NASA Swarmathon Virtual Competition Searching and Retrieval of Objects in the Designated Field

Akane Simpson, Michael Hendrix, Lala Coulibaly, Amila R. K. De Silva, Jonathan Antee, Kennede Tran, Gary White, Mahalia Jackson, Jeffrey Clough, Busby Sanders, Klylena Mitchell

Academic Supervisors

Dr. Yenumula B Reddy and Dr. Jaruwan Mesit

Department of Computer Science, Grambling State University
Grambling, LA 71245

***Abstract*— Grambling robotics team initially build Vex robot and demonstrated the various functions that includes picking a ball and dropping in a tray, creating the robot with ultrasonic sensor and making the robot sense objects at a certain distance and turn to avoid, program robot to travel along line, and finally the robot follows the light using light sensor. We studied the Swarmathon simulation project and ran the simulations. The robot moved around but could not pick the box. Later we studied the source code and fixed the code to pick the boxes and bring to the central place. We modified the code and improved the sensing and picking the boxes faster to the central location.**

I. Introduction

The computer science students of Grambling State University are taking the participation in NASA Virtual Swarmathon challenges among the minority serving universities. The students understood that Swarm robotics is the study, design, and implementation of robot actions in the designated virtual field. The movements of robots are controlled and developed efficient algorithms to perform collective behavior of the robots to perform designated actions. In this competition, the students should not use physical robots. The must use and modify the code provided by Swarm group and modify the code in such a way that the robots in the field must perform the actions autonomously. The requirements of the Swarm compaction is to modify the code so that the robots takes autonomous actions and maximize their effort to collect the boxes in a specified time.

The principle goal of the team was to understand the Swarmathon code and improve the code to pick (collect) the boxes in the field to a central location. In the first step, The group analyzed the code and modified the code to sense the box and pick it to central location. The next step, they modified the code to continue picking the boxed to central location. Finally, they concentrated on time factor (efficiency) so that the robot moves complete field and pick the boxes faster to central location.

The project design and management was the approach used by the team for Swarmathon robots. The team's approach was divide the project into sub-projects and each person handles one sub-project part. The team leader then put the parts together and test the project. In the proposed Virtual Swarmathon project the Grambling team formed into eight groups. The code analyze group, algorithm development, implementation, and integration. The assigned work is as follows.

Akane Simpson	Overall project design
Akane Simson and Busby Sanders	Brisge
Michael Hendrix and Amila R. K. De Silva	Detection
Jonathan Antee, Lala Coulibaly, and Gary White	Git, Mobility
Kennede Tran and Mahalia Jackson	Tech Report
Jeffrey Clough and Klylena Mitchell	Algorithms

The team leader Mr. Simson observed that the simulation code provided by Swarmathon need to be improved. The robot in the field moves, senses the box in the field, and does not pick it. The first attempt is pick the sensed box and bring it to the central location. The second attempt is to search the remaining boxes in the field and bring them to central location. The search continues till all the boxes were collected.

This project objective is to enhance the code base of the swarmies so that they are better able to find, retrieve and return tags found in the simulator that would then be transformed in to real world usage. The project is broken down into 8 parts.

We focused on some key parts we believe will improve the swarmies ability to search. The parts are mobility, detection and algorithm structure. In mobility, we focused on the search controller, the pickup controller and the drop off controller. The most important part in mobility would be the search controller as it contains our search algorithm that helps the swarmie to find tags effectively. The pickup controller was the second most important part of this section as it would be the main control of moving these blocks back to the goal.

In detection, we had a small group focus on avoiding obstacles in the path by manipulating the obstacle detection to try and avoid colliding other robots and walls. The other minor teams had minor roles to play such as the git team that managed all changes to code and responsible of making the teams code available to all. The technical report team helped to document the process and life of the team working together.

The algorithm group modifies the code to improve the speed and networking of all box locations. They also worked on the location of each box and optimize the search and pick the boxes. The idea was discussed and convert it into practical. The code debugging is another part of the group. The discussion, design, analysis of code, implementation, and debugging, and testing was repeated till the deadline.

Searching the boxes in the field and retrieving them is main parts of the project. The efficiency is the next part of the project. Once these two parts are achieved, the students worked on autonomy of robots and pattern of picking the boxes. In order to achieve these goals, the robot need to receive the continuous signals (information) from base-station and sensors programed to boxes to be updated. The updated information helps to locate closest box and bring to central location or base-station.

To retrieve the boxes from the field, the robot must receive the information from base-station and environment. This increases the coordination between robot, environment, and base-station. The group working with coordinating effort of robot, environment and base-station provides the analysis of code. Once the coordinating analysis is implemented, the robot gain the real-time information to collect the boxes

Start the robot at a random place in the field and generate the default path to reach first box. If more than one robot in the field, they must communicate otherwise they collide. In the initial step, we must observe that the robots (if you use more than one) are communicating. Multiple experiments helps to confirm the communication between the robots. The algorithms must help least collisions, and least turning adjustments.

II. RELATED WORKS

Boston dynamics website has many examples [1]. BigDog is one of the example of the robot that can do many activities like walk, climb, and carries weights. Our high school students completed all these activities using Vex robotics hardware and programming. Similarly, Atlas for lifting items, PETMAN for testing chemicals, cheetah for fast running, SandFlea that drives like a car and jump 30 feet with safe landing.

Robot operating system (ROS) introduces the software platform to build code across multiple platforms [2]. It provides the tools and libraries for building robot operable code and robot operations. ROS operating system has low level device control, commonly used functionality, message passing between processes, and package management. Our students like to use, but due to time limitations they used Floral4 as well as C++ for vex robot operations. Later they changed to Swarmathon simulation code as per project requirements.

III. Methods

A. Searching

The DFS (Depth First Search) search algorithm was done by the lead programmer, who researched the depth first search. He researched C++ and pulled all the information he had from multiple sources with other group members. We chose DFS over UCS (Uniform Cost Search) because DFS was simpler and easier to program into the simulation. It took three days to program the DFS into the simulation. During the entire time, he was trying to get stack and graph functions to work bur was hit with multiple errors. After, much time and effort he figured it out with the help of the other programmers.

The DFS is a search algorithm that tells the robots in the simulation to search in a key area so they do not go to opposite ends of the map. The programs graph function tells it which area to keep searching and locks in the position where it finds a target and limits its search to that area. The stack function stores the location of the target from the graph and if no new targets are in the area it will delete the area out of the memory and wait for the graph function to store it in the next area.

The main function calls both the stack and the graph function and syncs them together so that they work in unison.

B. Mobility

At first, the robots would not pick up the targets at the first attempt, which caused it to enter a loop of attempts until the swarmie is able to pick up the target. By changing the trials to be able to pick up a target and we did not always drop the target into the destination. Some other robots picked up the targets that one robot dropped in the destination. We were not avoiding the obstacles. The robot was not targeting its targets.

We changed lockTarget to true because most of the time the target block was less than the target distance. We also changed the centerSeen (Central collection) to true because we wanted the robots to put their target in the center of the goal. The mapCount was changed from 0 to 10 to your memory of the mapAverage array. We increased the mapHistorySize because we wanted a bigger selection of points in order to calculate the map average position.

IV. Experiments

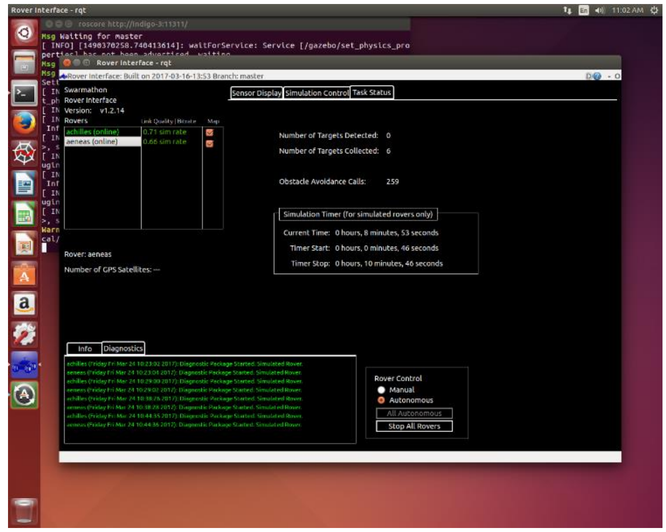

Figure 1. Image shows performance of swarmies in one of the experiments

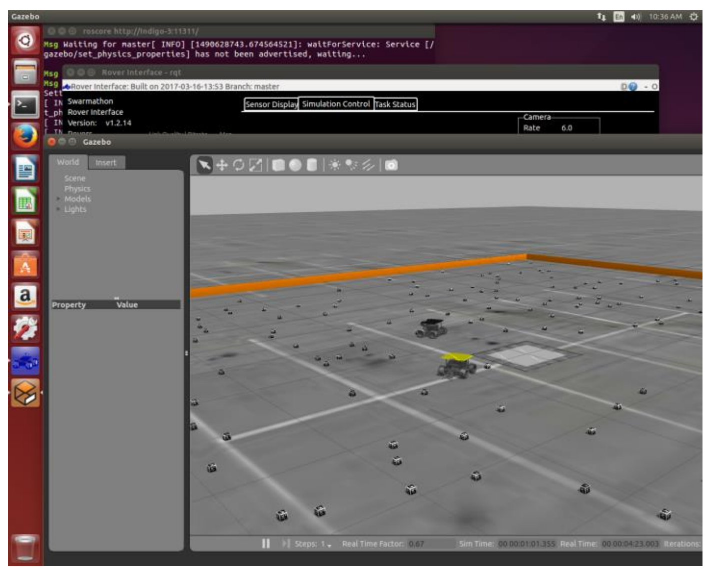

Figure 2. Screenshot of two swarmies locating targets in simulation.

The project had conducted several tests for changes in the code to better understand the scope of certain variables in the mobility, Search-Controller, Pickup-Controller and the Drop-Off-Controller C++ files. With each adjustment, it revealed more problems that the team needed to address and solve. This, along with hardware problems in installation, broken packages and older graphic drivers, made it difficult to use the ROS Indigo simulator to its best potential.

The testing condition for each experiment used only two rovers. Each experiment required the use 256 tags in either clustered or uniform setting to see how the swarmies would operate.

V. RESULTS

In the initial experiment the robot would search for blocks randomly, ignoring those that are closest to it. This increases the time taken for the blocks to be retrieved and thereby reducing the tag score for the round. In the second experiment, there was a noticeable change in the behavior of the swarmies when gathering tags, the number of blocks increased from 2 to 5 in the span of 10 minutes. In the third experiment, no changes in the simulation at all after adjusting the speed of the swarmies in the mobility file. In the fourth experiment the search algorithm was included in the SearchController header. This resulted in an increase in tags retrieved from the play area, with a ratio of one tag per minute for one swarmie. Learning from the previous test the team learned that other parts of the robot needed to be adjusted such as the Pickup Controller and the Drop Off Controller. These changes would help mitigate the problems seen in the simulator with failed attempts of retrieving the tag as well as long wait time to return to the center goal. In the last experiment, there was an unintended issue with the team's system that cause several of our packages to fail on build. This in turn prevented us from testing our code and to focus on debugging the code. The problem packages that the team found were the diagnostics package and the mobility package.

VI. CONCLUSION

In summary, the team attempted their best, however, were not able to complete the search algorithm as best as they desired. Due to time constraints, lack of experience, and installation issues the Grambling State team struggled to make deadlines and to perform any meaningful changes to the source code of the swarmies. This inhibited our ability to perform our best, leaving much room for improvement and lessons to be learned.

REFERENCES

[1] http://www.bostondynamics.com/robot_bigdog.html
[2] https://cse.sc.edu/~jokane/agitr/agitr-letter.pdf