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. 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: Probabilistic roadmap algorithm (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.

After understanding the PRM algorithm, you have to implement and test it on the real Turtlebot.

First, you have to choose a place for you test (in Shannon building) and build a map of it. It can just be a couple of walls or a rack or doorway – just for you to make your test. You can use e.g. https://se.mathworks.com/help/nav/ref/binaryoccupancymap.html.

The steps will be

1) Use PRM to find a path from point A to B. Choose e.g. a distance of 5-10 meters where the robot has to turn some corners/walls (just to make it a little challenging)
2) Use Pure Pursuit algorithm from last week to follow the path.
3) Measure the accuracy. How close do you come to point B ? both in reality with the robot and also in your matlab model (simulation)

Experiment with the different parameters of the model to achieve robust behavior.