

## PS9: ARIAC boxfiller

This is a longer problem set, constituting your final project (but distinct from the individual projects in EECS 473). This problem set extends the previous work in unloading a box of parts. Your objective is to get a Shipment Completion Score of 9 points. This requires that: all parts from an order are present; all parts have passed both quality inspection stations; all parts are placed precisely where desired (within the competition specifications); and a drone has picked up the shipment with the correct shipment label. After calling the drone, look at the window from which you launched ARIAC to see it display your score. Take a screenshot of this result to submit as proof of your success.

When launching ARIAC with the options below, you will start with a box that has 3 parts (piston rod and two gears). However, you'll find that the parts are not placed sufficiently precisely, and that the quality inspector detects a flawed part. After making corrections at quality station 1, advance the box to quality inspection station 2, where you may find that the second quality sensors detects another faulty part. Update the box to make the order perfect.

In doing so, you will need to update the "box\_inspector" library, since the functions for box inspection at station Q2 do not yet exist. (You can emulate the Q1 inspection code to extend this library). You will need to extend your box\_unloader node to do part-location adjustments and to acquire new parts from inventory (for which you will need to use the BinInventory library). You will need to invoke additional robot behavior functions (to fetch a part from inventory and place it in the box). Note that the gripper may fail during manipulation attempts.

Your node should start the competition, and it should subscribe to incoming orders. The order name will need to be used when you call the drone to pick up your completed shipment.

To run the simulation, do:

In your `ros_ws/src` directory, get our team code with:  
[https://github.com/cwru-robotics/cwru\\_ariac\\_2018.git](https://github.com/cwru-robotics/cwru_ariac_2018.git)

Start up Gazebo simulating ARIAC in a specific test mode with:  
`roslaunch cwru_ariac_launch sample_environment.launch fill_demo_shipment:=true`

Open three terminals and start the following nodes:

Start up robot move action server:  
`roslaunch kuka_move_as kuka_behavior_as2`

Start up the conveyor action server:  
`roslaunch conveyor_as conveyor_as`

Start up your new node (presumably based on your prior box\_unloader):  
`roslaunch shipment_filler your_node_name`

Submit a report describing your approach, describing your results, and including any interesting observations. Include a link to your modified code on github. Upload a movie of your solution. Include a screenshot of your ariac launch window showing your score after shipping your box. Hopefully, you achieve the max score of 9.