SeekerSwarm : Swarm For Locating Lost Objects In The Warehouse

ENPM808X Final Project Proposal

INTRODUCTION

In the ever-evolving landscape of warehouse management, the need for efficient and intelligent solutions has become paramount. In collaboration with Acme Robotics, we embark on a groundbreaking project to enhance their warehouse operations through the implementation of a multi-robot swarm system. This project aims to leverage the power of the Robot Operating System (ROS) to develop a sophisticated swarm of more than 20 TurtleBots capable of autonomously locating misplaced and valuable items within the warehouse. We plan to simulate the functionality in Gazebo and RViz before the deployment in the actual warehouse environment.

PROBLEM STATEMENT

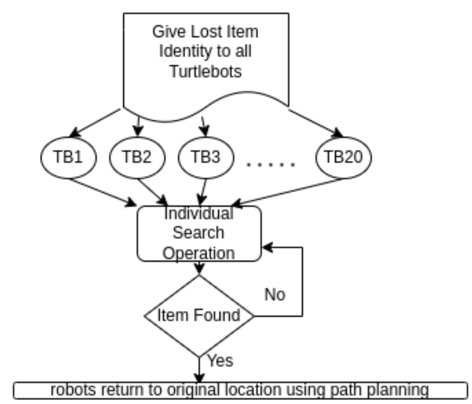
Acme Robotics faces a common challenge in warehouse logistics — the timely and accurate retrieval of valuable items that are often misplaced or stored suboptimally within the facility. The conventional methods of manual search and retrieval not only consume valuable time but also introduce the potential for errors and inefficiencies in the overall workflow. To address this challenge, our project focuses on the development of a state-of-the-art swarm system that utilizes ROS as the foundation, coupled with vision-based methods or Aruco markers for precise object detection. The swarm of TurtleBots will operate in tandem, scanning designated areas of the warehouse comprehensively and systematically. Upon detecting a lost item, the swarm will promptly report its location to the operator, and all the robots return to their respective home positions.

We plan to validate the functionality by simulating it in Gazebo and RViz to ensure that the swarm operates seamlessly and effectively in the actual operational environment.The successful implementation of this swarm system promises to significantly enhance Acme Robotics' operational efficiency, reduce costs associated with item retrieval, and set a new standard for intelligent warehouse management system.

METHODOLOGY

As a proof of concept, we plan to simulate our idea in a Gazebo environment with a group of turtlebots moving and communicating with each other in a simulated warehouse space. The algorithm proceeds as follows: Firstly, all the turtlebots are deployed at different initial locations in the warehouse space. Once the bots are deployed, each turtlebot starts a search operation around some specified radius to look for the lost item. Every turtlebot scans the Aruco marker present on the lost item which is unique for each good in the warehouse/ utilizes the camera to scan for the item using opencv.

While each turtlebot starts exploring, SLAM is utilized to localize it and avoid dynamic obstacles such as other turtlebots. It can also be used to generate both local and global maps of the environment for better optimization which is an ambitious goal for the project. If any of the turtlebots find the lost item, its coordinates are reported to the operator and other turtlebots. All the turtlebots move back to their original stations. Navigation packages in ros such as nav2 are utilized for turtlebots to generate an efficient path between their current location and goal location.



SOFTWARE DEVELOPMENT PROCESS

The project aims to follow the software engineering practice Agile Iterative Process which involves breaking down the project into tasks and organizing them into 3-week iterations. Throughout each week of development, there is a consistent focus on UML design and high-level project design. The team conducts regular sprint meetings and employs a collaborative approach to work utilizing tools such as Version Control Git, including the practice of pair programming. Furthermore, we plan to verify our implementation continuously using the Test Driven Development approach.

SOFTWARE TECHNOLOGIES

Programming Languages: C++, Python
Development Tools: Git, Cppcheck, Cpplint, VSCode, ROS2, Gazebo, RViz
Testing tools: Valgrind , GTest | Documentation: Doxygen
Continuous Integration and Code Coverage: GitHub CI, CodeCov

EXTERNAL DEPENDENCIES

System Requirements: ROS2 Humble, Ubuntu 22.04
ROS Libraries: Nav2, GMapping (SLAM), OpenCV Bridge, Aruco ROS, Gazebo
License: Apache License 2.0

ROS REPs (ROS Enhancement Proposal)

- REP 2001 -- ROS 2 Variants (ROS.org) : This REP serves as a guide for users to understand the available variants and choose the one that best suits their needs when working with ROS 2.
- REP 2005 -- ROS 2 Common Packages (ROS.org) : This REP describes the common packages of ROS 2. We make use of the existing common ROS2 packages for running simulations.
- REP 2004 -- Package Quality Categories (ROS.org) : This REP establishes a set of categories or quality levels to convey the maturity and quality of packages within the ROS (Robot Operating System) ecosystem. We propose to follow these guidelines.
- REP 103 -- Standard Units of Measure and Coordinate Conventions (ROS.org) : This REP establishes standardized units (SI) and coordinate conventions within ROS, aiming to reduce integration issues and computation errors arising from inconsistent practices. The document outlines base and derived SI units, as well as coordinate frame conventions, ensuring right-handed systems and providing specific suffix conventions for different frames.

POTENTIAL RISKS AND MITIGATION

1. **Dynamic Obstacle Avoidance:** As this is a multi robot environment each robot has to generate a path or search avoiding dynamic obstacles which could mainly be humans or other robots. To mitigate this we plan to study different planning algorithms available with the navigation package and choose the efficient one. Also, we would like to improve the robots localization through efficient SLAM implementation

DELIVERABLES

1. Professional-level source code with Unit tests and Developer-level documentation.
2. Functional user-runnable demo: A lost item will be spawned at a random location in Gazebo. All the turtlebots will move to scan the designated areas until one of them finds the object. Once the object is detected, its location will be displayed on the terminal, and turtlebots will return to their home positions. Mission accomplished!
3. Description of proposed algorithm, potential risks, technology used and dependencies.
4. Technical presentation.

TEAM MEMBERS AND ORGANIZATION

Driver	Navigator	Design Keeper
Neha Nitin Madhekar(UID:119374436)	Vinay Krishna Bukka(UID: 118176680)	Rashmi Kapu(UID: 119461754)

We have adopted the Test Driven Development (TDD) methodology. In the initial phase (Phase 0 and 1), Rashmi will assume the role of the designer, Neha will act as the driver, and Vinay will take on the role of the navigator. This follows a pair programming approach where the designer oversees the tasks and makes design changes . As we proceed with subsequent tasks, these roles will be rotated among team members.

REFERENCES

1. Software Engineering Current Practice by Vaclav Rajlich
2. A. Salunke, C. Patil, R. Mude and R. D. Joshi, "Simultaneous Localization and Mapping (SLAM) in Swarm Robots for Map-Merging and Uniform Map Generation Using ROS," 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, 2023, pp. 411-415, doi: 10.1109/ICCAE56788.2023.10111365.