ELEE 4200/5200: Autonomous Mobility Robotics
Term I, 2018
Homework 8: Find the Door

Guidelines:

- Due date: Tuesday, November 20, 2018 by 12 Noon.
- Each group of no more than two students must work on its own in completing this assignment! Feel free to consult with others (and with me and the TAs) in developing solution ideas, but the final implemented code must be your work product alone. Refer to the Syllabus, where the policy on academic integrity is clearly outlined, our classroom discussion on this topic, and consult with me if you have any questions!
- State the full names and T# of the students in the group on the cover page of every document that you submit.
- Submit the report by responding to this assignment posting in Blackboard.
- The submission should at least include the following documents, bundled together into a single zip file with the name *YourNameHW8* (use one of the group member names).
    - The main report (following the template provided).
    - The main report in 'pdf' form.
    - The MATLAB program code.
- A hard copy (printout) of the 'pdf' report with MATLAB code; staple all pages together and follow the TA's instructions on how to submit.

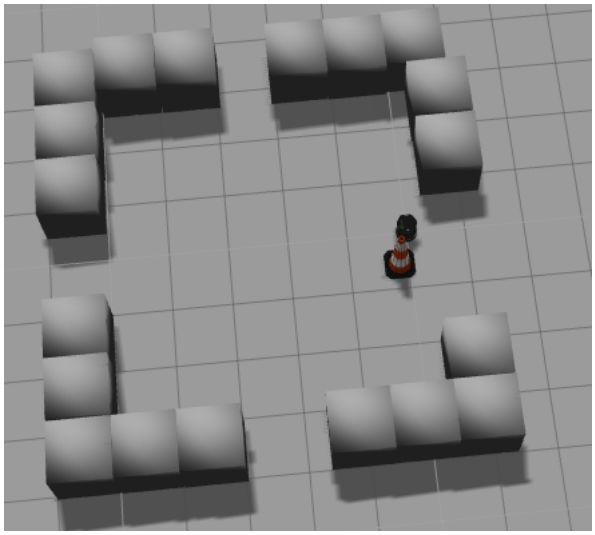

Figure 1

Broad Goals:

a) To construct an algorithm that requires the use of cognition techniques.
b) Specifically, to construct a program in MATLAB that drives a simulated Turtlebot (in Gazebo) out from an enclosed area through the widest opening (see Figure 1).
c) To accomplish this in two ways – with and without limited vision – by using the Kinect Lidar and the Hokuyo Lidar respectively.

Specific Tasks:

- Design an algorithm that enables the Turtlebot to find its way out of the enclosed area through the widest opening.
- Do this first using the Kinect Lidar and then the Hokuyo Lidar to understand and be able to compensate for the effects of limited vision.
- Use the given Gazebo environment ("new_find_door.world") specifically uploaded to Blackboard for this assignment. Instructions were provided in an earlier assignment on how to incorporate the specified Gazebo world.
- **The presence of the obstacle (traffic cone) should not be considered as changing the width of the door opening!**
- Work with the given initial robot position. This position can be set by using the following command:

  rosservice call /gazebo/set_model_state '{model_state: { model_name: mobile_base, pose: { position: { x: 2.4683, y: -4.5436 ,z: 0 }, orientation: {x: -0.0029, y: -0.0028, z: -0.7175, w: 0.6965} }, twist: { linear: {x: 0 , y: 0 ,z: 0 } , angular: { x: 0.0 , y: 0 , z: 0 } } , reference_frame: world } }'
- For any robot run that you execute, you must provide a plot of the robot path in your report (based on "model state") and other plots as you see fit. Also, include (meaningful) videos as you think are appropriate and support your descriptions in the report.
- Choosing an initial pose for the robot that points it at the widest opening is not acceptable. Work with the given initial robot position. Also, in general, the algorithm should work for any initial pose and environment. But, if your algorithm is restricted by some assumptions that you made, clearly state them in your report, as limitations of the approach. However, you can assume that the enclosure is approximately rectangular and that the doors are roughly centered in each of the four walls. Also remember, the perceived width of the opening will vary based on the viewing angle.
- The robot is permitted to stop and look around (pivot), as part of a strategy to determine where the widest opening is.

Other Related Information:

- In addition to new strategies, this assignment requires elements implemented in some of the earlier assignments – drive to a goal and obstacle avoidance (while driving through the chosen door), perhaps even wall-following.
- Any strategy is fine. However, if you adopt a strategy based on the concept of mapping, you need to construct the map yourself and not use a pre-constructed mapping algorithm in MATLAB/ROS.
- Explain your algorithm clearly in your report! Just a reminder that it is not just the final results that are of interest here. You are also responsible for explaining how you arrived at your program and the attendant robot behavior. As I have said before, *the process is more important than seemingly correct results.* This is embodied in the following sequence – an algorithmic idea translated correctly to code, examination of the attendant behavior of the robot in multiple experiments, modification of the algorithm to change behavior desirably, etc.