ENPM 662: Introduction to Robot Modeling

Project 2: Group 2 Report

Barista Robot

Group Members: Manas Desai, Grayson Gilbert, Kunj Golwala, Shreya Kalyanaraman

---

1. Introduction

The development of autonomous systems has revolutionized various industries, enabling automation of repetitive tasks and improving efficiency. This report focuses on designing and implementing an autonomous coffee-making robot, a system aimed at automating the process of brewing and serving coffee. The primary objective is to create a robot capable of performing tasks such as identifying cups, dispensing ingredients, and delivering consistent coffee quality with minimal human intervention. The autonomous coffee-making robot is designed to operate seamlessly in environments such as cafes, offices, and homes. The technologies that we used for doing this project were:

1. ROS 2 (Robot Operating System): The backbone of the project, ROS 2 provides a robust framework for controlling and integrating various robotic components. Its modular architecture facilitates communication between nodes responsible for perception, planning, and actuation.
2. Gazebo Simulation: A powerful simulation environment used for testing and validating the robot's performance in realistic scenarios before deployment. It allows for fine-tuning the robot's behavior without physical hardware.
3. RViz: A visualization tool used to monitor sensor data and validate the robot's perception capabilities, particularly for vision-based tasks like cup detection.

2. Application

The autonomous coffee-making robot has versatile applications across various domains. In commercial settings such as cafes and coffee shops, it automates the entire coffee preparation process, reducing wait times during peak hours while ensuring consistent quality. Offices can benefit from quick, customizable coffee options, enhancing employee satisfaction and productivity without requiring dedicated staff.

The system significantly improves service efficiency by automating repetitive tasks, allowing businesses to serve more customers in less time. Its scalability makes it adaptable for environments ranging from small cafes to large-scale operations, maintaining quality and speed regardless of demand.

Additionally, the robot integrates seamlessly with smart home devices or mobile applications, enabling users to order coffee remotely or schedule brewing times. This feature aligns with the growing trend of automation and convenience in daily life.

By addressing these practical needs, the autonomous coffee-making robot showcases how robotics can transform routine tasks into efficient, scalable, and user-friendly solutions.

3. Robot Type

We selected a Universal Robots UR10e collaborative robot arm for the barista robot application. The UR10e is a 6 degree of freedom (DOF) serial manipulator with a 1300mm reach and a 12.5 kg payload capacity. Its stationary base is fixed to a pedestal, elevating it roughly 3 feet off the ground. We decided to use a serial manipulator instead of other options like a mobile robot or gantry-style robot because the serial manipulator can easily reach all the required workspace areas while maintaining a small overall footprint. The UR10e manipulator was selected specifically, due to its wide use in industry, extensive reach, sufficient payload capacity, and high precision repeatability. This ensured that the barista robot would be able to reach all required areas of the workspace as well as safely and reliably handle the coffee cups during the brewing process. Attached to the end of the arm is a modified Robotiq 2F-140 gripper. The gripper was converted into a serial manipulator to simplify the simulation process. This gripper in particular was selected because it is directly compatible with the UR10e, and it can apply 10N-125N of grip force, enabling consistent cup handling without slippage. Also attached to the robot near the end effector is an Intel RealSense D435 Camera. This camera enables the robot to detect the presence of cups within the workspace and gather other visual information.

4. DOF and Dimensions

The UR10e is a 6 DOF serial manipulator. The manufacturer-specified dimensions are described in the image below.

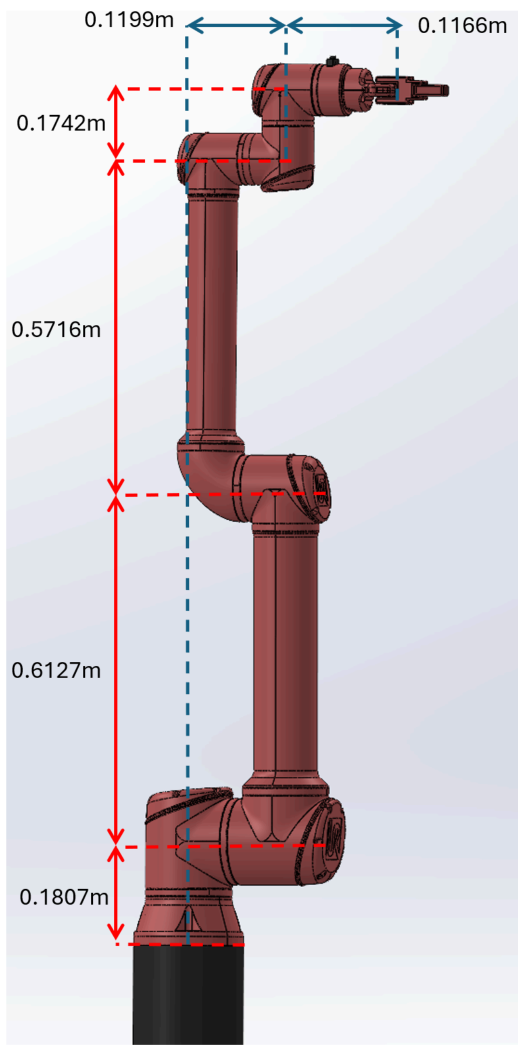

Figure 1: Dimensions of the Barista Robot

5. CAD Models

The CAD assembly was too large to commit to the Github Repository. Instead, the zip file containing all parts and assembly files is attached to the Canvas project submission.

6. DH Parameters

i) Assigning DH Frames: The frame assignment for the UR10e is specified in the following image.

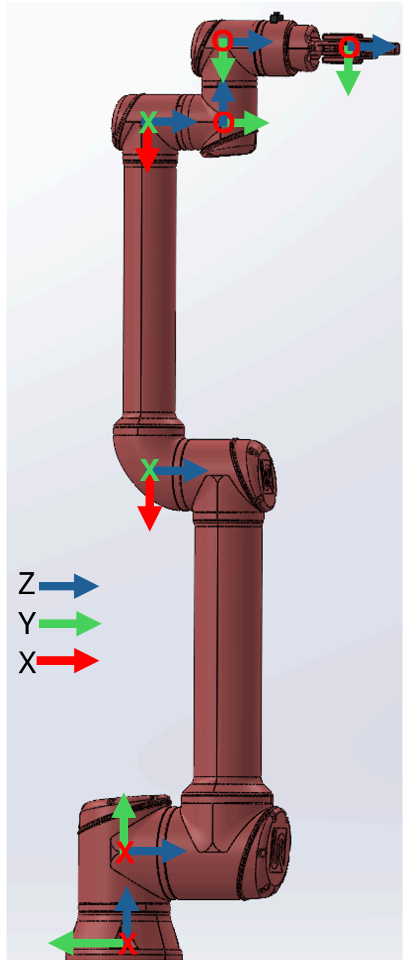

Figure 2: Frame assignment for the Barista Robot

ii) DH Table: Utilizing the frames specified in the image above and the dimensions of the manipulator shown in the DOF and Dimensions section, the resulting DH Table is listed below.

DH Table	theta (rad)	a (m)	d (m)	alpha (rad)
Joint 1	$\theta 1$	0	0.1807	$\pi/2$
Joint 2	$\theta 2$	-0.6127	0	0
Joint 3	$\theta 3$	-0.57155	0	0
Joint 4	$\theta 4$	0	0.1741	$\pi/2$
Joint 5	$\theta 5$	0	0.1199	$-\pi/2$

Joint 6	θ6	0	0.1166	0

Table 1: DH parameter table for the Barista Robot

7. Forward Kinematics -

Forward kinematics is a fundamental concept in robotics, allowing us to compute the position and orientation of the robot's end-effector based on its joint parameters. The Denavit-Hartenberg (DH) convention was used for this project to systematically derive the transformation matrices between adjacent joints. The DH table provides a structured way to define the robot's geometry, specifying parameters such as link lengths, offsets, and joint angles.

The process begins by computing individual transformation matrices for each joint using the DH parameters. These matrices describe how each joint and link is positioned relative to its predecessor. The transformation matrices are then multiplied sequentially to obtain the final transformation matrix, which represents the pose of the end-effector in the base frame of the robot.

$$T_0^6 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4 \cdot T_4^5 \cdot T_5^6 \tag{1}$$

The final transformation matrix encapsulates both the position and orientation of the end-effector. The position is extracted from the first three elements of the last column of the matrix, corresponding to T[3,0], T[3,1], and T[3,2], which give the x, y, and z coordinates, respectively. The orientation is represented by the rotation components of the matrix. Mathematically, this can be expressed as: (1)

Where T_0^6 is the final transformation matrix from the base frame to the end-effector frame, and each T represents a transformation between two consecutive joints.

8. Forward Kinematics Validation -

We visualized the end-effector position for different joint angles in the gazebo and analyzed whether they made sense intuitively or not.

i) Test Pose 1:

[Joint 1 - 0.0 , Joint 2 - pi/2 , Joint 3 - 0.0 , Joint 4 - 0.0 , Joint 5 - 0.0 , Joint 6 - 0.0]

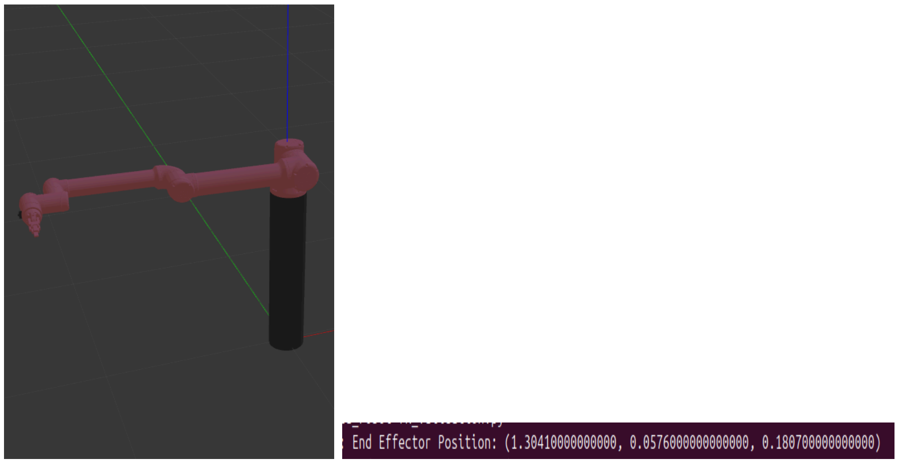

Figure 3: Forward Kinematics validation Pose 1

ii) Test Pose 2:

[Joint 1 - 0.0 , Joint 2 - 0.0 , Joint 3 - pi/2 , Joint 4 - 0.0 , Joint 5 - 0.0 , Joint 6 - 0.0 }

Figure 4: Forward Kinematics validation Pose 2

iii) Test Pose 3:

[Joint 1 - 0.0 , Joint 2 - pi/2 , Joint 3 - 0.0 , Joint 4 - (-pi/2) , Joint 5 - (-pi/2) , Joint 6 - 0.0]

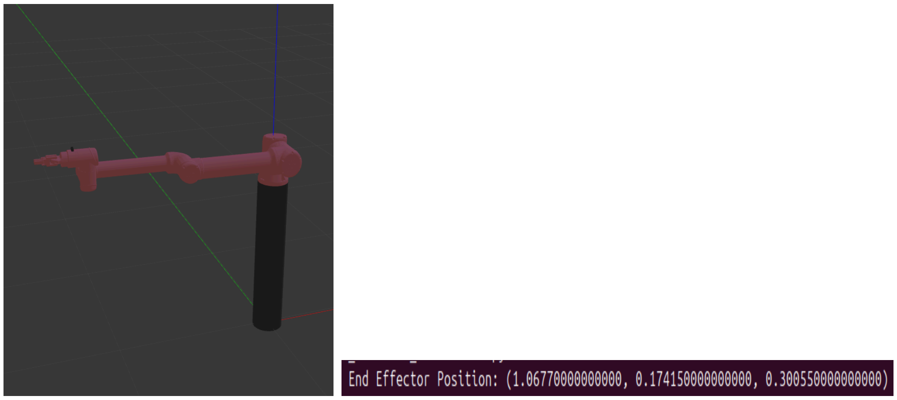

Figure 4: Forward Kinematics validation Pose 3

9. Inverse Kinematics -

Inverse kinematics is a critical component of robotic control that determines the joint configurations required to achieve a desired end-effector position and orientation. For the barista robot, inverse kinematics enables precise positioning of the robotic arm to perform tasks such as picking up cups, operating dispensers, and delivering coffee.

i) Overview

The inverse kinematics problem involves calculating the joint angles of a robotic arm that correspond to a given position and orientation of the end-effector in Cartesian space. Unlike forward kinematics, which has a straightforward solution, inverse kinematics is more complex.

ii) Mathematical Approach

The solution to inverse kinematics involves solving a set of nonlinear equations that relate the joint angles to the end-effector's desired position and orientation. This can be expressed as:

$$x_{\text{desired}} = f(q) \qquad (2)$$

Where:

- $x_{\text{desired}} \in R^6$ - The desired end-effector position and orientation in Cartesian space.

  The 3 positional values: [x,y,z] represent the coordinates of the end-effector.

  The 3 orientation values: [roll, pitch, yaw] represent the end-effector's orientation in Euler angles.

- f(q) - The forward kinematics function that maps the joint angles q to the end-effector's position and orientation.
- $q \in R^6$ - The vector of 6 joint angles for the UR10e robot, corresponding to its 6 revolute joints is the vector of joint angles. $q = [q_1 , q_2 , q_3 , q_4 , q_5 , q_6]$

Using the **Jacobian matrix**, we linearize the relationship between joint velocities (q_dot) and end-effector velocities (x_dot) :

$$x' = J\, q' \tag{3}$$

By inverting the Jacobian, we compute the required joint velocities:

$$q' = J^{+}\, x' \tag{4}$$

Where J^{+} is the **pseudoinverse** of the Jacobian, used to handle cases where J is non-square or near-singular.

To move from joint velocities q' to joint angles q numerical integration is applied over time:

$$\mathbf{q}(t) = \mathbf{q}(t_0) + \int_{t_0}^{t} \dot{\mathbf{q}}(t')\, dt' \tag{5}$$

Euler Integration: A simple numerical method for integration:

$$q(t + \Delta t) = q(t) + q'(t) \cdot \Delta t \tag{6}$$

where, Δt: Time step for integration.

The process involves computing q' (joint velocities) using inverse kinematics, updating q (joint angles) via numerical integration, and sending these values to the robot's actuators.

iii) Implementation of the Barista Robot

Inverse kinematics was used to ensure that the barista robot was able to reach exact positions including serving counters and coffee dispensers.

The robot was able to correctly orient the end-effector to handle tasks like pouring coffee and grabbing glasses.

Configurations, where the Jacobian matrix might become singular, were avoided by using the pseudoinverse of the Jacobian.

10. Inverse Kinematics Validation -

For validation of inverse kinematics, we used the trajectory provided in HW 2 (The semicircular trajectory connected by three straight lines) and made our Barista robot follow it. The trajectory followed by the end effector matched the desired trajectory as shown in Figure below -

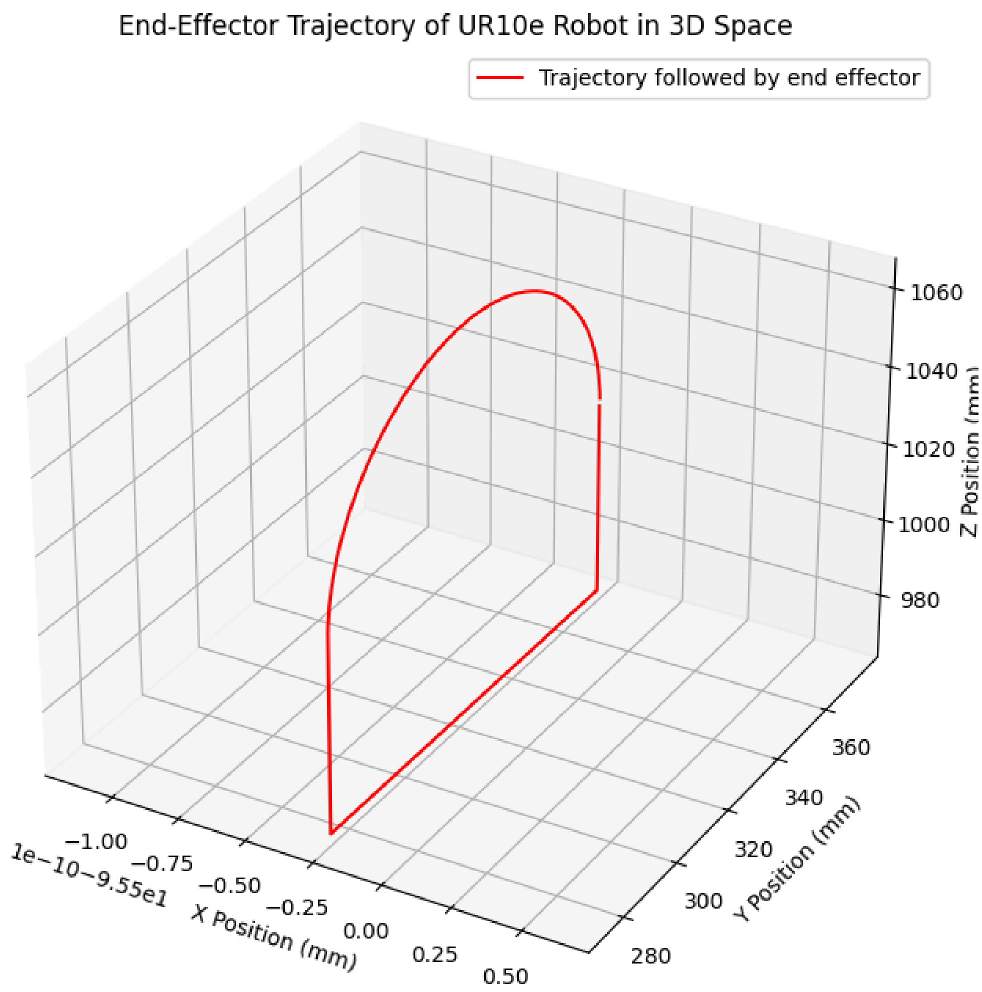

Figure 5 : Inverse Kinematics Validation - End effector trajectory of Barista Robot in 3D Space

The above generated trajectory plots showcase the successful and accurate implementation of inverse kinematics since the Barista Robot was able to follow a desired trajectory - made up of a desired set of values for end effector position and orientation.

11. Workspace Study -

Finding the operational limits within which the robot may efficiently execute tasks depends on the analysis of the workspace. The workspace for the Barista Robot is the 3D volume the robot can reach with its end-effector - that part of the robot in charge of interacting with the surroundings, such as grabbing a cup or running a coffee dispenser). Ensuring the robot can access all required areas in its surroundings - coffee dispensers, cup holders, the serving counter - requires an awareness of the workspace.

i) Concept of Workspace -

Based on its physical design - that is joint limitations, arm length, etc. The workspace of a robot is the extent of space the end-effector can reach. The workspace of the 6-DOF robotic arm can be theoretically described as a sphere with a radius matching the maximum reach of the arm by means of its joint angles and link lengths. With this spherical workspace, the robot may travel to any place in the 3D space within arm's reach.

However, in our application, we will limit the workspace to a hemisphere above the base of the robot. The robot does not need to reach or function in places below the table or serving counters, hence this restriction is created under counters or other surfaces where it would be impractical for the robot to interact with objects. We simplify the planning and operation of the robot by limiting the workspace to the upper hemisphere, guaranteeing that it only interacts with objects placed above the table, including coffee dispensers, cups, and other equipment.

ii) Reachable Workspace and Robot Reach

For the Barista Robot (UR10e), the maximum reach is determined by the arm's length and the range of motion of each joint. The arm has a reach of 1300 mm, meaning it can cover a spherical workspace with a radius of 1300 mm from the base of the robot.

The total workspace volume for the robot can be approximated as the volume of a hemisphere, with the formula for the volume of a hemisphere being:

$$V = \tfrac{2}{3} \boldsymbol{\pi} r^3 \tag{7}$$

Substituting the robot's maximum reach (r = 1.3 meters):

$$V = \tfrac{2}{3} \boldsymbol{\pi} (1.3)^3 = 3.08 \text{ m}^3 \tag{8}$$

The robot manipulator will thus cover in its limited workspace (hemisphere) an estimated volume of 3.08 cubic meters.

iii) Reachability and Task Constraints

Ensuring that the intended tasks fall inside the available workspace will help to guarantee that the Barista Robot can handle chores including picking up cups, dispensed coffee, and cup placement. Works including:

- The robot has to be able to orient its end-effector at the right height and angle to grab a cup.
- The robot needs to be able to get to the coffee dispensers so it may pour coffee into the cup without any collisions.

iv) Workspace Constraints

Physical Environment: The workspace is further constrained by obstacles in the environment, such as the coffee machine, counter, and other objects in the robot's vicinity. These constraints are taken into account when designing the robot's movement trajectories.

12. Assumptions -

The following assumptions were used throughout the barista robot project:

1. No real liquid is present in the simulation
2. Dispensers, cups, and tables are placed in fixed, predefined locations
3. All coffee cups are uniformly sized, lightweight, and rigid
4. There are no unexpected obstacles in the environment

13. Control Method

i) Open-Loop Control

The robotic arm operates using a predefined trajectory, ensuring simplicity in control and minimizing the need for real-time adjustments. This method is suitable for tasks with minimal external disturbances or variations.

ii) Trajectory Planning

The desired End-Effector Velocity is calculated to meet task requirements within a specified duration. Ensures smooth and efficient movement between stations, such as the serving table, coffee dispenser, and milk dispenser.

iii) Jacobian-Based Control

Jacobian-based control utilizes inverse velocity kinematics to ensure precise control of the robotic arm. The joint velocity vector is calculated using the below equation:

$$\dot{q} = J^{-1}\dot{x}, \tag{9}$$

where q’ represents the Jacobian matrix that relates joint velocities to the end-effector velocities, and x’ is the desired end-effector velocity. This method ensures accurate positioning and movement of the end effector, allowing smooth and seamless transitions during coffee preparation.

iv) Gravity Compensation

To counteract gravitational forces, joint torques are calculated using a gravity matrix. This compensation ensures Stability during arm movement and precise handling of lightweight objects, such as coffee cups.

14. Gripper Control Method

i) End-Effector Role

The gripper is essential for picking, holding, and releasing coffee cups during predefined sequences. It ensures secure handling without slippage, critical for consistent coffee-making operations.

ii) Control Mechanism

Position commands are sent directly to the gripper’s joints, enabling precise control of finger movements. Commands are published to the topic /gripper_position_controller/commands. These commands ensure the gripper’s adaptability to handle objects of varying sizes and shapes.

15. Gazebo and RViz Visualization -

i) Gazebo World: The custom Gazebo world for the project consists of 4 main stations:

1. Serving Table - an empty table where the finished coffee is placed
2. Cup Selection Table - a table where the cups are positioned for the robot to grab
3. Coffee Dispensing Table - the table where coffee is poured into the cup
4. Milk Dispensing Table - the table where the milk is poured into the coffee

The UR10e sits on top of a pedestal in the center of the 4 sections allowing easy access to reach all points within the workspace.

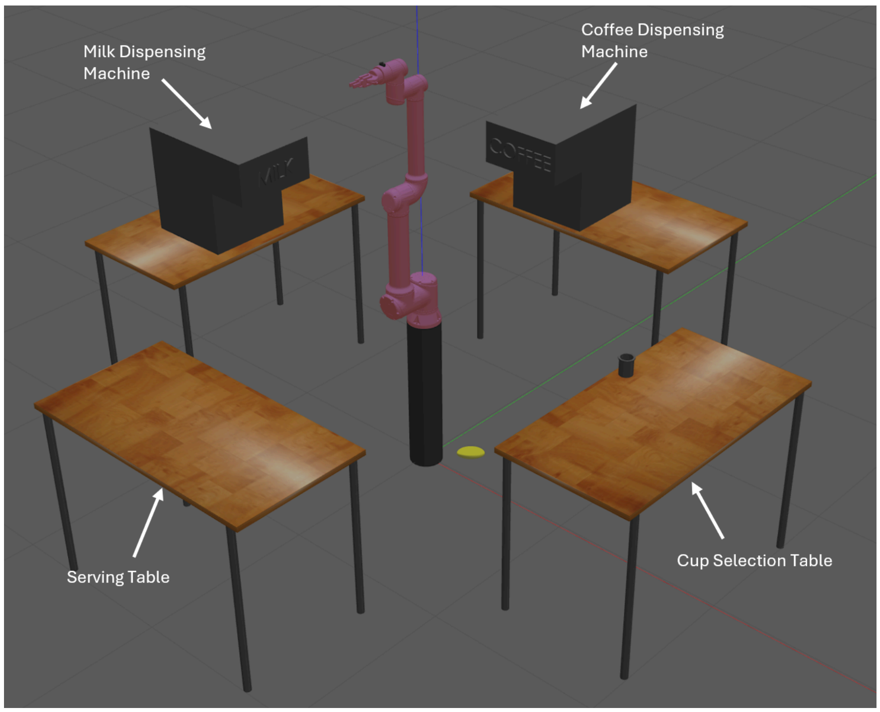

Figure 6: Gazebo environment for the simulation

ii) RViz Simulation: The RViz simulation provides a unique perspective on the robot throughout the simulation. The two main features used for our RViz simulation are the RobotModel display and the Image display. The Robot Model tool allows you to see exactly what the robot is doing in real-time, similar to the Gazebo environment. This tool was useful early on in the project when validating the joint transformations from the robot URDF file. The Image display shows exactly what the onboard camera is seeing during the simulation. This was essential to the development of the object detection system in the barista robot. Below is an image of the RViz simulation configured with the RobotModel and Image displays.

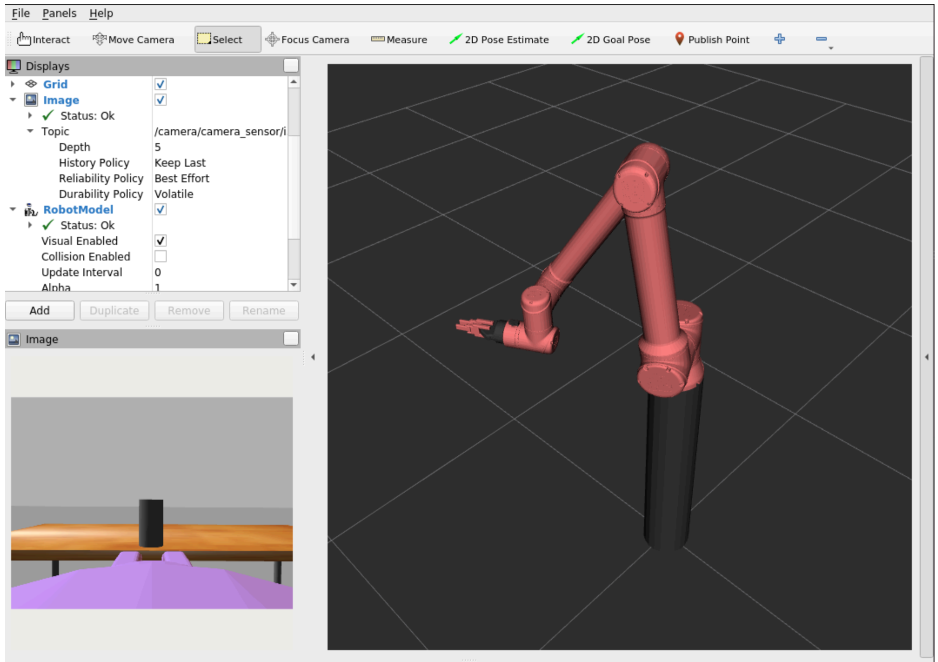

Figure 7: Rviz Simulation images of the Robot Model and the Camera feed

iii) Approach to Simulation: The general approach to the barista robot simulation begins with the UR10e spawning in its home configuration. This is not an ideal position to perform inverse kinematics, so the robot manually moves to its 'ready' position, which is a more favorable pose. From here, the barista robot is ready to repeatedly perform the coffee making sequence. The first step in this sequence is to check if the serving station is clear of cups. If it is not clear, the robot will return to the ready position and notify the user to clear the serving station and try again. If the table is clear, the coffee making sequence can continue. The next step for the robot is to navigate to the cup station, where it will pick up a coffee cup. Then, the UR10e moves and places the cup under the coffee dispensing machine, where it pours coffee into the cup. This process is repeated for the milk dispensing machine. After that, the barista robot will return to the serving station and serve the completed coffee order by placing it on the table.

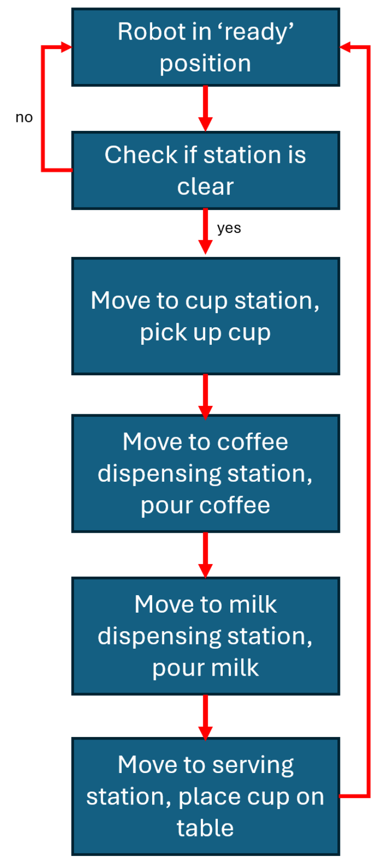

Figure 8: Flowchart of the coffee making process with the Barista Robot

Figure 9 visualizes the 3D plot of our generated trajectory during the complete coffee making process starting from cup detection and cup picking and ending at cup dropping,

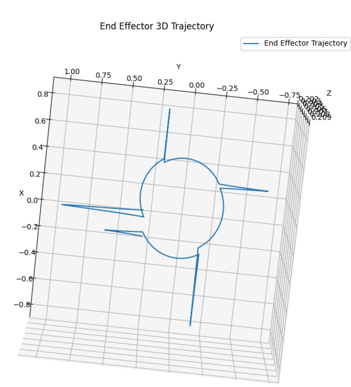

Figure 9: End effector trajectory plot during the complete coffee making process

16. Problems Faced -

i) Robot Arm Breaking: The joints connecting the links of the robot arm frequently separated or broke during the development of the project. Upon closer inspection, it looked like an issue within the simulated dynamics of the Gazebo environment. In order to address these issues, a joint effort controller was designed to provide the necessary torque in the joints of each link to counteract gravity. This method improved the robustness of the simulations, however, it did not fully resolve the issues we experienced. In order to fully develop the trajectory for the robot to travel, we frequently turned off physics within the Gazebo environment to eliminate the issue.

ii) Gripper Breaking: Similar to the robot arm, the gripper also caused a lot of problems within the project. Without any input, the gripper would begin to oscillate uncontrollably until the entire simulation broke. Another problem arose when trying to pick up objects. When closing the gripper around the cup, the collisions with the objects would break the simulations, preventing the robot from successfully retrieving a cup to pour a cup of coffee.

iii) Singularities: The robot spawns in the Gazebo environment in its vertical home position. Performing inverse kinematics from this position proved to be a challenge because it is a singularity condition. In order to resolve this we moved the robot into a more suitable pose that eliminated the potential for singularity throughout the entire trajectory.

iv) Custom world Objects: After creating the custom Gazebo world for the barista robot, we faced a problem trying to get this world to show up when other teammates launched the gazebo environment. The launch file itself worked, however, all the custom objects for the world would not appear. After testing out various methods to resolve the issue, we were unsuccessful in sharing the custom gazebo world.

17. Lessons Learned -

i) Robot Home Position: One of the key takeaways from the project was the importance of the initial robot joint configuration. Starting the robot in the regular home position of all theta values equal to zero was not optimal because this is a singularity. However, arbitrarily setting the initial robot configuration was also not preferable as it was difficult to know exactly how the inverse kinematics would move the various joint angles. This lead to encounters with other singularity positions. Additionally, returning to the initial configuration was not guaranteed even if the end effector was in the same position. After some testing, an optimal starting pose was found that enabled the robot to achieve a path while avoiding singularities, performing repeatable movements, and returning to the exact same initial pose position.

ii) RViz for Visualization: Throughout the project development, RViz was a key tool to successfully integrate sensors onboard the robot. Without RViz, it would have been extremely difficult to debug the object detection components within the system. As a standalone test, the camera and object detection algorithm worked reliably. However, after integrating it with the robot arm, it did not work at all. Using RViz and the live camera feed coming from the robot, we were able to quickly identify the alignment issues in the URDF which otherwise would have been impossible to pinpoint. After applying these small changes, the onboard object detection system worked consistently for our application.

iii) Rqt_graph: As the project progressed, the number of files and nodes within the system rapidly increased. It was unclear how each of these nodes interacted with one another while looking through the terminal. Using rqt_graph allowed us to quickly visualize the active nodes within our ROS project and see how they all communicated with each other. This proved to be extremely beneficial when attempting to resolve errors, or unexpected behavior in the system.

18. Conclusions -

i) Successful Simulation:

- Developed and successfully simulated an autonomous coffee-making robot using ROS 2 and Gazebo.

ii) Technological Integration:

- Designed custom control algorithms to automate coffee making and showed integration of robotic arm, sensors, and a custom gripper

iii) Benefits of Automation:

- Highlighted the potential benefits in service industries, including consistency in quality, reduction in human effort, and improved efficiency.

19. Future Work -

i) Continued development of the gripper functionality: Further refinement and optimization of the gripper to improve precision and adaptability for handling a variety of cup sizes and materials.

ii) Specialized coffee orders: Incorporating the ability to prepare and serve customized coffee orders, tailored to individual preferences.

iii) Autonomous trajectory generation: Developing algorithms for automatic trajectory planning, ensuring efficient and seamless robotic arm movements.

iv) Robust PID Control: Implementing robust PID control mechanisms for better stability and accuracy in the robotic arm's operations.

20. Links to Github Repository and Video Demonstrations -

Github Repository Link: https://github.com/GraysonGilbert/barista_robot

This is a private repository, however, collaboration invitations have been sent to the following email addresses:

1. koustubh@umd.edu
2. rmonfare@umd.edu

Video Demonstration Links:

1. Full Coffee Making Simulation: Click here for video
2. Objection Detection Simulation: Click here for video

21. References:

[1] Myhrvold, N. (2024, October 30). *Coffee | Origin, Types, Uses, History, & Facts*. Encyclopedia Britannica. https://www.britannica.com/topic/coffee

[2] *Robot coffee maker. Menu, ordering, and support.* (n.d.). https://cafe.rozum.com/robot-coffee-maker-process-and-menu

[3] Hochman, D. (2018, May 11). *This $25,000 robotic arm wants to put your Starbucks barista out of business*. CNBC. https://www.cnbc.com/2018/05/08/this-25000-robot-wants-to-put-your-starbucks-barista-out-of-business.html

[4] *ADAM | Robot Bartender | The Exciting Robotic Worker*. (n.d.). https://www.richtechrobotics.com/adam?srsltid=AfmBOoruhOCgLkDsAPlcJT_Q15ZqP9eyKvaHYKC7FdU8jAJhJzvmaFJH

[5] ROS2 Tutorials. (n.d.). https://docs.ros.org/en/galactic/Tutorials.html

[6] ADAM | Robot Bartender | The Exciting Robotic Worker. (n.d.). https://www.richtechrobotics.com/