

Andrew Tarnoff

### **Modern Robot Programming Problem Set #8**

**CODE LINK:** [https://github.com/art81/EECS373/tree/master/problem\\_set\\_8/shipment\\_filler/src](https://github.com/art81/EECS373/tree/master/problem_set_8/shipment_filler/src)

Then click on “unload\_box.cpp”

#### **Approach:**

The first thing that I did in order to get a better grasp of how to approach this problem is read through the box\_inspector code. The important information that I got from here is that the “get\_bad\_part\_Q1” function returns the part of the bad part in reference to the world coordinates. From there I realized that all I had to do was find which model inside of the “orphan\_models\_wrt\_world” variable was closest (euclidean distance) to this bad part (transformed the part object into a 4x4 transform in case I was going to have to change what frame it was in reference to) and then remove that part from the box using “pick\_part\_from\_box.” After this, the next task was to remove the rest of the parts from the box. In order to do this I needed to utilize the “update\_inspection” function inside of “boxInspector” to update the “orphan\_models\_wrt\_world” variable after removing the first element from that array. Continue to do this until the array is empty and by then all of the objects in the box will have been removed.

#### **Results/Observations:**

My results show that the coordinates given by the quality camera and those held in the “orphan\_models\_wrt\_world” array differ only by a very small amount and when printing out the euclidean distances between all of them and the bad part coordinates, the minimum distance found was 0.002 which is very small and shows that the two poses are very accurate in reference to each other. Another observation that I made is that the robot moves really slow and I wish that I could send speed/trajectory duration to the “pick\_part\_from\_box” function as a parameter so that I could make this faster.