CS 3891/5891: Special Topics - The Algorithms of Robotics

Spring 2023

Final Project
Due depending on when you sign up for your presentation slot.

---

1 Introduction

For your final project, you will use algorithms you have implemented throughout this course on a different platform or scene. In this project, you will have a chance to dive deeper into the infrastructure of ROS, to gain more thorough understanding of the ROS tools. You will have two options: to build a RRT based path planner on another popular collaborative robot or to investigate use of the ROS navigation package.

This project should be done in pairs. You will present your implementation and turn in a short report before your presentation. Please sign up for a presentation slot here.

2 Option 1: RRT Algorithm Reproduction

We will use the Franka Emika robot, shown below in task 1. This task involves adapting the `CS3891 path planner` you have implemented in assignment 7 and 8 for the Franka. Recall that in the previous assignment, you were given the outlines of the code infrastructure in the `assignment_7_8_context.cpp` file and provided all the other details about how the robot and motion planner were constructed. You will need to use internet resources and code from the previous assignments to build the infrastructure from scratch.

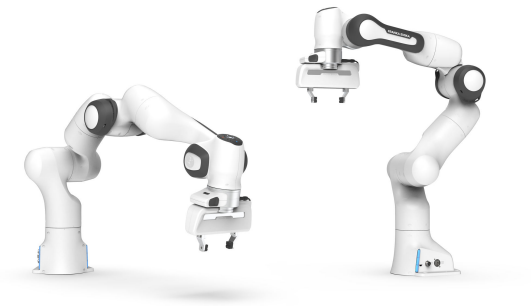

Figure 1: Two Franka Emika robots.

First, recall that in assignment 7, you tried MoveIt! by running a `demo.launch` file in `ur5_moveit_config` package, with OMPL based path planner. This package is located in `~/algorithms_of_robotics_workspace/src/ur5/universal_robot`. Now it's time to create the MoveIt! package for the Franka. Note that the first step is always to prepare the URDF files, which essentially define the physical relations among the robot joints. For your reference, the description files for Franka robot can be found from the Franka description package for Melodic. The first 5 sections in the ROS URDF tutorial describes how to use the URDF files.

Carefully compare the package folder for the `ur5/universal_robot` with the `assignment_7_8` package - you will then figure out what are the essential files to modify to replace the OMPL planner with the customized one. Then it's your turn to generate the MoveIt! package for the Franka robot. A comprehensive tutorial can be found from the MoveIt! setup assistant official webpage.

Finally, after you have the package successfully generated, and with the default motion planner replaced by the customized `CS3891 path planner`, you should draw our school logo - a capital "V" within the robot workspace. Note that in this task, you should plan with two same Franka arms, and each of the arm will draw one line of the "V". Therefore, you will either need to modify the URDF or xacro files to define the dual-arm setup, or to modify the MoveIt! package files, or to have two arms incorporated in RviZ, to achieve this. It is your job to figure out which setup is the most compatible with your customized planner. You will also need to put at least one obstacle per edge, to validate your planner finds a collision-free path around the obstacles (that's to say, the shape formed by start to goal points should be a "V", but the planned paths don't need to be). An example setup, demonstrated by another robot, is shown in Fig. 2.

Figure 2: Example setup of the V shape path planning. Two spherical obstacles are placed at each edge, so that when both arms start planning from top to down, their actual path will not form a "V". Note your robot will be different than the one shown here.

3 Option 2: Navigation in an Indoor Scenario

For this task, you should select the Turtlebot 3 as your platform. To get familiar with this robot, you should first complete the first 4 sections in the 'Simulation' tutorial, listed in this page.

Figure 3: The Turtlebot 3.

After this, for Turtlebot 3, build a map of the "house" world, as shown in Fig. 4. Next, properly label

each of the room in the world, and select a point in each room to represent it. For example, you can name the room in the middle with a table "living room" or simply "room 1", and the room in the left top corner with a bed "bed room" or simply "room 2".

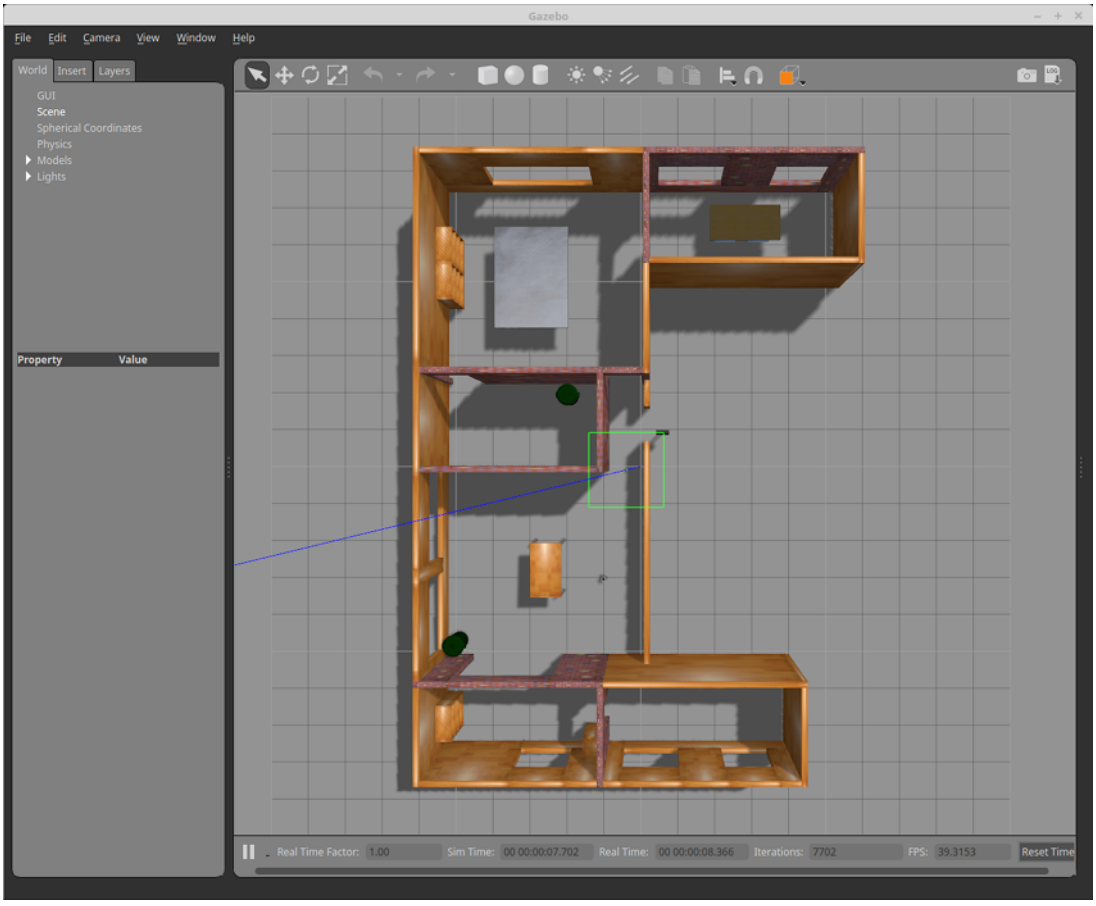

Figure 4: The "house" map, a given example world from the Turtlebot 3 tutorial.

Finally, your task is to accomplish the home service mission, illustrated in section 7.10.4.5 of this page - you will only need to achieve the navigation part of the mission. We will grade your work by 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.

4 Deliverables

You should hand in your Docker container where you implement the solutions. Make sure to include a README for how to run your code. If we cannot figure out how to run your code, you will be docked points. You will also need to turn in a short report describing your implementation on Brightspace. During the final exam period of this class, you will give a 15 min presentation of your project. For example, consider the following questions in your presentation:

1. Which task and robot platform did you choose?
2. What did you have to do to launch the simulation with that robot?
3. How did you generate the trajectory for your chosen robot?
4. Is there anything you did to improve the accuracy of the motions?
5. What were some challenges you faced and did you do anything to mitigate them?

Please have one group member submit the Docker container and the presentation to Brightspace. Everything is due to Brightspace prior to your presentation session.