the same time help every team member to increase his/her programming skills, and the opportunity to see how an engineer implements its knowledge to a daily problem that we could confront on a near future. In addition, the NASA’s Swarmathon challenges has been of great learning development skills that can be directly applied to a professional field setting.

REFERENCES

- Morgan Quigley, Brian Gerkey& William D. Smart, “Programming Robots with ROS”, 2015
- Jason M. O’Kane, “A Gentle Introduction to ROS”, 2014