

Navigation in an Indoor Scenario

Youngjae Moon, Rubin Zou

Table of Contents

1. Which task did we choose?
2. Which robot platform did we choose?
3. What did we have to do to navigate the robot?
4. What did we have to do to replace the global planner with a customized RRT-based planner?
5. How did we generate the trajectory for our chosen robot?
6. How have we completed the Navigation part of the Home Service mission?
7. Is there anything we did to improve the accuracy of the motions?
8. What were some challenges we faced and did we do anything to mitigate them?
9. Who did what for this project?
10. How to go beyond this project?

Which task did we choose?

- We have chosen “Option 2: Navigation in an Indoor Scenario”
- The objective of this project was to enable a TurtleBot3 Waffle Pi to autonomously navigate within a simulated "house" environment using the ROS Navigation Stack.
- We were aimed to replace the default planner with an RRT-based planner for improved path planning and for autonomous exploration of the environment.
- Then, we were asked to complete the navigation part of the Home Service Mission.

Which robot platform did we choose?

- We used the TurtleBot3 Waffle Pi as the robot platform.
- We used ROS Melodic for operating system.
- We used Gazebo for simulation.
- The "house" environment was launched using the turtlebot3_gazebo package.

What did we have to do to launch the simulation with the robot?

- After finishing all the set up,
 1. Source the setup.bash from the workspace
 - a. `source devel/setup.bash`
 2. We can follow the Tutorial to launch the TurtleBot 3 House map with the robot.
 - a. `export TURTLEBOT3_MODEL=waffle_pi`
 - b. `roslaunch turtlebot3_gazebo turtlebot3_house.launch`
 - c. `roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`
 3. We can launch the SLAM mode by modifying the code given from the Tutorial
 - a. `export TURTLEBOT3_MODEL=waffle_pi`
 - b. `roslaunch turtlebot3_gazebo turtlebot3_house.launch`

What did we have to do to navigate the robot?

1. Launch the gmapping package to perform SLAM:
 - a. `roslaunch turtlebot3_slam turtlebot3_slam.launch
slam_methods:=gmapping`
2. Launch the navigation stack:
 - a. `roslaunch turtlebot3_navigation turtlebot3_navigation.launch`
3. Run the modified Python node:
 - a. `roslaunch turtlebot3_slam_navigation_node
turtlebot3_slam_navigation_node.py`

What did we have to do to replace the global planner with a customized RRT-based planner?

- We have modified the `explore.py` that is located in the `nodes` folder of `Turtlebot3-Navigation-with-SLAM`.
- Then, executed the following command:
 - `chmod +x explore.py`
 - `roslaunch turtlebot_navigation_and_mapping explore.launch`

How did we generate the trajectory for our chosen robot?

- The trajectory for the TurtleBot3 Waffle Pi was generated using the ROS Navigation Stack and RRT algorithm that we have added/modified in `explore.py`.

How have we completed the Navigation part of the Home Service mission?

- What is the Home Service mission?
 - Starting the robot at a random room, asking it to navigate to another room, and it should then automatically find a round trip between these two.
- We were unable to finish this part

Is there anything we did to improve the accuracy of the motions?

- Not really except for incorporating RRT algorithm for our project.
- But there are some ways to improve the accuracy:
 - Tuning the parameters of the Navigation Stack. Examples include:
 - The inflation radius
 - Cost scaling factor
 - Planner frequencies.

What were some challenges we faced and did we do anything to mitigate them?

- While installing the necessary packages and following the Tutorial, we had a hard time figuring out what to exactly do for the project.
 - The Tutorial did not have much information so we had to utilise other sources including ChatGPT.
 - While ChatGPT was useful for us to grasp the overall picture, it was not useful in actual coding.
- Understanding the code given from various GitHub repositories.
 - Asked ChatGPT for source code summarization

What were some challenges we faced and did we do anything to mitigate them?

- Figuring out how to approach this project
 - We decided to modify the sample GitHub repository that the TA shared with us.
 - One of them contained `explore.py` and we have built on top of `explore.py`

Who did what for this project?

- In terms of division of the work, Youngjae has worked on how to set up for this project and determine the exact tasks to do for this project.
- We both worked together modifying `explore.py` but Rubin did a lot more work on this.
- Hence we decided to use Rubin's work for final submission.
- We were both stuck figuring out which files to exactly create or modify in what ways (including debugging).
- Yet it was Rubin that figured out how to resolve these problems.
- Thus, Youngjae has mostly done writing up this final project and final presentation.
- We decided to present together in class on May 2nd.

How to go beyond this project?

- To go beyond this project, we can do several things.
- First, we could perhaps further tune the navigation stack parameters or use different path finding algorithms for better performance.
- Second, we can perhaps use machine learning and the python opencv library to figure out the location of each room in a more clever manner.
- Last but not least, complete the navigation part of the Home Service Mission.

Check out our GitHub repo:

https://github.com/Pingumaniac/Nagivation_in_an_Indoor_Scenario

Thank you!

Any questions?