METR4202 Robotics & Automation
Semester 2, 2023
Team Project Brief

Level 0: Frontier Exploration

You have made it through six weeks of setup, preparation, Gazebo simulation development and have learned to implement mapping and navigation strategies on a simulated TurtleBot3 robot using ROS2. Congratulations!

Now, armed with your new knowledge and experience, you have landed on Frontier Bay and you are given your first task:

Develop an exploration strategy for your TurtleBot3 to construct a 2D occupancy grid map that includes every reachable location in the environment.

Setup and Requirements

- **Domain:** Gazebo simulation
- **Robot:** TurtleBot3 Waffle Pi
  - **Robot model type:** Differential drive
  - **Sensors:**
    - 2D LiDAR
    - Front-facing RGB camera
    - Default sensor parameters
  - **Navigation:** Autonomous
- **Map resolution:** 0.05m (default for `slam_toolbox`)
- **Environment:** Unknown
- **Spawn location:** Unknown

Interim Assessment

Teams can opt to undertake the team project interim assessment from the lab session in Week 8 onward. Assessments need to be booked in with the tutors at least 24h ahead of time by submission of the report and code (via Turnitin) to receive confirmation of a demonstration time slot. This assessment is a pass/fail hurdle, students and teams may make multiple attempts to pass the interim assessment until the end of semester. The interim assessment involves the following elements:

Component	Description
System Demonstration (team)	Live demonstration by the team of their robot in simulation accomplishing the first stage of the project brief during the prac session.
Oral Assessment (individual)	Students will be asked technical questions about their individual contributions, the system implementation and design choices.
Report Submission (team)	Written report describing all modules within the system and how they interact to achieve the target task. The report should clearly state which components were custom designed by the team and which were integrated from existing libraries.
Code Submission (team)	Complete git repository required to run the demo including relevant code comments and instructions for installation and operation. Provided as a link.

In the following checklists and criteria, all top level items must be checked for the team to pass.

System Demonstration Checklist

- ☐ Automatically sends waypoints to the robot
  - ○ Waypoints are automatically generated
  - ○ Waypoints are manually generated
- ☐ Uses SLAM to create a map
- ☐ Algorithm for detecting unexplored areas
  - ○ Finds unexplored areas
  - ○ Finds unexplored areas next to known and free cells
- ☐ Uses documented exploration strategy
  - ☐ Subscribes to `map`
  - ☐ Computes waypoint based on `map`
  - ☐ Computes waypoint that is known to be reachable
- ☐ Strategy works in all world files (entire space is visited)
  - ○ Some areas not detected by algorithm
  - ○ Entire space visited
  - ○ All unexplored areas detected by algorithm but some ignored according to exploration strategy
- ☐ Includes mechanism for detecting when exploration strategy fails
  - ☐ Subscribes to navigation status
  - ☐ Detects and reacts when navigation fails to find a valid path
  - ☐ Strategy implemented for not re-sending bad waypoints
- ☐ Algorithm for navigating robot to unexplored areas
  - ○ Uses `nav2` to move robot to waypoint
  - ○ Uses custom planning algorithm to move robot to waypoint
- ☐ Not hard-coded for certain test environments

Oral Assessment Criteria

- ☐ Describes a relevant and substantial technical contribution to the team project that is consistent with the overall teamwork breakdown
- ☐ Clearly explains their technical implementation
- ☐ Adequately justifies any algorithmic design choices

Report Criteria

- ☐ Includes a complete description of the overall system
- ☐ Includes a system diagram showing how all the components interact
- ☐ Includes a complete description of each of the developed components
- ☐ Clearly identifies and references any use of existing code or libraries
- ☐ Well-structured and clear

Code Criteria

- ☐ Code is available to clone from a git repository
- ☐ README.md instructions allow successful installation and running of the code
- ☐ Running the code successfully reproduces the live system demonstration
- ☐ Well-structured and clear
- ☐ Code is adequately commented