Autonomous Search and Rescue using swarm robots

Codename: SeekX

Proposal by: Mayank Deshpande (120387333) & Lowell Lobo (120095719)

I. Overview

A. Purpose

Acme Robotics requires a 5-yr plan for novel R&D application in the area of multi-agent or swarm robotics. This proposal focuses on their implementation in area of and Search and rescue (SAR). SAR operations are often challenging and hazardous, requiring personnel to navigate complex environments and locate individuals or objects in distress. Traditional SAR methods rely heavily on human operators, which can be time-consuming, resource-intensive, and potentially dangerous. The use of robotic systems offers a promising alternative, providing enhanced capabilities and reducing the risks associated with SAR missions.

B. Objectives

This project proposes to develop a simulated robotic system for search and rescue using ROS2 Humble, Gazebo, and multi-agent or swarm algorithms. The system will employ twenty robots to autonomously search a designated area in Gazebo to locate a specific object and provide its coordinates. The project aims to demonstrate the effectiveness of multi-agent approaches in coordinating and controlling multiple robots for SAR operations.

C. Scope

The project scope encompasses the development of a simulated robotic system focusing on autonomous search and rescue operations through a swarm of robots. The swarm will consist of a minimum of 20 robots, operating collaboratively in a Gazebo simulation environment. Implementation of multi-agent or swarm algorithms that facilitate efficient exploration and collaboration among the robots along with object recognition and localization algorithms to identify the target object is to be done.

D. Assumptions

- The project assumes that a map of the search area is provided in advance to the robots. This map serves as the basis for the simulation environment in Gazebo.
- It is assumed that the target object to be identified by the robots is stationary and not in motion during the course of the search and rescue operation
- The nature of the target object is assumed to be known in advance. In this scenario, the robots are programmed to identify a specific object based on predefined characteristics.
- While the target object is stationary, the initial position of the object within the search area is randomized for each simulation run. This adds variability to the robots' exploration and detection strategies.

E. Constraints

Simulating a large number of robots within a restrictive space and ensuring no collisions while considering optimal autonomous path planning is a challenging task. The runtime fps and memory management depend on the physical constraints of each system. Latency in terms of communication between all 20+ robots depend on the robot hardware as well as environmental factors. Bad weather or remote locations can lead to bad communication.

F. Deliverables

- C++ library/API based on ROS modules to perform search operations using 20 robots.
- Simulation of the project in Gazebo.
- GitHub repository of the project and README.
- Continuous Integration using GitHub CI and code coverage using CodeCov.
- Developer Level Documentation.

G. Evolution

Possible areas of evolution include:

- Integrate dynamic and changing environments within the simulation, requiring the swarm to adapt to evolving conditions.
- Expand the scope to involve multiple types of objects with varying priorities for identification.

H. Summary

The project will be completed within three weeks following the AIP model and pair programming. The project aims to showcase the potential of autonomous robotic swarms in search and rescue operations, with a focus on collaboration, adaptability, and efficiency in complex environments. The predefined map, stationary target object, and Gazebo simulation provide a controlled yet realistic platform for development and demonstration.

II. Reference Materials

Reference material consists of mainly C++, CMake, OpenCV, ROS2, Gazebo, and Turtlebot3 documentation. Documentation for the project will be present as either Doxygen documentation or GitHub README and will be present in the GitHub Repository.

III. Process

The project will adhere to the Agile Iterative Process (AIP) principles, employing Pair Programming for collaborative development. Driver and navigator roles will be alternated at defined intervals to ensure a balanced contribution from team members. The development process initiates with the setup of the Gazebo environment, including the world, area map, and robot spawns. Robots autonomously navigate through the map, systematically searching for specified objects. This iterative process continues until the target object is located. Upon finding the object, the discovering robot communicates its location. Testing of components is performed using GoogleTest, and system testing will be performed every iteration for overall functionality verification. Furthermore, Level 2 integration tests will be performed to evaluate performance of nodes in the global level

A. ROS Enhancement Proposals (REPs)

- **REP 105: Package Naming Conventions**: This REP establishes conventions for naming ROS packages, ensuring consistency in naming across the project.
- **REP 148: Standard for Robot Operating System Environment Variables**: This REP defines a standard for environment variables used in ROS, ensuring uniformity in configurations across the project.
- **REP 2005: ROS 2 Common Packages**: This REP defines a set of common packages for ROS 2, which are relevant to the project because it will be using ROS 2.
- **REP 2006: ROS 2 Vulnerability Disclosure Policy**: This REP defines the ROS 2 vulnerability disclosure policy, which is important for ensuring that the project is secure.

B. *UML and Activity Diagrams*

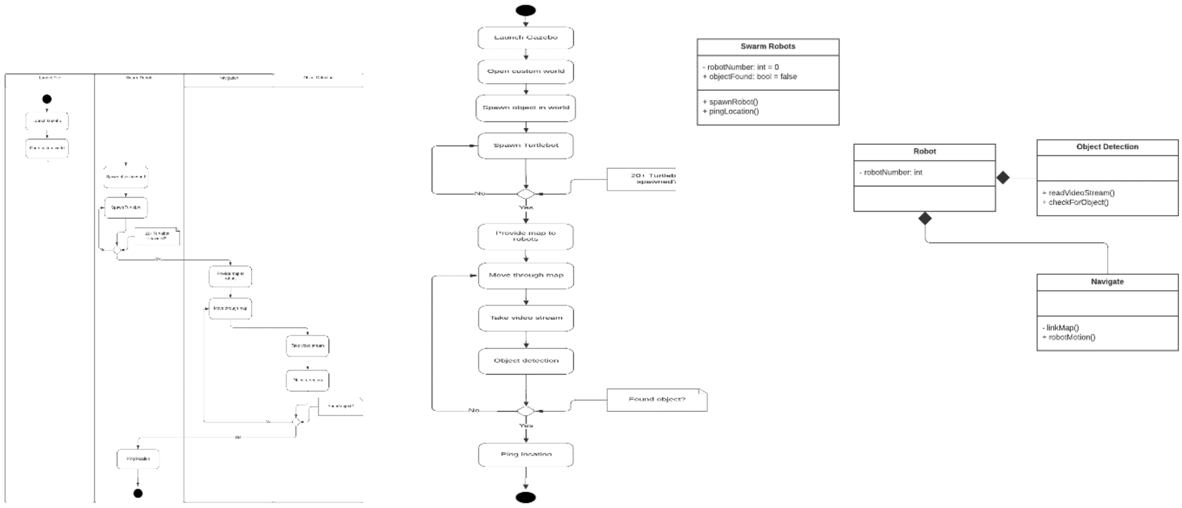

Figure 1 [Swim Lanes](), [Activity diagram]() and [UML Diagram]() of the model

IV. Organization

The project will follow a Pair Programming approach, where the roles of Driver and Navigator will be dynamically switched based on the ongoing implementation. Initially, Lowell will take on the role of the Driver, responsible for writing code, while Mayank will be the Navigator, providing feedback and suggestions. This collaborative interchange ensures a balanced contribution from both team members, fostering effective code development and knowledge exchange.

V. Technologies

The project environment is centred around the ROS2 framework for simulation and robotics, operating in a C++ development module with CMake build tools on Ubuntu 22.04. OpenCV, an Apache2.0 licensed library, is employed for image processing, while Turtlebot3, also Apache2.0 licensed, serves as the ROS2 package for simulating Turtlebot3 in Gazebo. Object detection relies on OpenCV's HSV manipulation algorithms. GitHub functions as the version control platform for collaborative development, code hosting, and code coverage reports, managing unit tests for code functionality assessments. The technology stack emphasizes compatibility, efficiency, and open-source principles through the use of Apache2.0 licenses.

VI. Management

The project will take three weeks to complete over three iterations, consisting of Phase-1, Phase-2 and Presentation, where each iteration is for one week. Since the project consists of only two members, only product backlog, daily meetings and iteration meetings in the AIP model will be performed. Project monitoring will be done using timely git commits, and the navigator ensures code quality. Daily meetings will be held for error correction based on code coverage and GitHub CI reports.