Homework 2: Differential Drive Navigation

i. Test different PID settings and thresholds for the PID navigation algorithm. Describe the interactions between the linear and angular PID coefficients and the distance-to-goal threshold.

We have tested different PID setting and threshold for the PID navigation algorithm. We have seen that PID coefficients act a proportional gain to the linear and angular velocities. We have the following conclusions:

- The higher the PID coefficients are, the higher the velocities are.
- If the PID coefficients are too high the system does more oscillations to reach the goal. If the PID coefficients are lower the system does less oscillations.
- If the PID coefficients are too low the system has a slower and smoother convergence.
- The distance-to-goal threshold is just an arbitrary number to consider a position near the goal as the correct goal.
- If the PID coefficients are too high but the distance-to-goal threshold is higher, it will cause the system to oscillate less times when oscillations get small.
- If the PID coefficients are too high and the distance-to-goal threshold is lower, it will cause the system to oscillate more times until it reaches the goal.

ii. Test the obstacles avoidance algorithm in simulation and with the real robot. How does it behave? Does it avoid obstacles and reach the goal? Why or why not?

Using this bug algorithm, the simulated robot touches the obstacle with the right side of the robot because we're not taking into account the robot's width and height. With the real robot is happening the same, it avoids the obstacles but it touches them. And that's definitely not good.

iii. Think of a way to improve the obstacle avoidance and implement and test it. Describe your proposed improvement and whether it succeeded.

As we have said in simulation and with the real robot, even it avoids the obstacles correctly, it touches the obstacles with the right side of the robot. So we have thought to add the size of the robot when we check if a point in the sonar is touching something. So we created a new grid with dilated obstacles and we check the rangefinder with that new grid. This will work in simulation but not with the real robot as we do not do any check with the sonar values from the robot.

Another feature we have added is that the user can't assign the goal location inside the obstacle as it as no sense.