University of North Carolina at Pembroke
AI and Robotics Class Team

Technical Report:
National Aeronautics and Space Administration Swarmathon Competition (2017)

Faculty Advisor:
Selvarajah Mohanarajah, Ph. D.

Student Team Members:
Heather C. Giotis
Nathan N. Kenani
Jonathan R. Kistner
Bayron J. Perdomo

***Abstract*—In the NASA Swarmathon competition a small group of robots searches for and returns resources to a centralized location. The competition includes Virtual and Physical participation options for teams across the country. All student teams are provided with the same original base code onto which they may implement improvements that run autonomously onboard the robots in a Robot Operating System. This report documents the completed changes that the team at the University of North Carolina at Pembroke undertook to improve the mobility package of the NASA Swarmathon robots, as well as the team's plans for ongoing improvements to the project following the competition submission deadlines.**

Keywords - swarm; robotics; artificial intelligence, ROS, localization

I. INTRODUCTION: KNOWING YOU KNOW NOTHING IS THE FIRST STEP TO LEARNING

The UNCP Team's experience is ultimately focused on the experience of learning and exploring. The NASA Swarmathon competition was undertaken as part of a Spring Semester introduction to AI and Robotics Class. Our team consists of beginners to every possible part of the collaborative programming process, AI algorithms, the ROS environment, and even the C++ language we determined we would use.

Initially, the process of understanding the basics of AI began within the classroom. Our Faculty Advisor and teacher Dr. Mohanarajah eased us in to search algorithms, understanding heuristics, localized searches, game theory and other areas of AI. Always the question of "How might we apply this to something like the Swarmie Robot?" was at the forefront of our learning experience in class. This often helped us translate our understanding of the abstract concepts taught into a real-world application we would be expected to implement in the coming weeks. Understanding not simply how an algorithm worked, but how it might be useful to our own goals became a lens through which we engaged with the material in classroom.

Prior to the mid-term break the team met infrequently, working on getting Ubuntu 14.04 installed, the Swarmathon ROS and libraries loaded. Gradually we worked on progressing our way through the ROS tutorials made available and resources on the Swarmathon discussion boards. Following the break we began meeting more frequently, focusing in on the various issues with the current implementation that we wanted to address with our own solutions. Our broader project goals have always laid out beyond the more modest goals we set for our small team to accomplish by the competition deadline.

II. RELATED WORK IN ALGORITHMS AND SEARCHES

A. *Utilizing Different Searches:*

Our team's current project plan with regards to Algorithm Development centers on a three step search process coupled with an Array of Quadtree utilized to track searched locations. The first step, a spiral search for five minutes to gather those targets closest to the center location, is the one we undertook to complete prior to the Swarmathon competition deadline.

The first phase of the search is a Archimedian spiral utilizing line segments outside the 50 meter central dropoff point surrounding the center location at (0,0). After entering starting positions, each rover with search in a line segmented spiral pattern for five minutes moving outward from the central dropoff area. The goal here is to collect data points for the Array once it is implemented, while also collecting those targets closest to the goal area in our first search pattern.

B. *Search Algorithms: A* and Developing Heuristics*

A form of best-first search, A* Search Algorithms evaluate potential pathways to a goal by combining the cost to reach the a node in a pathway *g(n)*, and the cost to get from the node to the goal *h(n)*.

$$f(n) = g(n) + h(n) \qquad (1)$$

Since g(n) gives the total cost from the start node in the path up to the present node, and h(n) estimates the cost to the goal from the current node, this means f(n) represents the estimated cost of the cheapest solution through n. A* search is usually utilized in traversal of trees, graphs, or arrays. Our work in this area heavily depends on final planned implementation of our Array or Quadtree for mapping locations as they are searched to inform our final algorithm and develop a heuristic for the search. We are currently examining both A* search and a random walk with testing for optimization to determine which will be the final stage of the search

C. Local Search Algorithms and Random Walks

Initially, the path to the goal for an individual Swarmie is irrelevant. The issue is in finding the target itself. In real world applications for Swarmies, a resource may not be tapped after the first instance of collection, and evaluation of both the expediency of the path and the remaining value of the resource may be required in the search algorithm. With this in mind, we set out to examine search algorithms that don't initially search out the best path to the goal.

Local search algorithms operate utilizing a current location and evaluate movement to neighboring locations. Local search algorithms such as random restart will be implemented together with our Array or Quadtree and a random walk with a preference for unexplored spaces. Initially, an informed modified version of the random walk will be utilized as the second phase of the search. Again, this section of the algorithm and the A* will be implemented in our final project, but we do not anticipate testing will be completed by the Swarmathon Submission date.

III. Related work in robotic Optimization

The second area of concern for the swarmie was in optimization of the swarmie robot and it's interactions with the world. Our team's current project plan for the optimization of the robot revolves around 1) Correcting for ODOM Drift over time. 2) Improving Goal Location Recognition 3) Improving Collision (and wall) Avoidance. For the Swarmathon competition deadline, we chose to focus on the issue of Correcting the ODOM Drift over time, anticipating it may help with our other stated goals in the project.

A. Localization

There are three frames of reference that localization for the Swarmies work under. The Local Frame is the lowest, a 3-axis grid with an X-axis aligned to the Swarmie's current heading, with all movement happening along that X-axis line. The ODOM frame is a EKF that fuses data from the IMU, encoders and camera link. The Map Frame adds GPS data to the ODOM.

B. The Problem With Noise and Drift

Although the ODOM Frame is the best for the robot to drive in and is accurate in the short term over smaller distances, it experiences a problem of a potentially uncapped drift as small incremental discrepancies in physical location occur with calculated location. Over time these differences can be huge, and result in a Swarmie dropping collected targets outside of the goal area, or ending up completely off course from an intended search location. Meanwhile, the Map Frame experiences a noisy jump from the GPS, with discrepancies of up to a few meters. Obviously, switching to the Map frame alone would not solve the issue. The ideal solution became apparent: We needed a way to correct the ODOM's drift over time, preferably without being subject to the jumps of the Map Frame's GPS.

IV. Methods

A. Flow Charts

Understanding the current mobility package fully and identifying how it worked and decisions were made became a top priority as we began discussing issues and how they might be solved. We were careful to understand that we were augmenting a program, and needed to be sure of how mobility interacted with the other controllers in order to achieve the most sensible solutions in context of the current implementation. While the entire team expressed an interest in exploring writing our own implementation from scratch at some point in the future, a part of the challenges seemed to be finding elegant solutions to the problems within the current framework. Because of this, decision making flow charts for the mobility package as it is and as we intend to alter it became an important part of our planning process.

University of North Carolina at Pembroke

V. Experiments

The majority of our experiments took place in the Virtual Environment. Although we had originally intended to collaborate with another College for a Physical Competition their withdraw from the competition ended those plans and focused our efforts to the Virtual Competition. While working on implementing changes we utilized ten minute incremental searches until successful implementation had been achieved. Because we are planning to do optimization tests at the end of the project, we did not engage in those test at this stage of the work.

VI. Conclusion and Further work

With the completion of our first tasks, those of the Spiral Search Algorithm and the optimization of the localization, we have begun to move on to the second phases of both search algorithm optimization and robotic optimization, Improving Goal Location Recognition and the Implementation of the Array or Quadtree to map searched locations. The opportunity to learn so many new skills has been exhilarating to our Team's collective imaginations, and carried us into continuing our efforts beyond the Swarmathon Competition.

[1] Stuart J. Russel, Peter Norvig, "Artifical Intelligence: A Modern Approach, 3rd ed. Pearson Education, Limited. 2015