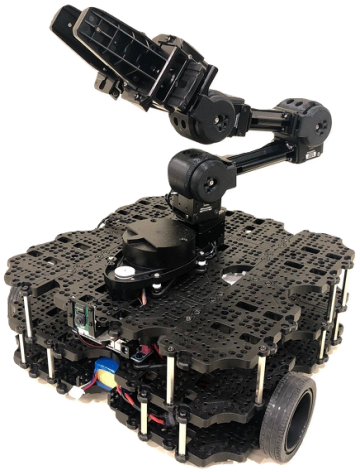
ENPM 808X: Software Development for Robotics

FINAL PROJECT PROPOSAL

Rad**R**over**R**escue: EcoSweeping Safely in Radioactive Zones (RRR)



Team Members - Group 5

Akashkumar Parmar - 118737430

Sai Teja Gilukara - 119369623

Overview

Robots are the solution for safely managing the growing volume of radioactive waste from nuclear power plants, medical facilities, and industries. Traditional methods involve risks to human workers due to radiation exposure. Acme Robotics offers a safer and more efficient alternative. Our robots can withstand radiation, operate in hazardous environments, and perform precise tasks, reducing human exposure and ensuring safe waste management.

Proposed Pipeline

The software development process for outdoor radioactive waste pickup will use Agile Iterative Processes and pair-programming techniques over three weeks. We'll verify the implementation systematically with Test-Driven Development (TDD). We'll exchange driver and navigator roles in each phase for better collaboration and efficiency while upholding software engineering practices.

Components of the Software Implementation:

Navigation
Objective: To navigate and locate radio-active waste sites in the outdoor environment.
Implementation: Utilize a grid search algorithm with the Navigation2 (Nav2) ROS package, incorporating Simultaneous Localization and Mapping (SLAM) for real-time mapping updates.

Perception
Objective: Detect radio-active waste sites using computer vision algorithms.
Implementation: Develop a computer vision-based algorithm for robust detection of radio-active waste sites in outdoor scenes. This may involve image processing, object recognition, and machine learning techniques.

Collection:
Objective: Implement a system to safely collect and contain radio-active waste.
Implementation: Design and integrate mechanisms for safe and secure collection. This may involve robotic arms or other specialized tools suitable for handling radio-active waste.

Transportation:
Objective: Efficiently transport collected radio-active waste to a designated disposal area.
Implementation: Develop a transportation system, potentially leveraging autonomous vehicles or robotic platforms, ensuring the secure movement of the collected waste to a specified disposal location.

Software Strategies and Technologies

We'll use C++'s Object-Oriented programming for scalability and adaptability. Our development environment in Visual Studio Code IDE will include clangd, cppcheck, valgrind, and cpplint for code analysis. We'll simulate using ROS2 Humble Hawksbill, RViz, and Gazebo. For cross-platform compilation, CMake and colcon will be our build systems. The project will be licensed under Apache 2.0 for open-source collaboration and contributions.

Strategies:

Pair Programming, Test-Driven Development, AIP

Technologies:

Programming Language and OS: C++, Ubuntu 22.04
Other tools for testing and development: Git, Valgrind, GoogleTest, cpplint, cppcheck, clangd, CMake
ROS Packages: Nav2, cv bridge, TurtleBot3
Continuous Integration and Code Coverage: CodeCov, GitHub CI
Libraries: OpenCV, NumCpp, Math, Eigen
Software Documentation: Doxygen

Risks and Mitigation

A potential risk lies in the trade-off between image processing speed for obstacle detection in the radioactive waste-picking robot's journey. The robot may encounter challenges due to its unfamiliarity with the environment and the use of a grid-based search algorithm, potentially affecting its ability to identify waste or reach the intended goal.
To address this concern, we will constrain the size and positioning of radioactive objects in world space. This approach enhances the robot's ability to locate and manage waste effectively. Our assumption is that the robot must navigate to the target, simulate pickup, and safely reach the disposal zone.

Final Deliverables

The project's main deliverables include a comprehensive ROS2-humble package with robust features for simulating radioactive waste detection and handling in outdoor environments. The package will also incorporate advanced visualization components to enrich the simulation experience and explore the robot's capabilities fully. Additionally, we will provide clear documentation using Doxygen and a detailed README for user-friendly usage.

References

https://emanual.robotis.com/docs/en/platform/turtlebot3/manipulation/#manipulation