

Andrew Tarnoff

Modern Robot Programming Problem Set #8

CODE LINK:

https://github.com/art81/EECS373/tree/master/problem_set_8_9/shipment_filler/src

Then click on “fill_order.cpp”

Approach:

My approach to tackling this problem set was to first make all of the arrays describing the environment into global variables (desired_models_wrt_world for example) so that I could make helper functions to do all the work and I didn't have to pass a bunch of variables each time. The first helper function that I decided to make was “moveBoxToStationConvertOrderToWorldAndReturnPose” which moved the box to a certain station (Q1, Q2, or shipping depot) and also updated the pose of the box with respect to the world if it was being moved to either Q1 or Q2. The next ones that I decided to create were “removeBadPartsFromBox”, “putMissingPartsIntoBox”, “adjustMisplacedPartsInBox”, and “fixBoxAtStation”. The first three are all self explanatory based on the name and take in a station identifier so that they can be used generally at quality station 1 or 2. The last one that I mentioned, fixBoxAtStation, uses all of the helper functions I described to take in a station number and completely fulfill the order to the best of the robots ability at that station. This makes it so that all the main method has to do is move the box to quality station 1, call “fixBoxAtStation” with the identifier meaning station one, then move the box to quality station 2, call “fixBoxAtStation” again but for station two, and then move the box to the shipping depot and call the drone. This setup makes it so that if any changes need to be made to the helper functions, the main method does not need to be changed and changes only need to be made in one spot which makes making changes very easy.

Results/Observations:

My results for this problem set are very good in that I got the simulation to run through and consistently get a score of 9. The only setbacks that I had were that it takes a very long time for the simulation to run. This is because for some reason my simulation runs slower than I expect, this is talking about simple motions from the robot and conveyor belt speed. The other thing that I noticed is that when I tell the robot to pick something up and place it at a new spot it doesn't always get it right and it has to go back to home and then try to re-adjust the same part. I don't exactly know why it would work the second time if it didn't work the first time but I guess that is something to explore more so that it just works the first time. The last major observation that I had about the problem set is that when you send the box to the shipping depot you actually move the next box passed inspection station 1. I was wondering how this issue was handled last year and my idea was that you have the robot pick the box off of the conveyor belt and then place it at the desired inspection station instead of using the conveyor belt.