FinderBots: Swarm Intelligence for Enhanced Object Finding in Warehouses

Datta Lohith Gannavarapu (UID: 119455395)
Dheeraj Vishnubhotla (UID: 120427367)
Nazrin Gurbanova (UID: 120426469)

Introduction

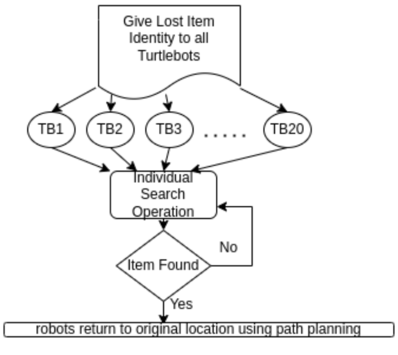
In the dynamic field of warehouse management, optimizing operations for faster and more accurate item retrieval has become essential. Partnering with Acme Robotics, this project proposes the development of an autonomous multi-robot swarm system, **FinderBots**, using the Robot Operating System (ROS). This system will employ a swarm of over 20 TurtleBots to locate misplaced items within the warehouse environment autonomously. Initially, simulations will be conducted in Gazebo and RViz to validate functionality before real-world deployment.

Problem Statement

Acme Robotics faces challenges in efficiently locating valuable items that are often mislocated in their warehouses. Traditional methods of manual search are labor-intensive, prone to human error, and hinder workflow efficiency. **FinderBots** aims to address these issues by developing a cutting-edge swarm system based on ROS. Utilizing Aruco markers and computer vision, the TurtleBots will scan designated warehouse areas in a coordinated manner. Upon locating a missing item, the bots will relay the location to the operator, after which they will return to their initial positions.

Simulations in Gazebo and RViz will confirm the swarm's robustness, ensuring operational efficiency in the warehouse setting. By implementing this system, Acme Robotics can expect increased efficiency, reduced item retrieval costs, and a transformative approach to warehouse management.

Methodology


Give Lost Item Identity to all Turtlebots
TB1
TB2
TB3
.....
TB20
Individual Search Operation
Item Found
No
Yes
robots return to original location using path planning

Our proof of concept involves a Gazebo simulation of TurtleBots executing a search algorithm in a warehouse. Initially, TurtleBots are deployed at various starting points. Each robot will scan a defined radius for lost items, employing either Aruco markers or OpenCV-based visual detection.

As each TurtleBot navigates, SLAM technology will support localization and obstacle avoidance, improving mapping accuracy through local and global environmental scans. Upon detecting the target item, the locating bot transmits the coordinates to the operator and other bots, followed by a return to base. The ROS Navigation stack will optimize pathfinding for the bots from their current location back to their home station.

Software Development Process

This project employs the **Agile Iterative Process (AIP)**. Development is organized into 3-week cycles, emphasizing regular sprint reviews, UML diagrams, and high-level design. Tools such as Git version control and pair programming are fundamental, with Test-Driven Development (TDD) ensuring continuous verification.

Software Technologies

- **Languages**: C++, Python
- **Tools**: Git, Cppcheck, Cpplint, VSCode, ROS2, Gazebo, RViz
- **Testing**: Valgrind, GTest
- **Documentation**: Doxygen
- **CI 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 Enhancement Proposals (REPs)

- **REP 2001**: Guide to ROS 2 Variants
- **REP 2005**: ROS 2 Common Packages
- **REP 2004**: Package Quality Categories
- **REP 103**: Standard Units of Measure and Coordinate Conventions

Potential Risks and Mitigation

- **Dynamic Obstacle Avoidance**: Path planning must account for moving obstacles like humans or other robots. Mitigation strategies involve testing various navigation algorithms and enhancing SLAM localization.

Deliverables

1. Professional source code with unit tests and developer-level documentation.
2. A demo showcasing item detection by the swarm, with item location output in the terminal and all robots returning to base.
3. Documentation of the algorithm, identified risks, technologies, and dependencies.
4. A technical presentation.

Team Organization

For Phase 0, these are the roles.

- **Driver**: Datta Lohith Gannavarapu
- **Navigator**: Nazrin Gurbanova
- **Design Keeper**: Dheeraj Vishnubhotla

Roles will rotate throughout the project, with adherence to TDD and a collaborative design-review approach.

References

1. Rajlich, V., *Software Engineering Current Practice.*
2. Salunke, A., et al., "SLAM in Swarm Robots," *International Conference on Computer and Automation Engineering*, 2023.