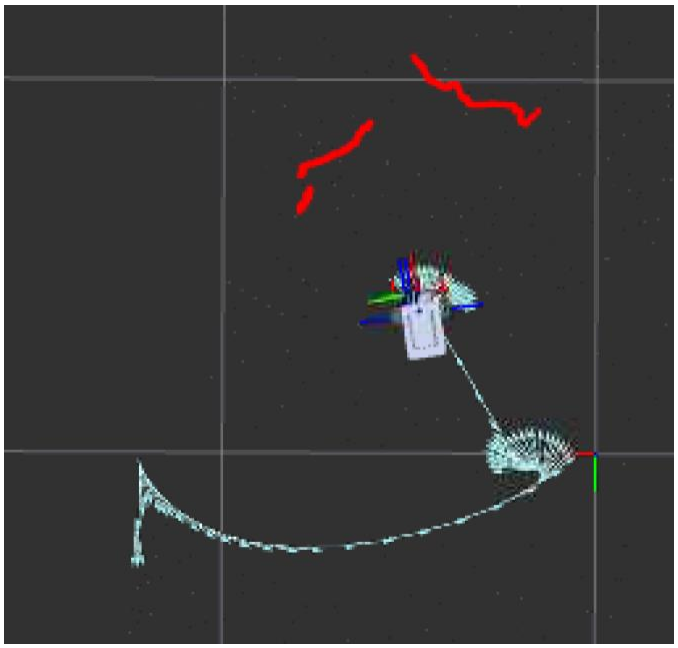
Week 6: Localization

5/6/22

ME/EE/CS 169

Tyler Nguyen and Lorenzo Shaikewitz

1 Make sure (Fake) Laserscanner is Happy

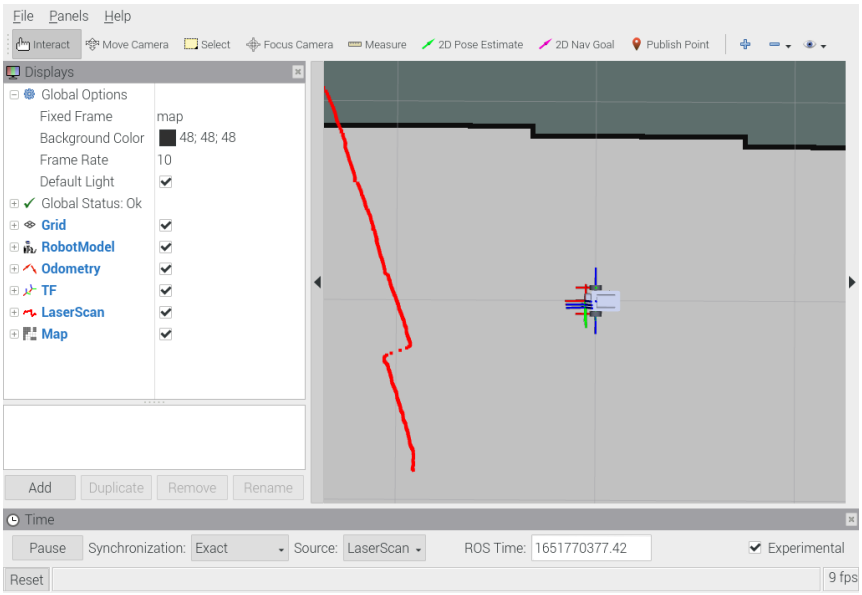


2 Stop if Obstacle Appears

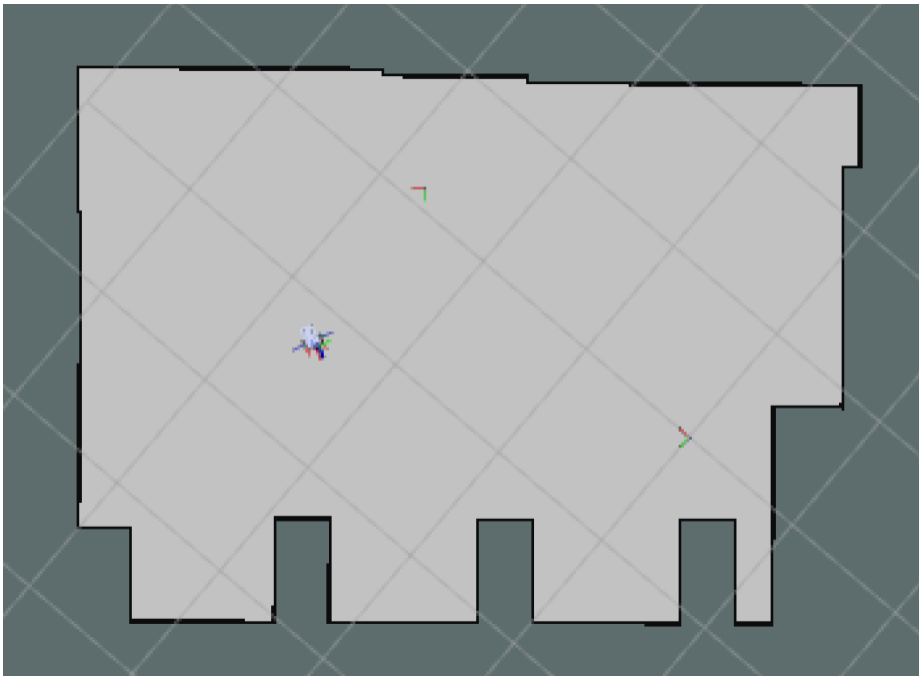
We added a callback to the laser scan messages in our driver code, for which when it detects a range smaller than our ESTOP LIMIT, we force the robot to halt. It works alright!

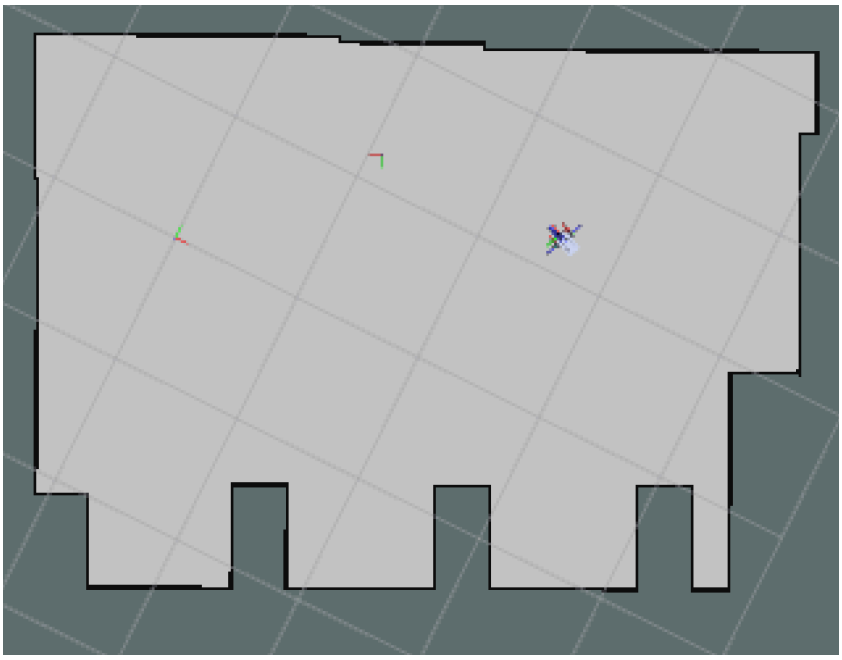
3 Add the map

Map has been added!



4 Implement a Default Localization Node

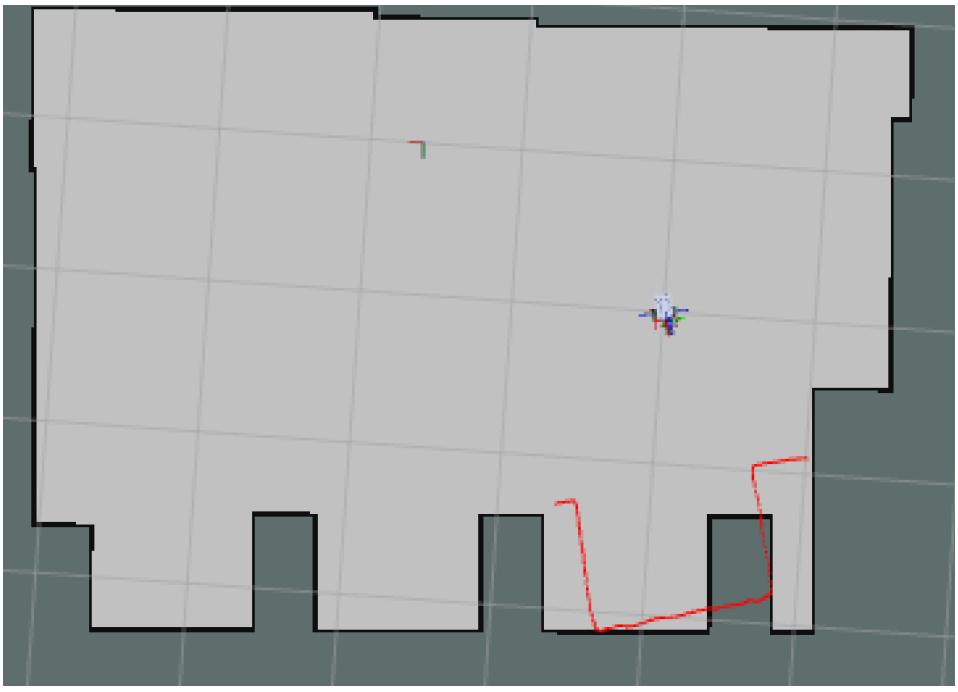


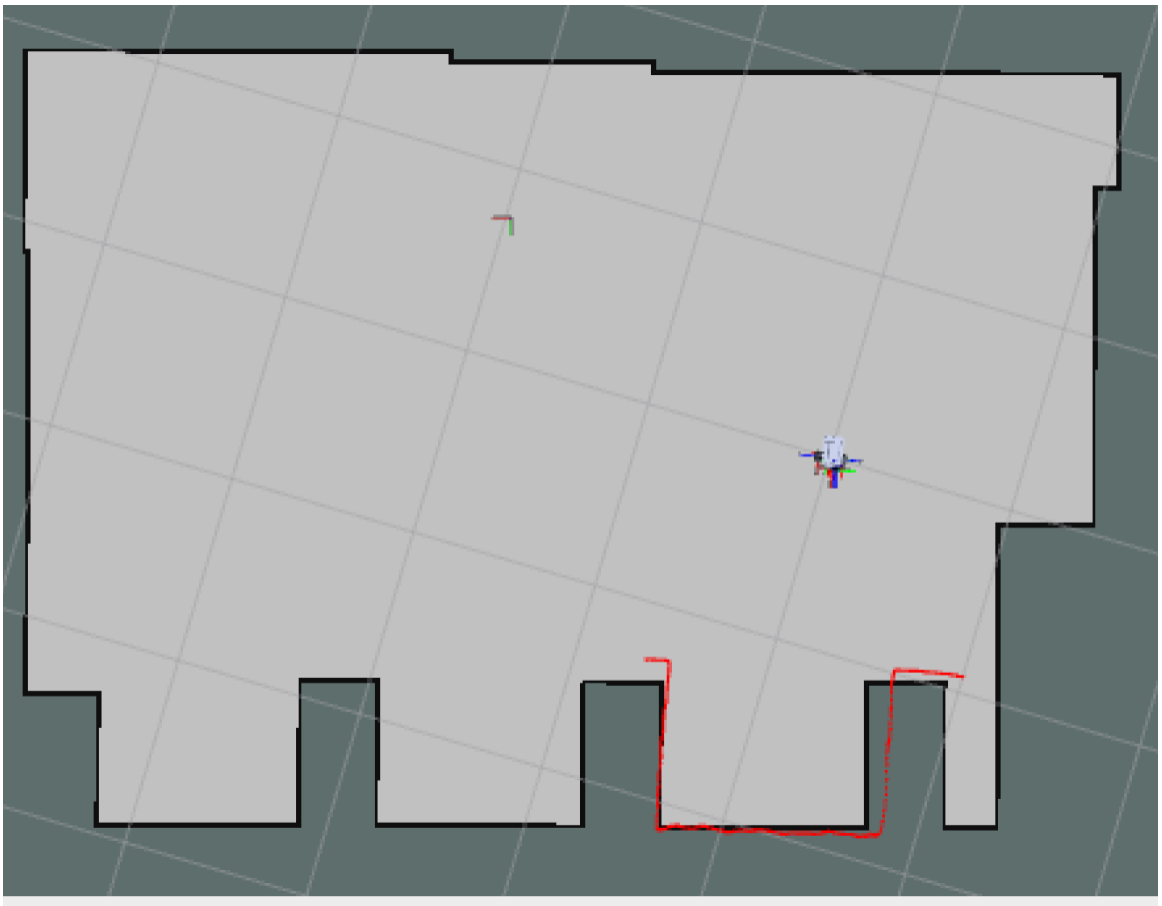


With a default localization node, we can now use the 2D Pose Estimate utility with RVIZ to change the location of the odom frame with respect to the map frame. In these figures, we use 2D pose estimate to "move" the location of the robot in the map without actually driving.

6 Implement the Continual Localization

To implement continual localization, we initially preprocess the map into a grid of the nearest walls, using numpy's argmin function (We found that this was very . We then utilize a point-to-line weighted least squares, weighted by 1 over the distance.





With a fractional update of 1/25, we find that the localization accounts for drift, while not being too jittery. However, when moving the fractional update is too large. We suspect that we will see improvements if we have a lower fractional update for moving, and higher when stationary.