Homework 1

DEADLINE: 23th December

Implement a routine that lets Tiago navigate inside the environment similar to Figure 1. The environment is composed of two rooms with non-static obstacles (i.e. you can move them in Gazebo) and a narrow space. Tiago has to navigate from point A to point B.

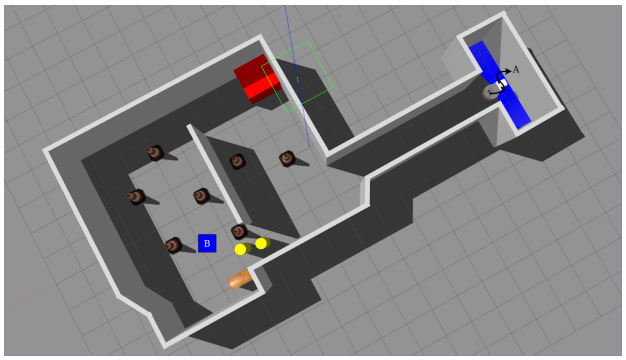

Figure 1

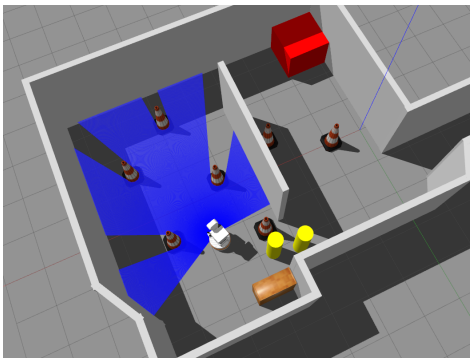

Figure 2

The user can input the position of Point B by command line. In B the robot has to recognize, and output to the screen, the position of the non-static obstacles, e.g. the cones, that are in front of it (Figure 2). The static obstacles that are part of the map, e.g. the walls, haven't to be recognized. **Suggestion:** use the laserscan (topic /scan).

Your code must be implemented using the following structure:

- The user can input the position of point B by command line;
- Your code must be implemented with an Action Client/Server infrastructure:
    - The action client receives the input from the user;
    - The action client calls the action server that executes all the task;
    - The action server sends the final list of positions of the obstacles as results to the action client;
    - **Optional:** the action client receives feedback from the server and outputs the current status of the task (e.g. the robot is moving, the robot reached point B, etc.).

Extra points

Typically in narrow passages the navigation stack is inefficient. Therefore, the sensors of the robot are used to be able to navigate, e.g. laser and/or sonar. In this homework, to get extra points, we ask you to implement your own routine in order to navigate through the narrow passage using the laser.
In this case, you have to specify in the email and in the report that you also implemented this solution.

Instruction to start the homework

Follow these instructions to install the environment and start the homework:

- Clone your repository into your workspace, it will initially contain all the environment and launch files necessary to start the development. You can either use the package inside the repository for developing your solution into it, or create your own package and deliver this one (**NB.** remember to add your additional package into the same repository when delivering the solution):
  ```
  $> git clone
  https://username@bitbucket.org/iaslab-unipd/intelligentroboti
  cs_group_XX.git
  ```
- Start the simulation (cmd console 1):
  Simulation with empty world
  ```
  $> roslaunch tiago_iaslab_simulation start_simulation.launch
  world_name:=ias_lab_room_empty
  ```
  Simulation with all the obstacles
  ```
  $> roslaunch tiago_iaslab_simulation start_simulation.launch
  world_name:=ias_lab_room_full
  ```
- Navigation stack (cmd console 2):
  ```
  $> roslaunch tiago_iaslab_simulation navigation.launch
  ```

Instruction to submit your solution

To submit your solution you **MUST** do the following:

- Push your code into your group bitbucket repository;
- Prepare a small pdf report (max. 2 pages) that explains your solution;
- The report, or the readme file in the repository, **must** contain the necessary commands to correctly execute and test your code (e.g. roslaunch and rosrun commands);
- Prepare a short video (e.g. screen record) that shows the execution of your homework. This is necessary in case we cannot easily reproduce your code;
- Send us an email that confirms the submission and attach all the necessary information (e.g. report, link to the repo, additional info, etc.).
- Optional: if you use Docker you can also submit the docker image. Thus, you have to give us the name of the repository and tag or the link of your repository in Docker Hub.