1. 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.
   1. Increased velocity horribly messes up collision detection – rate of testing cannot keep up with excessive speeds.
   2. Similarly, keeping high linear and angular velocity and changing the DtG threshold either causes it to overshoot the goal completely if the threshold is small or to not stop at the appropriate point if the threshold is large.
   3. The error for over/undershooting becomes far smaller at lower linear velocity rates.
   4. A high linear velocity combined with an extremely small angular velocity creates extreme problems, as the linear velocity remains constantly high while the robot attempts to slowly turn, causing the robot to move extremely far from the goal point.
2. Test the obstacle avoidance algorithm in simulation and with the real robot (if working). How does it behave? Does it avoid obstacles and reach the goal? Why or why not?
   1. The bug algorithm works reasonably well in most situations. It avoids the square obstacle if it lies between the robot and the goal point. Since the bug algorithm in general will always find its goal (excepting the inclusion of obstacles that completely encompass the robot) However, in specific cases, the robot will become unable to move if the goal point lies around a corner of the obstacle and also exists close to a side of the obstacle. In this scenario, the robot will approach the corner of the obstacle and stop, oscillating between two sides of an extremely tight angle until the user provides a new goal location.
3. Think of a way to improve the obstacle avoidance algorithm and implement and test it. Describe your proposed improvement and whether it succeeded.
   1. Pre-implementation
      1. The primary improvement aims to add a “sweep” behavior to allow the bug algorithm to pick the most optimal direction when presented with an obstacle that possesses angles.
   2. Post-implementation
      1. The improvement was a basic “three point” sweep, allowing the bug algorithm to better choose a direction based on how the edges or angles of the obstacle's hull changed. Unfortunately, the three points taken cause an oscillation in the robot much like the error that occurs around corners. It's possible that the updates fail to provide the required accuracy, that the code itself fails to move the robot appropriately, or that the tests near the end of the sweep routine do not use the correct criteria to choose the robot's turn direction. Possible improvements for this method may include additional point samples in the sweep (which may also allow for regular movement from the robot) or the ability to move the rangefinder itself rather than requiring the entire robot to move.