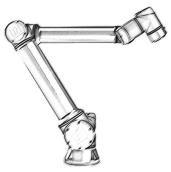


Z. Kootbally & C. Schlenoff
Spring 2023
UMD
College Park, MD

RWA4 (v1.0)

ENPM663: Building a Manufacturing Robot Software System

Due date: **Sunday, April 30, 2023, 9 pm**

Contents

1 Updates

This section describes updates added to this document since its first released. Updates include addition to the document and fixed typos. The version number seen on the cover page will be updated accordingly. There may be some typos even after proofreading the document. If this is the case, please let me know.

- **v0.0**: Original release of the document.

2 Conventions

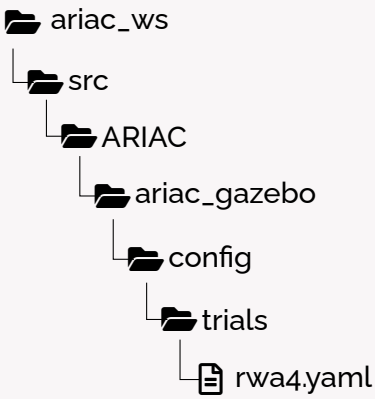
In this document, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

- This is a file.txt
- This is a folder
- This is a package
- This is an important note
- To do
- This is a warning
- This is a link
- This is a t topic
- This is a s service

3 Prerequisites

- Read about the High-priority Orders Challenge and the Faulty Parts Challenge.
- Read about the Combined Task.
- Retrieve the file rwa4.yaml from Canvas and place it in the config folder as shown in Figure 3.1.
- To start the environment with rwa4.yaml
  - `ros2 launch ariac_gazebo ariac.launch.py trial_name:=rwa4`

Figure 3.1: Trial file location.



4 Assembly

The functionalities for the ceiling robot is not yet placed in a separate Node as it is for the floor robot. You can do it yourself if you are in a rush to test assembly. I will try to make this available ASAP and I will update the prerequisites section when this is done.

5 Assignment Description

- ⚠ This is a group assignment.
- Reuse the package from RWA3 and augment it to handle the tasks required in this assignment.
- Resources needed for the assignment can be found in the official documentation.

This assignment consists of completing a combined task and a kitting task.

- The Order with the combined task is announced at the beginning of the competition. For the kitting part of the combined task, use the tray with id 0.
- The Order with the kitting task is announced 50 s after the competition has started.
  - This second Order is a high-priority Order which should be handled as soon as it is announced.

5.1 Challenges

- *High-priority Order*: A kitting Order, which is of high-priority, is announced 50 s after the competition has started. For this assignment, you will use the floor robot to perform kitting (including kitting during a combined task) and the

ceiling robot to perform assembly. The high-priority Order must be started as soon as it is announced. If the robot is holding a part when the new Order is announced, the competitor control system (CCS) should place the part before starting working on the new Order. Once the high-priority Order is completed, the CCS should resume the other Order. There is a need to keep track of the steps within an Order so it can be resumed.

- *Faulty Part Challenge*: The first part placed in quadrant on AGV 4 is faulty. Refer to the lecture on agility challenges for the quality check service. Once a part is detected as faulty, it must be removed and replaced with another one. Make sure you discard the faulty part in one of the blue bins.

6 Package Submission

We only accept packages submitted on Canvas. Before compressing your package, make sure you do the following:

1. Remove the build, log, and install folders.
2. Rebuild the workspace.
3. Re-run your Nodes with this assignment.

Make sure to include the instructions on how to run your Node or Nodes. The instructions must written in instructions.txt and placed in the folder etc in your package (create this folder if it does not exist).

7 Grading Rubric

- This assignment is worth 30 pts.
  - 15 pts will be awarded if your package can perform the trial defined in rwa4.yaml
  - 5 pts will be awarded for documentation. Although it is not required to generate HTML documentation for this assignment, it is crucial that you document your code. All Python classes, functions, and methods must be documented with docstring. All C++ classes, functions, and methods must be documented with Doxygen. You can document Python code with Doxygen but I never tried it.
  - 10 pts will be awarded if your code can handle a variation of rwa4.yaml. This will showcase that your CCS is agile and you did not hard code dynamic aspects. Your code should be able to perform the trial given the following modifications to rwa4.yaml:

1. The time announcement for the high-priority Order will be different.
2. The faulty part will be different. Instead of the pump, we will use a different part type (we will make sure purple pumps in bins are replaced with new part types). The CCS should be able to remove any faulty parts from the AGV.
3. Use different bins and change the slots for parts within the bins.
4. Change the quadrant for the part in the kitting Order.
5. Change the tray id.
6. Change the tray location in the workcell.