

Title: Develop a Simple Rover Simulation using Nav2 and ROS 2.0 - Humble

Objective: To demonstrate hands-on knowledge of a simple simulation of a rover using Nav2, ROS 2.0 - Humble, and Docker.

Overview: Create and demonstrate a simple rover that navigates in a sample lunar world, with a goal to reach its destination via 3 pre-defined waypoints. Use Gazebo Fortress as the simulator to demonstrate rover navigation capabilities. You may seek user inputs during script launch for any initialisation of simulation parameters. ROS nodes must run in a docker environment, and the simulation can run on the host machine's operating system if you are unable to run inside the docker container. Being able to visualise the simulation is important, set-up the package to prioritise it.

Tasks:

1. Setup a ROS 2.0 - Humble Environment with Nav2: Set up a ROS 2.0 - Humble environment using Docker. The environment should be able to run on Ubuntu 22.04.
2. Use Simulation models: Use [X1 Config 6 rover](#) and [Moon world model](#) for this task.
3. Implement Navigation using Nav2: Implement navigation for the rover using Nav2. The rover should be able to move around in the simulated environment without colliding with obstacles and complete a route consisting of 3-waypoints successfully. Note that successful navigation means the environment should start with a randomized set of obstacles, and the rover should demonstrate navigation to at least 1 waypoint with 100% success rate.
4. Develop a custom ROS node: Develop custom ROS node(s) that perform the following tasks. You are free to choose the format of the user inputs, but new custom ROS topics/services/actions must be created to perform these functions.
 - a. takes in user inputs for a goal definition and creates Nav2 goals from those inputs at the beginning of the simulated run.
 - b. Monitors for new user inputs during the simulated run for issuing an emergency stop command to the rover.
5. Develop a launch package: Develop a script that sets up the entire Docker environment, runs the unit tests, and launches the simulation. The script should also include clean-up options to remove all the files it created. We will run the scripts in a fresh Ubuntu 22.04 virtual machine, with no additional software installed.
6. Automated Tests: Setup basic automated tests for the implemented navigation system.
7. Documentation: Provide a brief documentation explaining your design choices, how to setup the environment, run the simulation, and use the launch script. It should also include any known bugs, limitations and issues that might be relevant for us to know before running the scripts.

Deliverables:

1. Docker file and associated launch script with setup, run, and clean-up options, with instructions to set up the environment
2. Link to the GitHub repository containing all the files
3. Documentation in a markdown format. Bonus points for documentation generated based on the code files.