67301 - MULTI ROBOT SYSTEMS- Spring 2021

Assignment 2

Sarah Keren and Mohammad Masarwy

The Taub Faculty of Computer Science
Technion - Israel Institute of Technology

Logistics

- This is the second out of the three assignments of this course.
- The due date for this assignment is 24/12/2021, 23:55.
- You can work alone, in pairs, or in groups of three (in which case you will need my approval and will have an extra assignment).

Deliverables

You are required to submit 3 documents:

- A **single** python module containing your code named **assignment2.py** (make sure it is python2 compatible).
- A **presentation** describing your selected approaches to the problems you were required to solve. The presentation will be in **Latex**[1] in a format we will provide to you.

The two files will be submitted in a single zipped folder. The name of the folder is **Assignment2-ID1-ID2**.

[1] we will have a tutorial on latex

Setup

- You can choose between two options.
  - Option #1 - a single robot in a cleaning task and an inspection task.
  - Option #2 - a single robot in a cleaning task and an ’asking for assistance’ task.

The second option allows you greater flexibility, but will require more work (and requires my approval) !

Option #1

Setup

- You will be working with the turtlebots on a vacuum cleaning task and on an inspection task.
- You can use all the packages we have seen so far, including move_base.
- We will run your solutions both in simulation and on the real robots, so we recommend that you gain some experience working with the real robots.
- You will have to fulfil each task within the specified time bound.
- Each task is described below, and will be evaluated separately.

Background: Coverage and Inspection

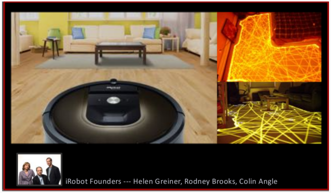
- There are many real life applications that require a robot, or a group of robots, to cover an area (i.e., to visit the entire area).
- Examples: a vacuum cleaning robot that needs to clean an entire room, intrusion detection, mine cleaning and search-and-rescue missions.

You will implement 2 different coverage missions for a single agent.

Spoiler: the final project will be the same (or similar) tasks, but with multiple robots.

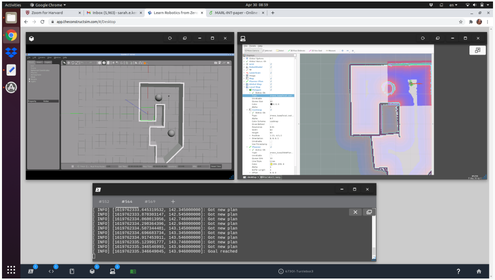
Task 1: cleaning a fully-mapped environment

- Your robot will be placed in a room that is fully mapped (you will only receive the map at execution time).
- Apart from the walls, there will be no other objects in the room.
- You will be allowed to localize your robot, using the relevant ROS command.
- We will place ’pieces of dirt’ in the room. In the gazebo we will mark them on the floor. In the real world test, a piece of dirt will be marked by an ’X’ on the floor (which the robot can’t sense).
- You will have a fixed amount of time (4 minutes) to collect as many pieces of dirt from the room as you can.


iRobot Founders --- Helen Greiner, Rodney Brooks, Colin Angle

Task 2: inspection task

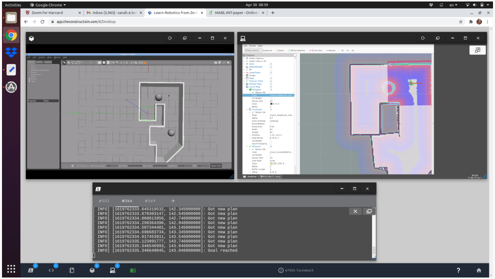
- Your robot will be placed in a room similar to the one described above. The difference is that in this setting, there are unmapped obstacles (spheres) that are distributed in the environment.
- Your algorithm will report the number of objects that are estimated to exist in the environment.
- You will have an unknown amount of time to detect the number of spheres in the room (the program will end unexpectedly at some time bound we will set).



```
[ INFO] [1619762333.645319532, 142.345000000]: Got new plan
[ INFO] [1619762333.870303147, 142.545000000]: Got new plan
[ INFO] [1619762334.060013956, 142.746000000]: Got new plan
[ INFO] [1619762334.298364396, 142.948000000]: Got new plan
[ INFO] [1619762334.507344481, 143.145000000]: Got new plan
[ INFO] [1619762334.696683734, 143.345000000]: Got new plan
[ INFO] [1619762334.917453911, 143.546000000]: Got new plan
[ INFO] [1619762335.123991777, 143.746000000]: Got new plan
[ INFO] [1619762335.346546993, 143.946000000]: Got new plan
[ INFO] [1619762335.346649045, 143.946000000]: Goal reached
```

Task 2: inspection task - continued

- Since you don’t know how much time you have for the task, you need to publish every 30 seconds a message with the following format:
    - topic: inspection_report
    - structure: a string with the format: ”X spheres detected at time TIMESTAMP”
    - We will check your answer at 3 time intervals (the last one will be of at least 4 minutes).



```
[ INFO] [1619762333.645319532, 142.345000000]: Got new plan
[ INFO] [1619762333.870303147, 142.545000000]: Got new plan
[ INFO] [1619762334.060013956, 142.746000000]: Got new plan
[ INFO] [1619762334.298364396, 142.948000000]: Got new plan
[ INFO] [1619762334.507344481, 143.145000000]: Got new plan
[ INFO] [1619762334.696683734, 143.345000000]: Got new plan
[ INFO] [1619762334.917453911, 143.546000000]: Got new plan
[ INFO] [1619762335.123991777, 143.746000000]: Got new plan
[ INFO] [1619762335.346546993, 143.946000000]: Got new plan
[ INFO] [1619762335.346649045, 143.946000000]: Goal reached
```

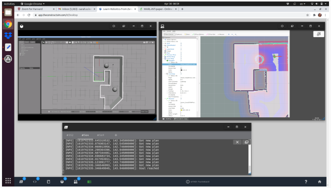
Possible extension for groups of 3

- The task is similar to task 2, but the objects in the room will have different colors (red or blue).
- In addition to the requirements of task 2, you will have to report the number of blue and red objects.
- Message structure: a string with the format: ”X red spheres and Y blue spheres detected at time TIMESTAMP”

Technical details

- Our repo *https://github.com/sarah-keren/MRS_236609* contains:
  - A python module named assignment2.py (located in the scripts folder) which is the file we will be using to run your code and the **ONLY** file we will be using.

Don’t forget to get the latest version of the repo.



```
[ INFO] [1619762333.645319522, 142.345000000]: Got new plan
[ INFO] [1619762333.870303147, 142.545000000]: Got new plan
[ INFO] [1619762334.080813956, 142.746000000]: Got new plan
[ INFO] [1619762334.298384396, 142.948000000]: Got new plan
[ INFO] [1619762334.507344481, 143.145000000]: Got new plan
[ INFO] [1619762334.696683734, 143.345000000]: Got new plan
[ INFO] [1619762334.917453911, 143.546000000]: Got new plan
[ INFO] [1619762335.123991777, 143.746000000]: Got new plan
[ INFO] [1619762335.346546093, 143.946000000]: Got new plan
[ INFO] [1619762335.346649045, 143.946000000]: Goal reached
```

Technical details (continued)

- This is a video that shows how to add obstacles to Gazebo: *`https://youtu.be/RJmVwhzMpZg`*
- You can save the world file using the 'save' command of the Gazebo.
- The repo also contains a world file with a room with 2 speheres: worlds/closed_room_with_2_spheres.world
- Note, the spheres are not in the map !

Pay attention

- Do not ! include ANY package that is not already installed by default. If you want a package that is not installed - you need our approval.
- You need to make sure your code is python2 compatible.

Option #2

Setup

- You will complete Task 1 as described above with the Turtlebot.
- Instead of the inspection task, you will complete a ’request for help’ task.

An Autonomous Agent Requesting Help

- **Main agenda**: Your robot will need to understand its own limitations with regards to its ability to accomplish a task.
- You can choose which robot you want to work with.
- You will need to account for some form of uncertainty the robot needs to ’deal with’: localization uncertainty, navigation cost estimation, uncertainty regarding grip poses, etc.
- You will need to **randomly** various possible configurations, in which the robot will assess its capabilities and request for help.

An Autonomous Agent Requesting Help

- Note: This is a task that requires my approval.
- You will only be allowed to choose this option if you have a rich and interesting representation of uncertainty and an innovative way to quantify it.

Presentation

Working with Latex

- The Latex template you need to use for your presentation can be found here: *https://www.overleaf.com/read/qqcgbpqjxrfq*
- We will have a tutorial on Latex and will provide technical support on moodle.

Presentation Structure

- It requires you to specify for each task:
- **Problem Description** - an informal description of the problem you are solving.
- **Model** - a formal description of the problem.
- A description of your **Suggested Approach**.
- Key **implementation details**.
- The **performance guarantees** you approach offers. No need for formal proofs, only an intuitive explanation.
- **Details from the process**: fun facts from the process, such as failed approaches etc.

Important Notes

Note!

This is probably not the final description of the assignment.

As you start working on it, you will have questions, and I will updated the description accordingly.

Good luck and don’t forget to enjoy the process.