Andrew Tarnoff

Mobile Robot Programming Problem Set #3

CODE LINK:
https://github.com/art81/EECS373/tree/master/MobileRobotics/mobile_PS3_wall_follow/src
Then click on "wall_follower.cpp"

Theory of Operation:

       The main two improvements that I made to the program were adding in a "ping tolerance" for making sure the lowest distance ping was directly on the left and also adding in a feature where after the robot finds the wall (from circular trajectory) it will make itself parallel with the wall and move forward a safe amount (determined by ping in front used to make the robot parallel).

       The first change that I made was so that the robot makes sure that the shortest ping is directly to the left but plus or minus pingTolerance pings away instead of just directly on the left. This makes it so that spinning clockwise doesn't happen as much and it isn't as important because my second change lines the robot up with the wall anyways. I also made a similar change where when spinning in a circle to re-find the wall, I only do that until any ping is smaller than the WALL_FOLLOW_RADIUS because my next change will make the robot parallel with the wall anyways.

       The second change that I made was to make the robot face parallel to the wall on its left. In order to do this I found an average of pings' corresponding x's and y's directly to the left and pi/4 radians ahead from directly left. The function that does this is "determineXDiffOfWallOnLeft" and it will return the xDiff between these two averages (after calculating them). The robot is then turned counterclockwise until the xDiff is sufficiently small. When this xDiff is sufficiently small, we know that the robot must be close to parallel with the wall because the perpendicular distance from the robot to the wall is the same directly to the left and pi/4 radians ahead. After this, the robot takes the y value associated with the pings pi/4 radians ahead and moved forward that much because we know that we are at least close to parallel with the wall up until that point.