ENPM663: Building a Manufacturing Robotic Software System

RWA #3

v1.0

Lecturer: Z. Kootbally

School: University of Maryland

Semester/Year: Spring/2025

MARYLAND APPLIED GRADUATE ENGINEERING

Table of Contents

Changelog

- *v1.0*: Original version.

Conventions

Conventions

bad practice	👎
best practice	👍
code syntax	</>
example	⚗
exercise	✎
file	📄
folder	📂
guideline	🚀
note	📝
question	?
task	☑
terminology	📜
warning	🔥
web link	link
package	📦

ROS node	n /node
ROS topic	t /topic
ROS message	m /message
ROS parameter	p /parameter
ROS frame	f /frame

Prerequisites

- Download **rwa3_spring2025.yaml** from Canvas and use it for this assignment.
- The updated trial includes parts located in a single bin and on the conveyor belt.
- The conveyor belt begins moving once the competition is initiated.

Guidelines

This assignment must be completed as a *group*. All instructions outlined below must be followed precisely. Failure to adhere to these rules may result in a grade of *zero*.

- You may not reuse solutions or components created by others for similar assignments.
- Your work must remain confidential — do not share your files or code with anyone.
- If you choose to use GitHub, your repository must be set to *private*.
- You are encouraged to discuss general concepts and ideas with peers, but sharing specific implementation details is strictly prohibited.
- *The use of AI-generated code (e.g., ChatGPT, Copilot) is not allowed.*

Overview

In this assignment, students will configure sensors in the ARIAC system to detect key elements: trays, parts in bins, and parts on the conveyor belt. The sensing solution must be robust, efficient, and provide accurate information to support robot manipulation tasks.

This assignment is a continuation of the previous one. After receiving an order, students will use sensors to find parts and trays needed in the order(s). The system must log the poses (position and orientation) of these key elements in the ***sensor frames***, following specific requirements.

Package

Package

Building upon the previous assignment's package, students will now extend it with additional sensor functionalities to meet the new requirements.

The package must be submitted via Canvas, and students must ensure that the sensor file is included.

Make sure that the instructions for launching the package nodes are provided in a text file.

Task Description

Task Description

Order Processing Requirements

1. For each received order, identify the required parts and trays.
2. Configure sensors to detect these elements in the environment.
3. Log only the necessary elements based on the following rules:
   - Log only the exact number of parts needed (e.g., if an order requires 2 red batteries, log only 2 out of the available 4).
   - Log insufficient parts warnings (e.g., if an order requires 3 red batteries but only 2 are available).
   - Log exactly one tray of each required type (e.g., if an order requires one tray ID 2 and two are available, log only one).
   - Track and log parts moving on the conveyor belt.

Robust Sensor Placement Strategy

Resources

- [ ] Trial Configuration: Information on how to customize a trial configuration file.

Robust Sensor Placement Strategy

For a robust sensor placement, students will need to edit the provided trial configuration file, **rwa3_spring2025.yaml**, to strategically place parts and trays in different configurations to test sensor coverage from multiple angles. The trial configuration file allows students to define:

- Exact positions and orientations of trays.
- Part types, quantities, and arrangements in bins.
- Timing and sequence of parts arriving on the conveyor belt.

> This whole course focuses only on kitting tasks.

Technical Requirements

Resources

- [ ] **Sensor Specifications**: Information on available sensor types, capabilities, and limitations.
- [ ] **Sensor Communications**: Details on ROS topics and messages used for each sensor type.
- [ ] **Sensor Configuration Files**: Guidelines for defining sensor types and positioning them in the environment.

Sensor Configuration

Create a sensor configuration file following the ARIAC documentation format.

- Position sensors to cover all critical areas: tray locations, part bins, and conveyor belt.
- Select appropriate sensor types for each detection task.
- Optimize sensor placement to minimize the number of sensors while ensuring complete coverage.

Tray Detection

Students configure and implement sensors to accurately identify and locate trays in the environment. The system must determine the precise pose (position and orientation) of each tray in the *sensor frame*. Students should develop algorithms that can distinguish between different tray types based on their ID markers.

Parts Detection in Bins

Students implement a robust sensing solution to detect and identify parts stored in stationary bins throughout the environment. The system should determine the type, color, and exact pose of each part within the bin. Students must address scenarios such as varying part orientations. Efficient algorithms should be developed to process sensor data and provide accurate information about the parts inventory in each bin, with special attention to detecting the specific parts required for current orders.

Parts Detection on Conveyor Belt

Students create a dynamic sensing system capable of tracking parts as they move along the conveyor belt. The solution must detect part types, colors, and calculate their current poses in real-time, accounting for the continuous motion of the belt. Students should implement predictive algorithms to anticipate future part positions based on belt speed. The system should handle varying part arrival patterns and maintain tracking accuracy.

> In **rwa3_spring2025.yaml**, purple pumps and blue batteries can only be found on the conveyor belt.

Logging

For each order, log part information using the following template:

```
-Order ID:
    -Tray ID: [x, y, z] [a, b, c, d]
    -Parts:
        -color type:
            -Location: bin#
            -[x, y, z] [a, b, c, d]
        -color type:
            -Location: conveyor
            -First detection: [x, y, z] [a, b, c, d]
            -Prediction [1s]: [x, y, z] [a, b, c, d]
            -Prediction [2s]: [x, y, z] [a, b, c, d]
            …
        …
```

Conveyor tracking requirements:

- Students should log the initial detection of each part when it first enters the sensor's field of view.
- The system should continuously track the part's position but only log significant updates at regular time intervals (e.g., every 1s) to maintain tracking history.

Demonstration

A logging example containing specific information as soon as an order part is detected is presented below (note that the numeric values are illustrative and not based on real data).

```
-Order KT02:
    -Tray 2: [0.25, 0.30, 0.00] [0.0, 0.0, 0.0, 1.0]
    -Parts:
        -orange regulator:
            -Location: bin5
            -[0.12, 0.45, 0.00] [0.0, 0.0, 0.0, 1.0]
```

When a required part for the order is detected on the conveyor belt, the system logs the following information.

```
-Order KT02:
    -Tray 2: [0.25, 0.30, 0.00] [0.0, 0.0, 0.0, 1.0]
    -Parts:
        -orange regulator:
            -Location: bin5
            -[0.12, 0.45, 0.00] [0.0, 0.0, 0.0, 1.0]
        -blue battery:
            -Location: conveyor
            -First detection: [0.12, 0.51, 0.00] [0.0, 0.0, 0.0, 1.0]
            -Prediction [1s]: [0.12, 0.53, 0.00] [0.0, 0.0, 0.0, 1.0]
            -Prediction [2s]: [0.12, 0.55, 0.00] [0.0, 0.0, 0.0, 1.0]
            …
```

Deliverable

Deliverable

Compress the package into ***a single file*** and submit it on Canvas by the due date.

Late submissions will incur a penalty according to the guidelines specified in the syllabus, with exceptions for any valid reason.

- Valid reasons include a doctor's note, proof of travel, or a note from a professor/MAGE.

> Students with special circumstances may submit their assignments late without incurring any penalties. However, it is required that these students inform me in advance of their intention to submit their work past the deadline.

Grading Rubric

Grading Rubric

Grading Rubric *30 pts*

- *Sensor Configuration File (5 pts)*
  - *3 pts*: Proper sensor placement for full coverage of trays, bins, and conveyor.
  - *2 pts*: Each sensor in the configuration file is assigned a clear and descriptive name. For example, ***rgbd_left_bins*** is better than the generic name ***sensor1***.
- *Parts Reported in Bins (8 pts)*
  - *3 pts*: Correct detection of part types, colors, and poses in bins.
  - *3 pts*: Reports only the number of parts required by the order.
  - *2 pts*: Accurate logging format.
- *Parts Reported on Conveyor Belt (14 pts)*
  - *5 pts*: Timely detection of parts entering the sensor's field of view.
  - *5 pts*: Provides accurate pose predictions for conveyor parts at 1-second intervals, based on belt speed and initial detection data.
  - *2 pts*: Consistent tracking and logging of moving parts.
  - *2 pts*: Accurate logging format.
- *Correct Submission (3 pts)*
  - *3 pts*: The submission contains the sensor configuration file, instructions to run the node(s), and updated **package.xml**.

Deductions

- Late submission: *As per syllabus guidelines*.
- Use of AI-generated code: Assignment grade of *zero*.
- Code sharing/plagiarism: Assignment grade of *zero*.
- Non-private GitHub repository: Assignment grade of *zero*.
- Classes lack adequate documentation or insufficient inline comments: *-2 pts*.
- Hardcoding information that should be retrieved dynamically: *-5 pts*.
  - If you are not sure whether some information should be retrieved dynamically, ask me.