

Write Up

Home Service Robot

Karthik Mothiki

13th October 2021



Title: Robotics Software Engineer

Project: Home Service Robot

In this project, you will use everything you learned in the Nanodegree Program to build a Home Service Robot in ROS.

Prerequisites/Dependencies required

1. Gazebo ≥ 7.0
2. ROS Kinetic
3. ROS navigation package

Packages Used:

1. turtlebot
2. turtlebot_interactions
3. turtlebot_simulator
4. slam_gmapping
5. add_markers
6. Pick_objects

Turtlebot:

The turtlebot stack provides all the basic drivers for running and using a Turtlebot with ROS.

In this package, we used some sub-packages

- Turtlebot
- Turtlebot_bringup
- Turtlebot_capabilities
- turtlebot_teleop

Turtlebot_interactions:

This is the evolution of the Turtlebot_rviz stack supporting user side interactions with the turtlebot.

There are some sub-packages in this package:

- turtlebot_dashboard
- Turtlebot_interactions
- Turtlebot_interactive_markers
- Turtlebot_rviz_launchers

Turtlebot_simulator:

Launchers for Gazebo simulation of the turtlebot

slam_mapping:

This package is used to create slam and mapping functions. To create a functional map of the given world we use this package.


add_markers:

This package is created to present virtual objects on the map. Files in this package are:

1. **home_service.launch**: Launch rviz with specify rviz configuration file
2. **Add_markers.cpp**: C++ script, communicate with pick_objects node and control the marker appearance to simulate object pick up and drop off

pick_objects:

This package is created to pick the virtual objects on the map. Files in this package are:

- 
1. **pick_objects.cpp**: C++ script, communicate with add_markers node and command the robot to pick up the object

scripts:

Shell scripting is used to launch and run multiple files and nodes respectfully. Scripts used in this project are:

1. **add_marker.sh**: Shell script file to deploy a turtlebot inside your environment, model a virtual object with markers in rviz.
2. **home_service.sh**: Shell script file to deploy a turtlebot inside your environment, simulate a full home service robot capable of navigating to pick up and deliver virtual objects.
3. **pick_objects.sh**: Shell script file to deploy a turtlebot inside your environment, communicate with the ROS navigation stack and autonomously send successive goals for your robot to reach.
4. **test_navigation.sh**: Shell script file to deploy a turtlebot inside your environment, pick two different goals and test your robot's ability to reach them and orient itself with respect to them.
5. **test_slam.sh**: Shell script file to deploy a turtlebot inside your environment, control it with keyboard commands, interface it with a SLAM package, and visualize the map in rviz

rvizConfig:

The rviz file is stored in this folder with all the preset topics.

map:

The mapped world is saved in this folder.