

Autonomous Mapping and Navigation Project Documentation

Overview

This project is a ROS 2 package designed for autonomous mapping and navigation using a TurtleBot3 in a simulated or real-world environment. The project leverages the SLAM toolbox and Nav2 stack to explore and map the environment, with the option to save maps and restart exploration.

Related Pages

- **Cylinder Filter and Placement** : Documentation on the Cylinder Filter and Placement nodes.
- **Map Display Nodes** : Documentation on the map display nodes for showing maps during and after the mapping process.

The project contains two main components:

1. **Autonomous Mapping**: Handles the exploration of unknown environments, restarts exploration upon request, and saves maps for further processing.
2. **Localizer and Navigation**: Handles goal setting, localization, and navigating to goals within the mapped environment.

Subscriber Topics

The ROS package subscribes to the following topics to receive data necessary for localization and navigation:

- **/amcl_pose** : The current pose estimate from the Adaptive Monte Carlo Localization (geometry_msgs/msg/PoseWithCovarianceStamped).
- **/scam_goals** : Waypoints or goals that the TurtleBot3 should navigate towards (geometry_msgs/msg/PoseArray).

Publisher Topics

The package publishes to the following topics to control the TurtleBot3 and provide visualization data:

- **/goal_pose** : The next goal for the TurtleBot3 (geometry_msgs::msg::PoseStamped).
- **/initialpose** : The initial pose for the TurtleBot3 at the start of the localization (geometry_msgs::msg::PoseWithCovarianceStamped).

Services

The package provides the following services to control the mapping and navigation process:

- **/restart_explore_lite** : Restarts the exploration process, clearing previous exploration data and restarting from the current location (std_srvs/srv/Trigger).
- **/save_map** : Saves the current map to the specified directory (std_srvs/srv/Trigger).

Classes

The package implements the following core classes:

- **AutonomousMapping** : This class manages the entire exploration process, including restarting exploration and saving the map.
- **LocalizerAndNavigation** : This class handles goal setting, publishing initial pose, and navigating towards the goals.

Nodes

The ROS package consists of two main nodes:

1. **autonomous_mapping**: This node starts the simulation environment, runs the exploration (using explore_lite), and handles the services for restarting the exploration and saving the map.
2. **localizer_and_navigation**: This node subscribes to goals and AMCL pose, runs a state machine to navigate towards the goals, and handles goal publishing and state management.

Unit Testing

Unit tests are provided to ensure the reliability of the system under various conditions:

- **Test** map saving : Verifies that the map is saved correctly using the save_map service.
- **Test** exploration restart : Confirms that exploration can be restarted correctly using the restart_explore_lite service.
- **Test** goal navigation : Ensures that the TurtleBot3 can navigate to given goals using the state machine logic.

Each test is designed to validate the functional requirements and robustness of the system.

Compiling the Package

To compile the package, ensure your ROS2 environment is set up correctly, then navigate to your ROS2 workspace and execute the following command:

```
cd ~/ros2_ws/  
colcon build --symlink-install --packages-select autonomous_robot
```

Running the Autonomous Mapping Node

Start the autonomous mapping node:

```
ros2 run autonomous_robot autonomous_mapping
```

Running the Localizer and Navigation

Node

Start the localizer and navigation node:

```
ros2 run autonomous_robot localizer_and_navigation
```

Setting Goals

Goals can be set by publishing to the /scam_goals topic or using a text file to load predefined goals.

Executing the Mission

To start the mission and navigate towards goals:

```
ros2 run autonomous_robot localizer_and_navigation
```

This command will start the node, subscribe to the /scam_goals topic, and navigate through the goals while checking for the TurtleBot3's current pose.

Generated by [doxygen](#) 1.9.1