Tyler Anderson

EECS 476

GitHub: <https://github.com/TAdragon1/mobile_robotics_assignments/tree/master/ps4_twa16/traj_builder>

Lidar: <https://github.com/TAdragon1/mobile_robotics_assignments/tree/master/ps2_twa16/ps2>

The first step for build the braking trajectory is clearing the vector of states, so the only states that will be in the vector at end of the building function will be the states that bring the robot to a stop. When building the vector, I assume the robot is moving at max speed and max rotation. I am assuming the robot can move in both the x and y direction without rotating.

Using the start position, I calculate the following states using the maximum deceleration for both linear and angular velocity with 0.2s intervals. The linear velocity slows down much faster than the angular velocity, and that can be seen in my video. The linear velocity takes 2s to get from max speed to 0, and the angular velocity takes 5s to get from max to 0. 5s at 0.2s intervals gives us a final vector of states with 100 states taking the robot to the fastest stop possible without knowing the current velocity. Improvement for this function would be passing in current velocity so we know which directions robot is moving in, say rotating +z but moving backwards, then we could build a braking trajectory that covers all cases.

When editing my lidar program, I just added a call to the estop service if the lidar alarm was triggered. An improvement on this could be to call the clear estop service once the obstacle is removed. I did not implement that for this as I feel it would be safer to manually determine that it is safe for the robot to continue, and manually call the clear estop service.