ENPM700 Final Project Proposal: **Project NavGuide**

Rishie Raj
120425554

Uthappa Suresh Madettira
120305085

1 Introduction

We propose designing a swarm multi-robot system for warehouse applications at Acme Robotics, aimed at improving safety and efficiency in material handling operations. This system is vital in a warehouse environment where there are multiple tasks that need to be accomplished in parallel. The goal of the multi-robot system will be to navigate the warehouse environment successfully and reach there destinations.

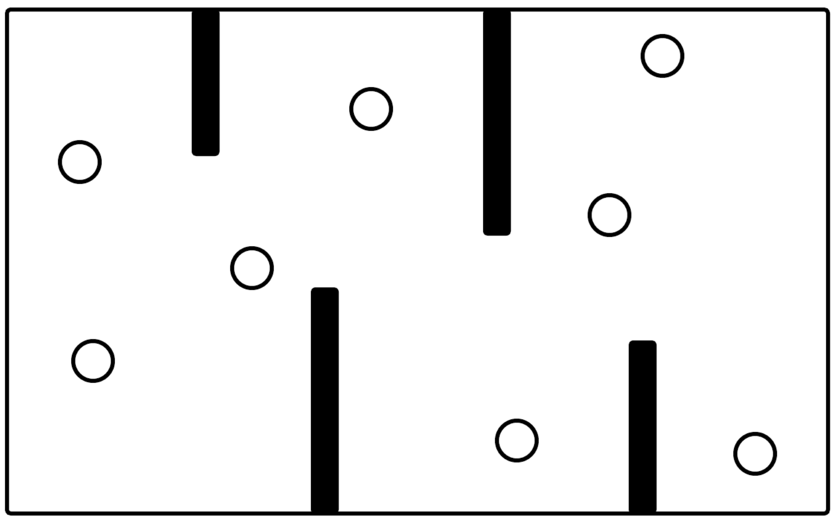

Figure 1: Warehouse Environment for Swarm System

The multi-robot system will be made up of ~ 20 TurtleBots that will navigate a custom built environment as shown above. They will localize each others position in real-time using the LIDAR sensor available as part of TurtleBot3.

2 Importance and Usage

A multi-robot swarm system is essential in a warehouse because there is a need for robots to interact with each other to perform tasks. This particular module is built in a self-contained manner such that it can be integrated directly into Acme Robotics products as a package and run as part of the whole system.

The software module designed by us will be used in the Acme warehouse robot for safe navigation on the shop-floor for which it requires detection and tracking of human obstacles.

3 Tools, Technologies and Algorithms

In order to build our module, we will be needing the following tools and algorithms.

- **RVO2 for Collision Avoidance:** An efficient algorithm for Optimal Reciprocal Collision Avoidance which can be used to navigate swarm systems.
- **ROS2 Humble:** The middleware that we will be using for implementation of this project.
- **Gazebo 11.0:** The physics simulator that will be used to visualize the implementation.
- **TurtleBot3 Burger:** The hardware platform that will be used for the simulation.
- **C++:** The primary programming language for real-time robotics applications.
- **CMake:** For building the system.

4 Design and Development Process

We will follow the Agile Iterative Process (AIP) for development. Therefore the project will be divided into 3 iterations based on the timelines provided for submissions. Each iteration will improve upon the previous implementation. The 1^{st} iteration will be for Phase-1 of the implementation and 2^{nd} & 3^{rd} will be for Phase-2. We are also using the the pair programming approach coupled with test-driven development. So the number of work hours will be divided equally between the two programmers. Every task in the product backlog sheet will have a driver and navigator in order to meet the pair-programming requirements.

Pair-Programming Schedule			
Role	**Iteration-1**	**Iteration-2**	**Iteration-3**
Driver	Rishie	Uthappa	Rishie
Navigator	Uthappa	Rishie	Uthappa

5 Risks/Bottlenecks

A major bottleneck for us will be the Gazebo simulation of nearly 20 robots as this is quite resource intensive. Hence we have to figure out a way to make it less intensive on the graphics so that the simulation can work properly.

There is also a learning curve for the RVO2 collision avoidance algorithm that we need to familiarize ourselves with to be able to integrate it into our product. We plan to try this out on a reduced scale first so the we understand the functionalities first and then scale to the product level.

6 Final Deliverables

The final deliverable to Acme would include the complete software module that when built and executed, will facilitate the navigation of multi-robot swarm system. Acme Robotics will also be presented with a demonstration where the swarm functionality of in a virtual Gazebo world will be shown by us on TurtleBot3 robots. The source code will be passing all test cases with a code coverage of greater than 90%, along with proper Doxygen documentation and Google style formatting.

7 References

[1] RVO2 Official Documentation: `https://gamma.cs.unc.edu/RVO2/`

[2] ROS2 Official Documentation:`https://docs.ros.org/en/humble/index.html`