Automated, Robotic Order Fulfillment

Matthew Boyd - Spring 2020 | *Faculty: Osama Alshaykh*

github.com/mcboyd-bu/auto-order-fulfillment

INTRODUCTION

Utilize computer vision, robot operating system (ROS), and multi-robot control systems to design a collaborative, multi-robot system to automate a logistics order fulfilment process. Originally planned as a system for helping fulfill retail orders by recognizing and manipulating individual retail products. As challenges arose over the course of the semester, the scope evolved to target an industrial warehouse pick-and-place order system instead.

KEY METRICS

- Order accuracy
- Order fulfillment time
- Collision avoidance
- Sensor management

GOAL

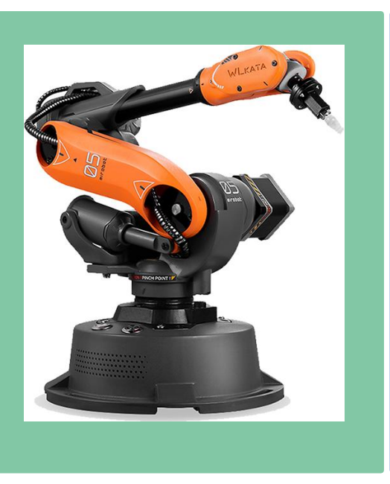
Learn and demonstrate some of the skills and tools needed to pursue a career as a robotics software engineer.

PROJECT BREAKDOWN AND ADJUSTMENTS

Semester Start

Initial Version

MECHANICAL
(2) 6-DOF Arms, 3D Camera, 1 Coord. Frame Transform



SIMULATION
2 Arms, Minimal Environment

COMPUTATION
ROS: 6 Topics

The 6-DOF arms originally scoped were unavailable.

Version 2: New Arms

MECHANICAL
(2) 4-DOF Arms, 3D Camera, 1 Coord. Frame Transform

SIMULATION
2 Arms, Minimal Environment

COMPUTATION
ROS: 6 Topics

Campus Closed! Working from home = no hardware.

Version 3: All Simulation

SIMULATION
(2) 6-DOF Arms, Conveyor Belt, Logical Camera, 2 Coord. Frame Transforms; Edit Pre-Built Environment

COMPUTATION
ROS: 10 Topics

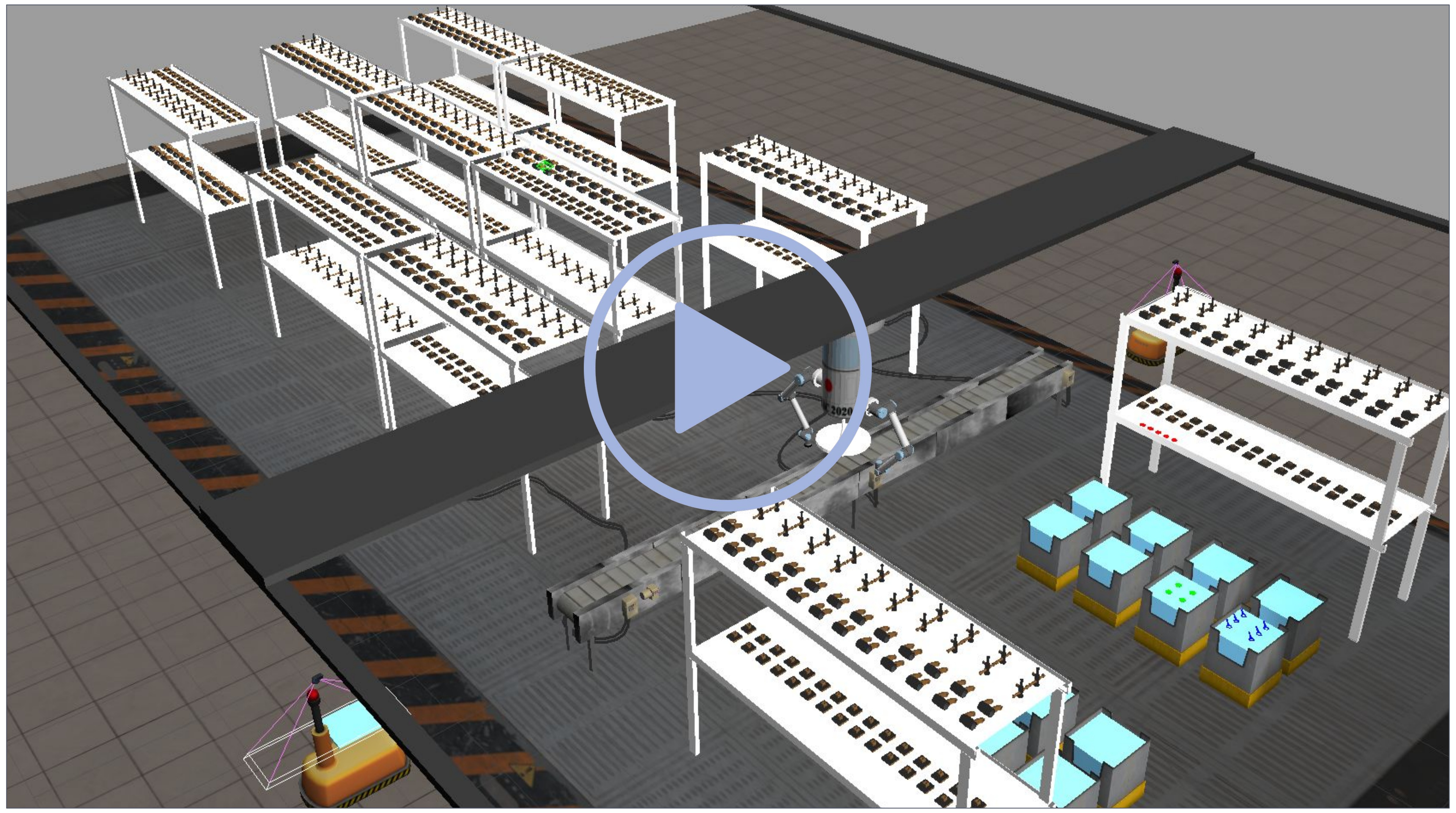
Simulation changes too extensive. 2020 ARIAC Competition a better fit.

Ver. 4: ARIAC Competition

SIMULATION
3-DOF Gantry on Rails, (2) 6-DOF Arms, (14) Logical Cameras, 8 Coord. Frame Transforms; Edit Extensive Pre-Built Environment

COMPUTATION
ROS: 24 Topics
MoveIT: Path Planning
Command / Control: Full Automation

FINAL SIMULATION ENVIRONMENT



SYSTEM COMPONENTS

Simulation: Gazebo 9.0 | Path Planning: MoveIT | Control: ROS 1.0 Melodic | Python, YAML

LEARNING

edX "Hello ROS" course:
- 6 weeks, ~80 hrs
- ROS, Gazebo, MoveIT, rViz

Research for New Arms:
- Cost vs. Capabilities
- Simulation Assets

Investigate Pre-Built Gazebo Environments & Attempt to Modify

Outdated 4-DOF Arm Simulation Asset:
- Fork Outdated Code
- Manually Update from Gazebo 7 to Gazebo 9
- Publish Updated Code

Update Ideal New Simulation Environment:
- Remap Arm Joints
- Add Logical Cameras
- Design Control Algorithm

WHAT IS ARIAC?

Agile Robotics for Industrial Automation Competition

Sponsored annually by U.S. National Institute of Standards and Technology (NIST)

"...simulation-based competition designed to promote robot agility by utilizing the latest advances in artificial intelligence and robot planning."

ABOUT THE SIMULATION

ARIAC provides the Gazebo simulation environment pre-populated with primary components:

- Gantry robot attached to overhead sliding rails with (2) 6-DOF arms attached, 1 per side
- Pre-placed shelves, bins, and conveyor belt where parts needed to fulfill orders will spawn

My task: add cameras to detect parts in the environment, programmatically determine location of parts needed for orders, calculate path to each needed part, and then autonomously navigate the gantry robot to retrieve all the parts and deliver them to the correct cart.

Development Milestones:

- Initial env. w/ known part locations
- Add 14 cameras to identify parts
- Scenario A: Fulfill 2 orders autonomously
- Scenario B: Fulfill 1 order as grippers fail