Reactive navigation and path planning

Exercise 1 :

In Matlab, experiment with reactive navigation such as Bug2 algorithm and other planning algorithms.

Try to choose a real setup to model – use e.g. the "makemap"-function in CORKE toolbox to build an occupancy grid of your setup (the setup could e.g. be a wall + doorway in Shannon building, letting the robot move from one side of the wall to the other). Alternatively, use the BinaryOccupancyGrid -function in the robotic systems toolbox, https://se.mathworks.com/help/robotics/ug/occupancy-grids.html, to build an occupancy grid.

First, try to simulate the scenario in Matlab using the Bug2 algorithm. (You can also test other planning algorithms).

Afterwards, try to implement the Bug2 algorithm on your Turtlebot + Gazebo. If time allows, try to place "obstacles" in the path of the robot – to test how it can perform obstacle avoidance.

NOTE: Hints for the solution is given in excerpt "Bug2_example_Siegwart.pdf" from R. Siegwart, 2011.

Exercise 2 (MANDATORY) :

In Matlab, experiment with path planning – try with Probabilistic Roadmap (PRM) (https://se.mathworks.com/help/robotics/ug/probabilistic-roadmaps-prm.html). Start by following and understanding the example given in the link – then modify afterwards for your needs.

Next, you have to try to use it in combination with Gazebo to move Turtlebot around. NOTE: We still do not use the sensors, so it is simply a visualisation of the robot.

Use the "Building Editor" under menu item "Gazebo" on BB to build a map similar to the Matlab example above – If there are any problems with this, it is also o.k. to just use a "clean world" without any walls.. Then use PRM method to create a path between to points (like in the Matlab example). Then use Pure Pursuit to move the robot in both Matlab and in Gazebo.

In the report, you simply have to write what you did (which world/map used – with image/screenshot, settings of PRM, problems encountered, ..) and possible solutions to problems.