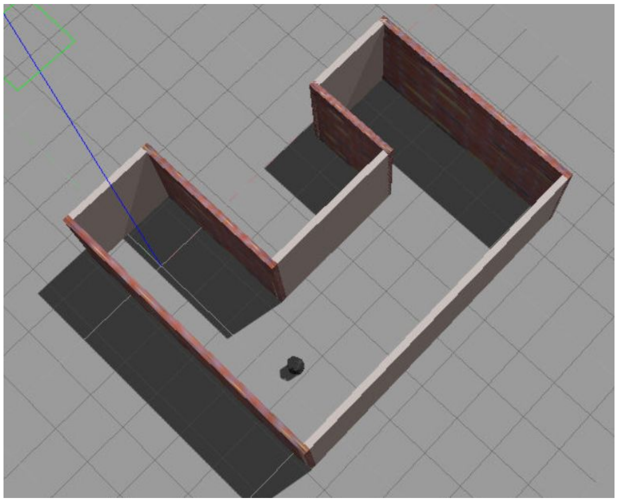
Home Service Robot

This is the final project of Robotics Software Engineer Nanodegree Program by Udacity.

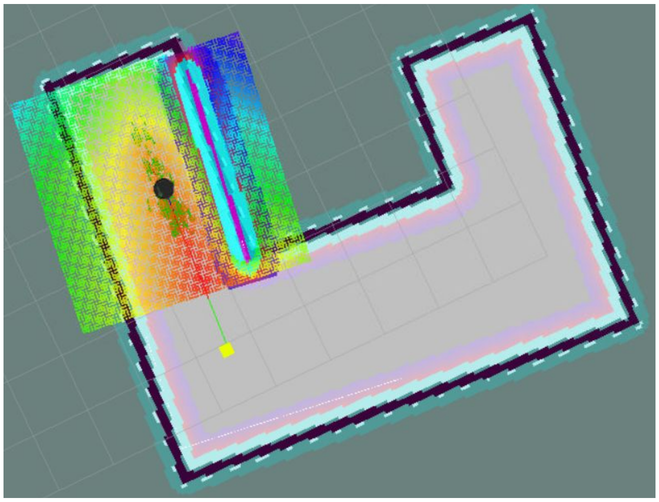
Project Description

The project attempts to combine SLAM and Navigation into a robot so that it can autonomously transport an object in a Gazebo environment [Fig.1]. By running `home_service.sh` file, the robot performs the following tasks:

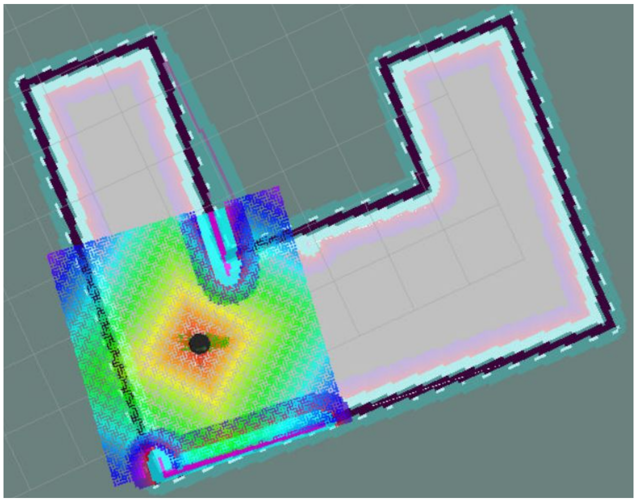
1. Navigating to a object at the pickup zone. [Fig.2]
2. Picking the object. [Fig.3]
3. Carrying the object to the drop off zone.
4. Dropping off the object to the zone. [Fig.4]



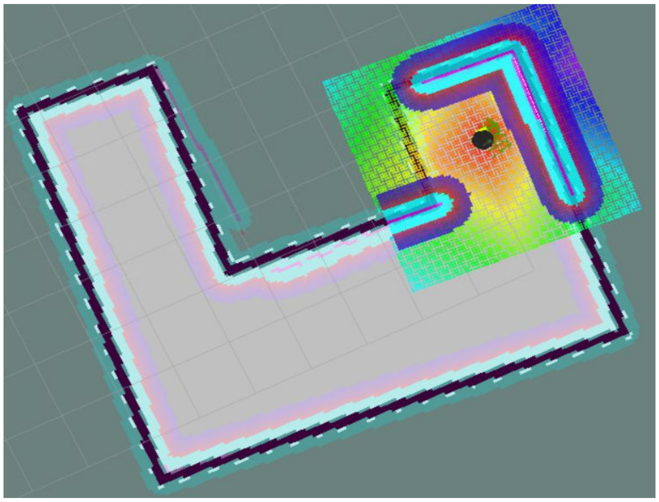
[Fig.1]: Gazebo environment and Turtlebot2



[Fig.2]: Navigation to the pickup zone



[Fig.3]: The robot picked the object.



[Fig.4]: The robot drops off the object.

Simulation Set Up

This repository consists of the following packages and directories:

1. Official ROS packages
   `gmapping` `turtlebot_teleop` `turtlebot_rviz_launchers` `turtlebot_gazebo`
2. My packages and directories
   `map`: Stores a gazebo world file and a map created by SLAM.
   `scripts`: Stores shell script (.sh) files.
   `rvizConfig`: Stores customized Rviz configuration files.
   `pick_objects`: a node that commands the robot to drive to the pickup and drop off zones.
   `add_markers`: a node that model the object with a marker in rviz.

Map Set Up

During the navigation, the robot keeps asking itself "Where am I?" and "How do I get there?". Utilizing a known map of the Gazebo simulation environment, it estimates its pose relative to the map and plans paths while avoiding obstacles in the environment.
So before the home service simulation, we need to create a map for the robot. By running `test_slam.sh` (i.e. launch `turtlebot`, `gmapping`, `rviz` and `keyboard teleop`), we can manually perform SLAM and create a map of the Gazebo environment.

Home Service Simulation

```
~ $ cd HomeServiceRobot
~/HomeServiceRobot $ catkin_make
~/HomeServiceRobot $ chmod +x ./src/scripts/home_service.sh
~/HomeServiceRobot $ ./src/scripts/home_service.sh
```

Behind the scenes of `home_service.sh`

The `home_service.sh` file launches `turtlebot`, `AMCL` and `rviz` and runs the following 2 nodes:

1. `pick_objects`
   communicates with the ROS navigation stack
   autonomously sends pickup and drop off goals for the robot.
2. `add_markers`
   publishes/hides a virtual object at the pickup and drop off zone
   subscribes to the robot's odometry and keeps track of the robot pose.

As such, in actually, the performed tasks are as follows:

1. `add_markers` node publishes a virtual object at the pickup zone.
2. `pick_objects` node commands the robot to drive the pickup zone.
3. `add_markers` node hides the marker once the robot reaches the pickup zone.
4. `add_markers` node wait 5 seconds to simulate a pickup.
5. `pick_objects` node commands the robot to drive the drop off zone.
6. `add_markers` node shows the marker at the drop off zone once the robot reaches it

During the navigation, `AMCL` localizes the robot inside the map created by SLAM and plans paths that can avoid obstacles.