Introduction to Robot Modeling - ENPM662

Group Project 2 - Barista Robot

Grayson Gilbert | Shreya Kalyanaraman | Manas Desai | Kunj Golwala

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Introduction

Project Overview:

- To design and develop an autonomous robot to make coffee.

Technologies Used:

- ROS 2: Robot Operating System for control and integration of robot components.
- Gazebo: Simulation environment for realistic testing and performance validation.
- Rviz: Used to visualize sensor data.

Impact:

- Potential application in cafes, offices, and home automation for efficient, consistent, and personalized coffee-making.
- While coffee machines simplify brewing, human intervention in tasks like filling cups and adding ingredients leads to inconsistency. Automating these steps reduces wait times, improves efficiency, and allows staff to focus on other tasks.

Application

Commercial Settings:

- Cafes & Coffee Shops: Automates coffee preparation, reducing wait times and ensuring consistent quality.
- Offices: Provides employees with quick, customizable coffee options without the need for a barista.

Service Efficiency:

- Enables businesses to serve more customers in less time, especially during peak hours, by automating repetitive tasks.

Scalability:

- The system can be scaled for different environments, from small cafes to large-scale operations, without compromising on quality or speed.

Integration with Other Systems:

- Can be integrated with smart home devices or apps, allowing users to order coffee remotely or schedule brewing times.

Robot Type

Manipulator Type:

- Universal Robots UR10e: Robotic arm fixed to a stationary, custom-designed base of suitable height.
- 6 Degrees of Freedom (DOF): Allows the arm to easily access the coffee dispenser, milk dispenser, and serving counter.

Reasons for Choosing UR10e:

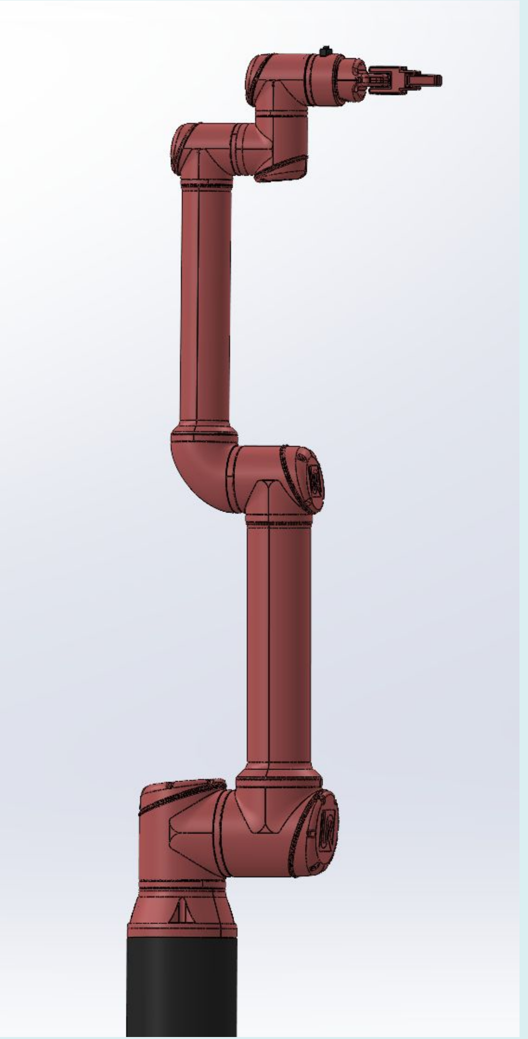
- Reach: 1300 mm reach to efficiently cover different workstations (cup stands, dispensers, tables, etc.)
- Payload Capacity: Can carry up to 12.5 kg, providing stability while making drinks.
- Precision: High repeatability (±0.1 mm), ensuring accuracy in handling ingredients.

End Effector:

- (Modified) Robotiq 2F-140 Gripper: Adaptive gripper with adjustable grip force (10 N to 125 N), suitable for safely handling cups without slippage, attached to the link 6 of UR 10E via a serial connection.

Sensors and Actuators:

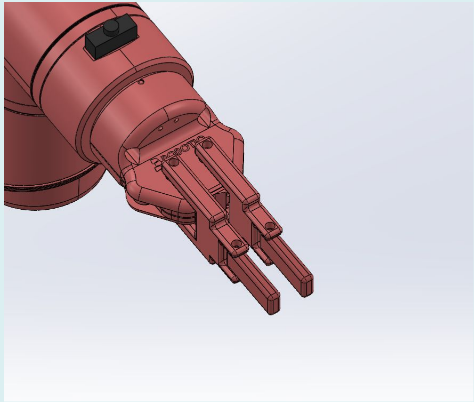
- Intel RealSense D435 Camera: RGBD camera to detect cup presence and gather workspace information.

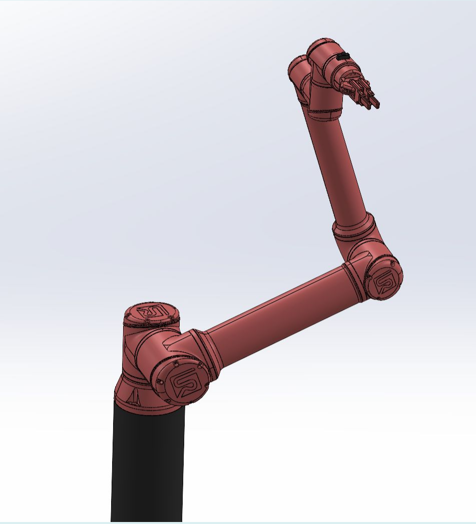


CAD Modeling Approach

Building the Assembly

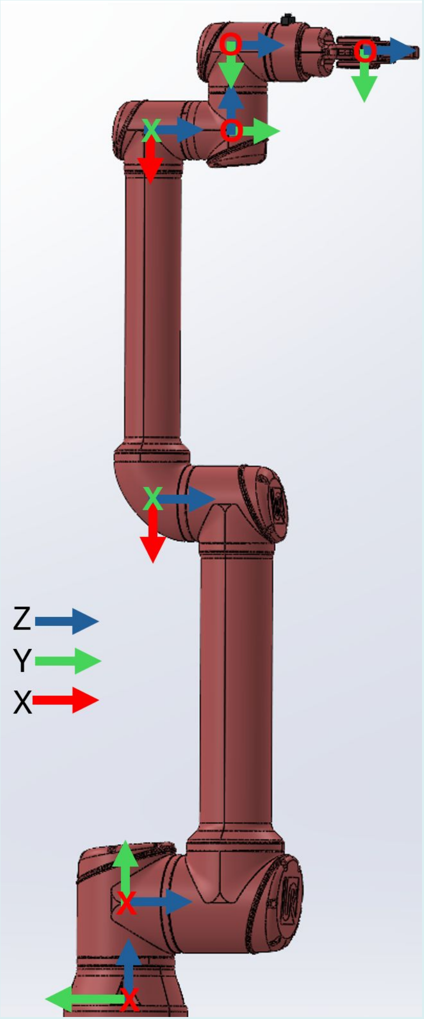
- CAD files are available for free download as a student; however, modifications were needed
- Rebuilt the UR10e assembly in Solidworks to use the URDF Exporter Tool
- The 2F-140 gripper is a parallel mechanism, so it was modified to become a serial gripper
- The full assembly (arm + gripper) was rebuilt in Solidworks and exported for use in the project
- Converted the URDF to a ROS 2 compatible version





DH Parameters

DH Table	theta (rad)	a (m)	d (m)	alpha (rad)	mass (kg)
Joint 1	θ1	0	0.1807	$\pi/2$	7.4
Joint 2	θ2	-0.6127	0	0	13.1
Joint 3	θ3	-0.57155	0	0	4.0
Joint 4	θ4	0	0.1741	$\pi/2$	2.1
Joint 5	θ5	0	0.1199	$-\pi/2$	2.0
Joint 6	θ6	0	0.1166	0	0.6



Workspace Study

Workspace Definition:

- The robot’s workspace is the 3D volume within which it can operate, defined by its reach and mobility.

Robot Reach:

- UR10e Arm: 1300 mm reach allows access to key areas such as coffee dispensers, cup holders, and serving counters.
- 6 Degrees of Freedom (DOF) enable interaction with objects at various angles and heights.

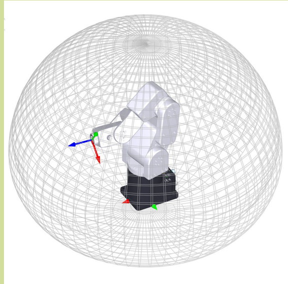


Image from Mecademic Industrial Robotics: https://mecademic.com/insights/academic-tutorials/workspace-six-axis-industrial-robot-arm/

Workspace Volume:

- The robot can cover an approximate 9.2 cubic meters (m^3) of space, assuming a spherical workspace with a radius of 1300 mm.

Reachability Analysis:

- The robot can reach within a circular workspace with a 1300 mm radius.

Forward Kinematics

- Used the DH table to compute individual transformation matrices between adjacent joints.

- Multiplied the transformation matrices to obtain the final transformation matrix and extracted the end effector position (x, y, z) from the first three rows of the final column (T[3,0], T[3,1], T[3,2]).

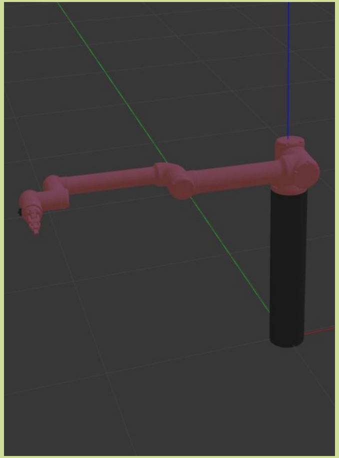
$$T_0^6 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4 \cdot T_4^5 \cdot T_5^6$$

```
Final Transformation Matrix:
 Matrix([[-(sin(theta_1)*sin(theta_5) - cos(theta_1)*cos(theta_5)*cos(theta_2 + theta_3 + theta_4))*cos(theta_6) + sin(theta_6)*sin(theta_2 + theta_3 + theta_4)*cos(theta_1), (sin(theta_1)*sin(theta_5) -
 cos(theta_1)*cos(theta_5)*cos(theta_2 + theta_3 + theta_4))*sin(theta_6) + sin(theta_2 + theta_3 + theta_4)*cos(theta_1)*cos(theta_6), sin(theta_1)*cos(theta_5) + sin(theta_5)*cos(theta_1)*cos(theta_2 +
 theta_3 + theta_4), 0.11655*sin(theta_1)*cos(theta_5) - 0.17415*sin(theta_1) + 0.6127*sin(theta_2)*cos(theta_1) + 0.11655*sin(theta_5)*cos(theta_1)*cos(theta_2 + theta_3 + theta_4) + 0.57155*sin(theta_2
 + theta_3)*cos(theta_1) + 0.11985*sin(theta_2 + theta_3 + theta_4)*cos(theta_1)], [(sin(theta_1)*cos(theta_5)*cos(theta_2 + theta_3 + theta_4) + sin(theta_5)*cos(theta_1))*cos(theta_6) + sin(theta_1)*si
n(theta_6)*sin(theta_2 + theta_3 + theta_4), -(sin(theta_1)*cos(theta_5)*cos(theta_2 + theta_3 + theta_4) + sin(theta_5)*cos(theta_1))*sin(theta_6) + sin(theta_1)*sin(theta_2 + theta_3 + theta_4)*cos(the
ta_6), sin(theta_1)*sin(theta_5)*cos(theta_2 + theta_3 + theta_4) - cos(theta_1)*cos(theta_5), 0.6127*sin(theta_1)*sin(theta_2) + 0.11655*sin(theta_1)*sin(theta_5)*cos(theta_2 + theta_3 + theta_4) + 0.57
155*sin(theta_1)*sin(theta_2 + theta_3) + 0.11985*sin(theta_1)*sin(theta_2 + theta_3 + theta_4) - 0.11655*cos(theta_1)*cos(theta_5) + 0.17415*cos(theta_1)], [sin(theta_6)*cos(theta_2 + theta_3 + theta_4)
 - sin(theta_2 + theta_3 + theta_4)*cos(theta_5)*cos(theta_6), sin(theta_6)*sin(theta_2 + theta_3 + theta_4)*cos(theta_5) + cos(theta_6)*cos(theta_2 + theta_3 + theta_4), -sin(theta_5)*sin(theta_2 + thet
a_3 + theta_4), -0.11655*sin(theta_5)*sin(theta_2 + theta_3 + theta_4) + 0.6127*cos(theta_2) + 0.57155*cos(theta_2 + theta_3) + 0.11985*cos(theta_2 + theta_3 + theta_4) + 0.1807], [0, 0, 0, 1]])
```

Forward Kinematics Validation

Test Pose 1:

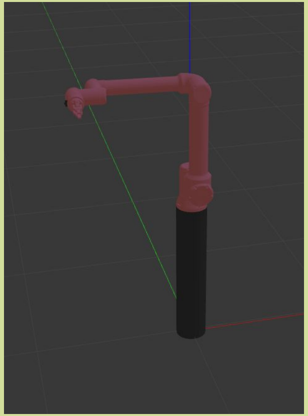
Joint 1 - 0.0
Joint 2 - pi/2
Joint 3 - 0.0
Joint 4 - 0.0
Joint 5 - 0.0
Joint 6 - 0.0



End Effector Position: (1.30410000000000, 0.0576000000000000, 0.180700000000000)

Test Pose 2:

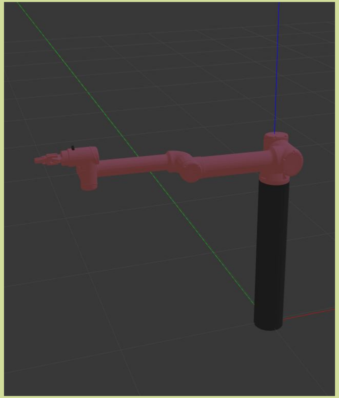
Joint 1 - 0.0
Joint 2 - 0.0
Joint 3 - pi/2
Joint 4 - 0.0
Joint 5 - 0.0
Joint 6 - 0.0



End Effector Position: (0.691400000000000, 0.0576000000000000, 0.793400000000000)

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



End Effector Position: (1.06770000000000, 0.174150000000000, 0.300550000000000)

Inverse Kinematics

Purpose: Compute joint angles required to achieve a given end-effector position and orientation.

Mathematical Approach:

- Use **Jacobian Matrix** to relate end-effector velocities to joint velocities.
- Solve for joint angles using **Inverse Velocity Kinematics** based on desired end-effector trajectories.

Method:

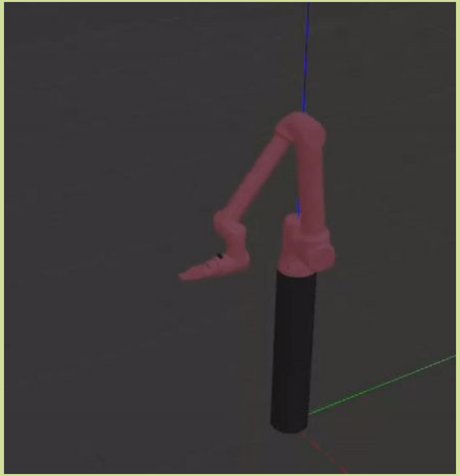
- **Jacobian Inversion**: Use the pseudo-inverse of the Jacobian matrix to calculate joint velocities.
- **Pre-Defined Trajectory:** Joint angles are computed to follow a pre-defined path to move between workstations

Challenges:

- **Singularities**: Avoid robot configurations where the Jacobian becomes non-invertible.
- **Multiple Solutions**: Handling situations where more than one set of joint angles can reach the same end-effector position (e.g., elbow-up vs. elbow-down).

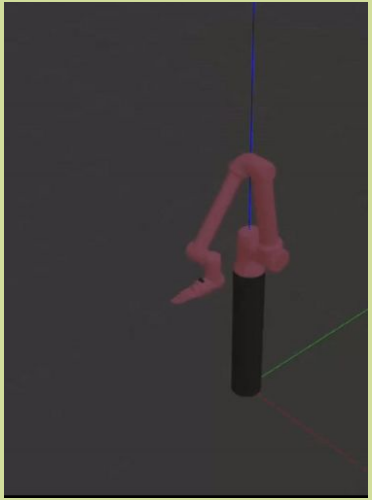
Inverse Kinematics Validation

Test Pose 1:



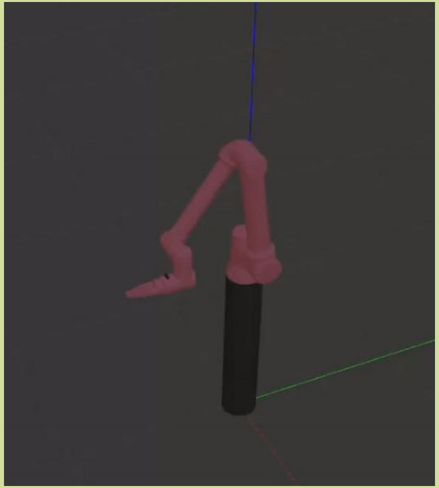
```
# Move End Effector in X Direction
x_vals = Matrix([0.50, 0.0, 0.0 , 0.0, 0.0, 0.0])
```

Test Pose 2:



```
# Move End Effector in Y Direction
x_vals = Matrix([0.0, 0.50, 0.0 , 0.0, 0.0, 0.0])
```

Test Pose 3:

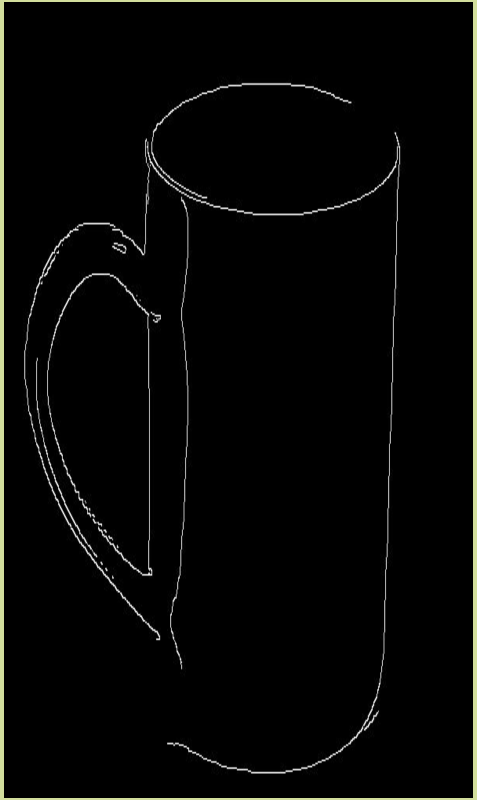


```
# Move End Effector in Z Direction
x_vals = Matrix([0.0, 0.0, 0.50 , 0.0, 0.0, 0.0])
```

Cup Detection

- **Efficient and Lightweight**: Classical computer vision (contour detection) avoids the computational overhead of deep learning models like YOLO or Mask R-CNN.

- **Shape-Agnostic**: Contour detection effectively identifies cups of various shapes.



Assumptions

- Assumptions:
  - No real liquid present during simulation
  - Dispensers, cups, and tables, located in fixed, predefined locations
  - Uniformly sized, lightweight, and rigid coffee cups
  - No obstacles

Robot Control Method

- Open-Loop Control: The robot arm follows a predefined trajectory
- Desired End-Effector Velocity: Computed based on the fixed task duration.
- Jacobian Control: Inverse velocity kinematics used to calculate joint angles for the robot trajectory.
- Gravity Compensation: Joint torques are calculated using the gravity matrix to counteract gravitational effects.

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$$

q_dot is the joint velocity vector.
J is the Jacobian matrix.
x_dot is the end effector velocity

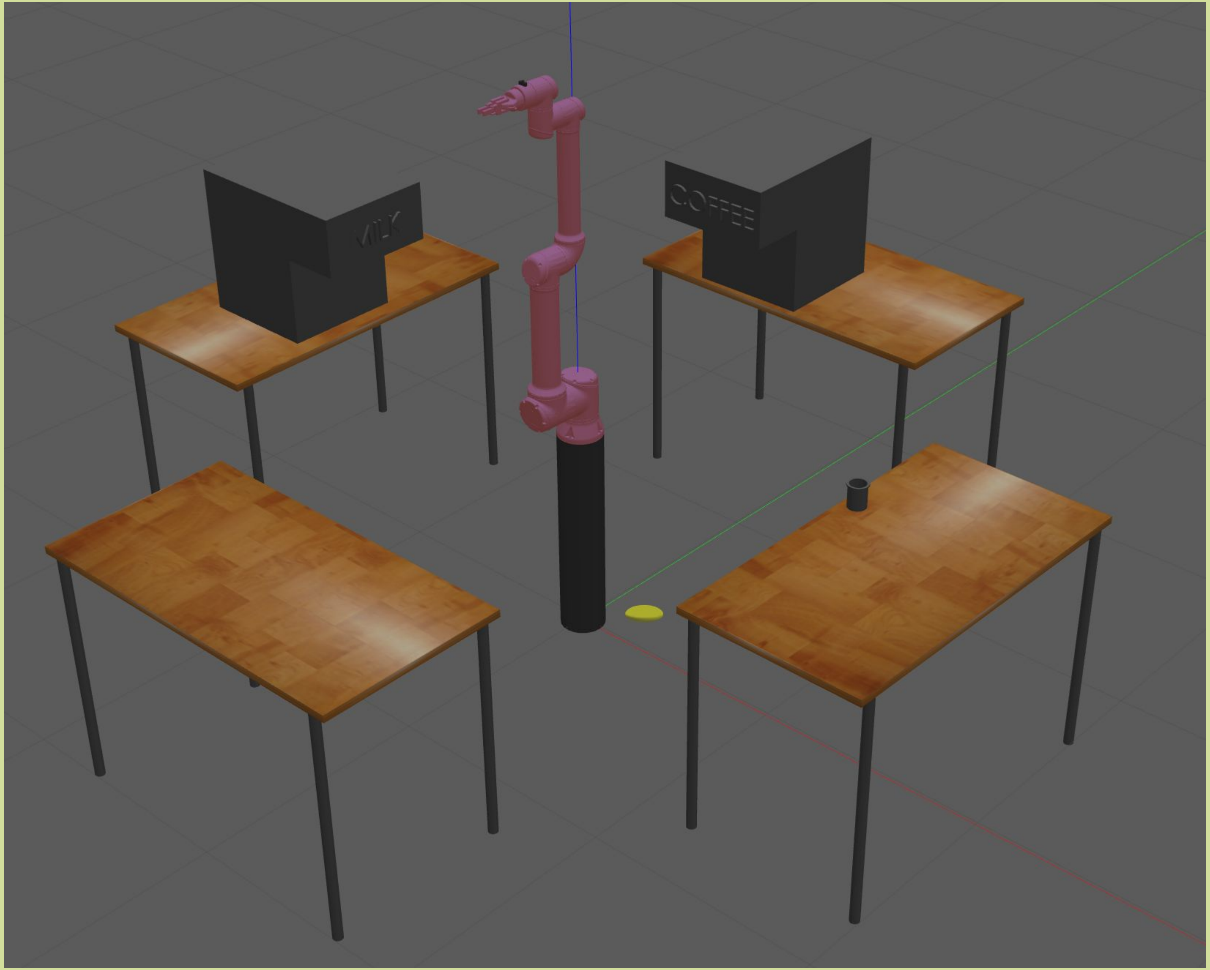
Gripper Control method

- Enables the Robotic arm to pick, hold, and release the cup at specific points during the coffee-making process.
- The gripper publisher sends position commands to the gripper’s joints to move the fingers.
- The commands are published to the topic /gripper_position_controller/commands.

Gazebo World

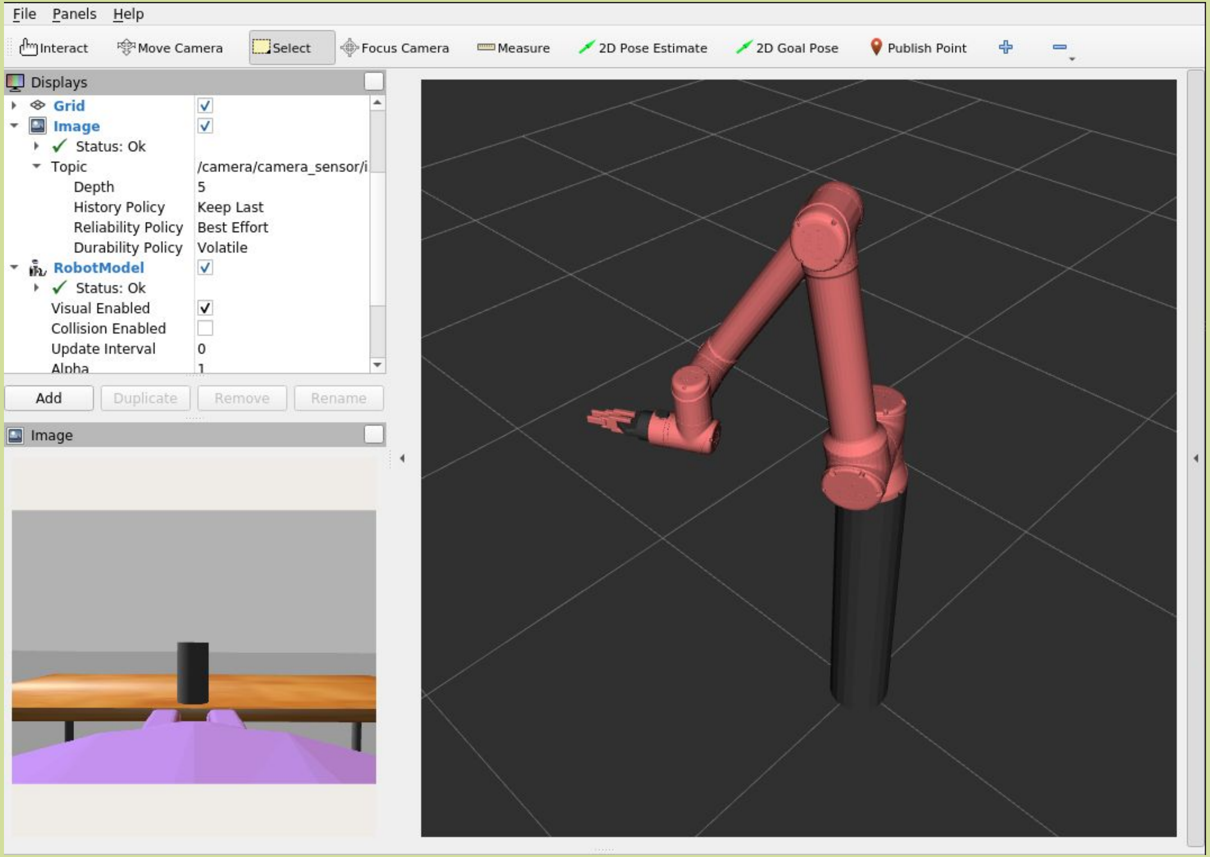
- Consists of 4 main stations:
  - Serving Table
  - Cup Selection Table
  - Coffee Dispensing Table
  - Milk Dispensing Table

- The robot sits on a pedestal in the center of the world, allowing easy access to all stations



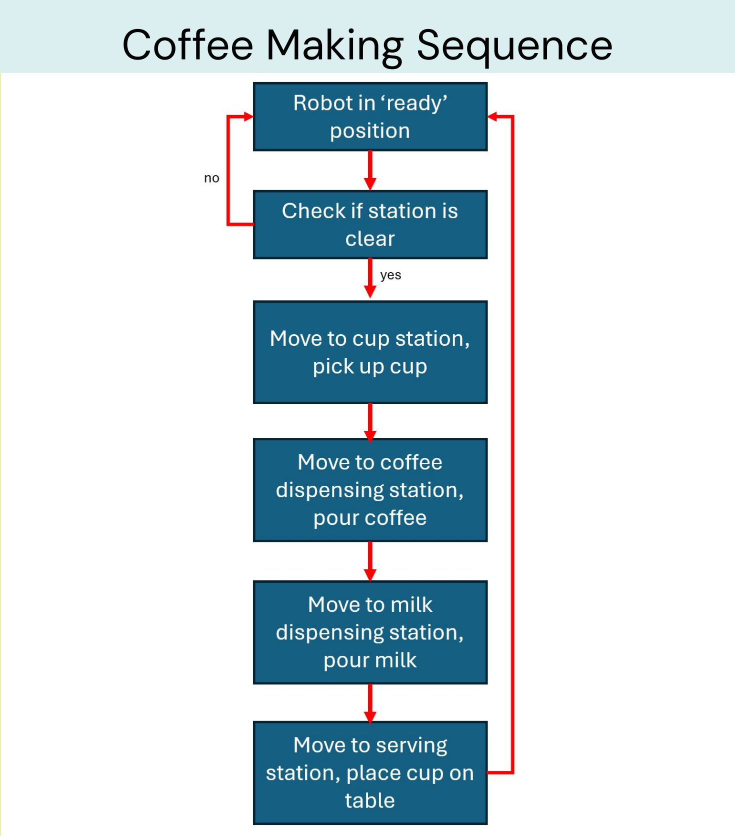
Rviz Simulation

- Visualizes the UR10e poses throughout the trajectory
- Depicts camera view at the end effector used for cup detection

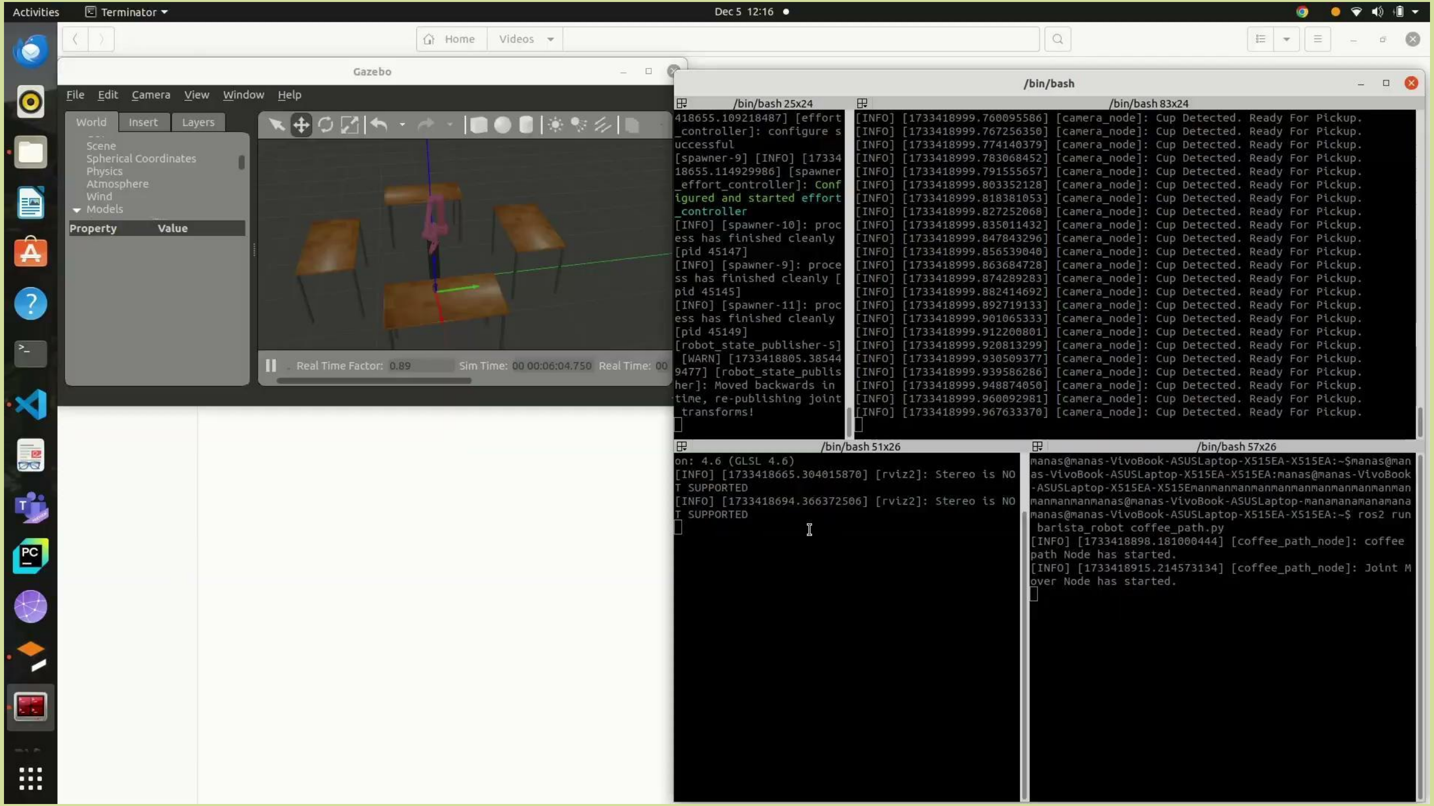


Approach to Simulation

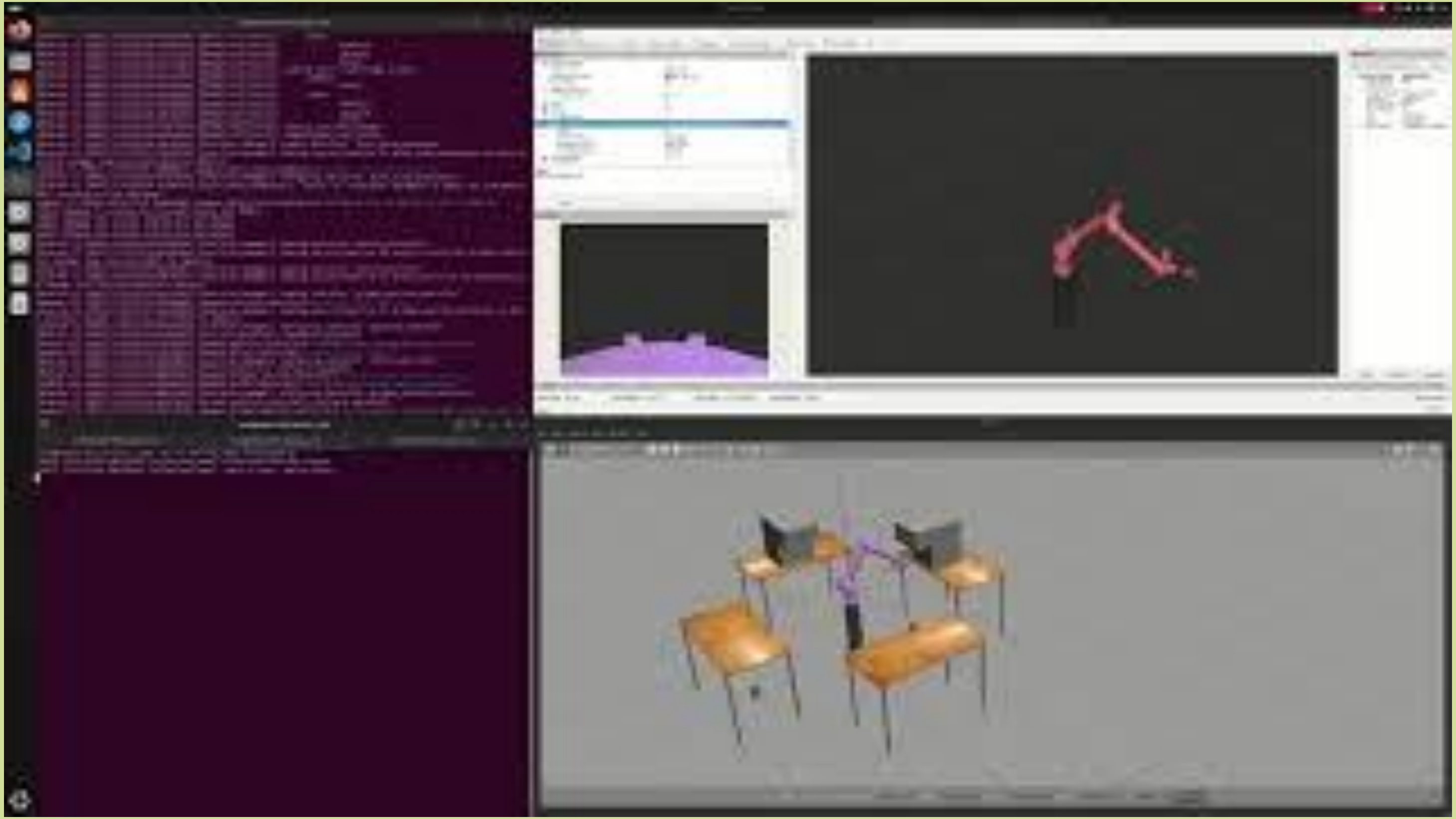
1. Robot spawns in home position
2. Move to ‘ready’ position
3. Begin coffee making sequence



Cup Detection Video

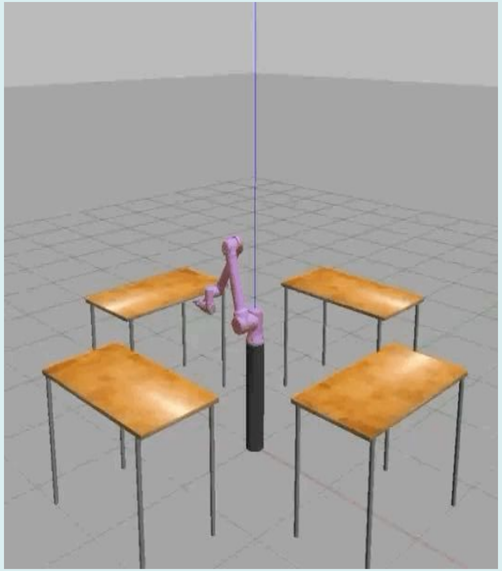


Simulation Video



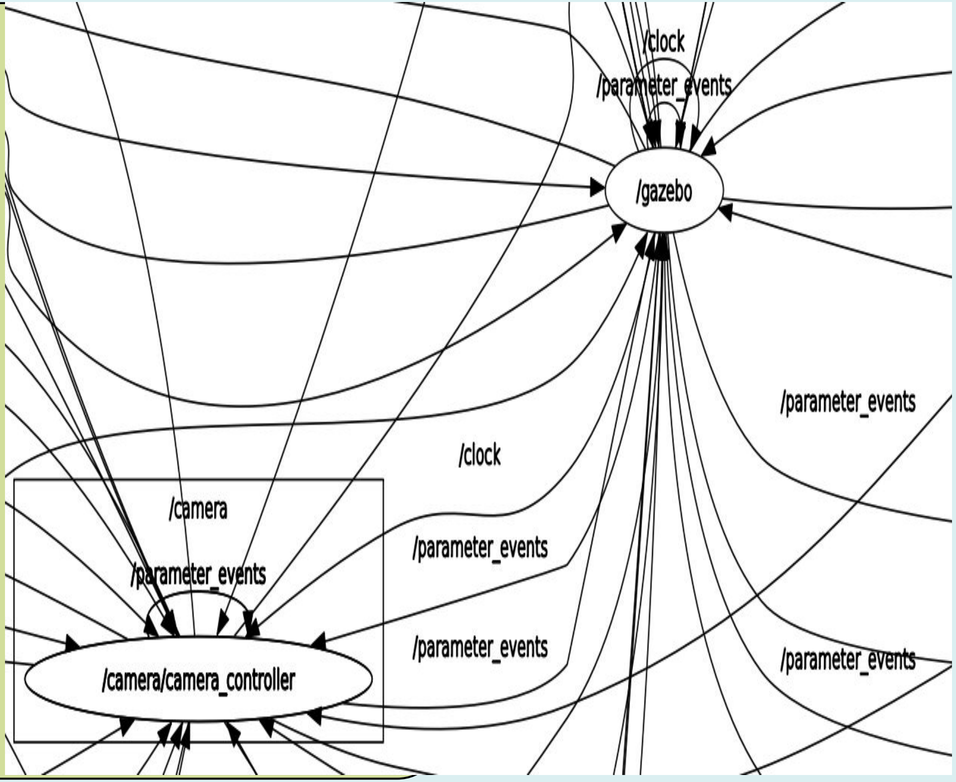
Problems

- **Robot arm breaking**
  - Joints separating during movements
  - Gravity creating additional challenges for accurate end effector positioning
- **Gripper breaking**
  - Oscillations in the link growing uncontrollably
- **Singularities**
  - Robot is uncontrollable in certain configurations
  - Robot arm moving but in an undesirable configuration
- **Custom world objects**
  - Custom objects not being located when sharing the custom world with teammates



Lessons Learned

- **Robot Home Position**: Configure within the workspace for optimal operation.
- **RViz for Visualization**: Offers better insights than echoing messages, especially for vision tasks.
- **rqt_graph**: Intuitively visualize communication between ROS nodes.



Conclusions

Successful Simulation:

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

Technological Integration:

- Demonstrated the integration of robotic arm , sensors, and a custom gripper and developed custom control algorithms to automate coffee preparation.

Benefits of Automation:

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

Future Work

- Specialized coffee orders (extra milk, no milk, etc.) automatically updating path
- Autonomous trajectory generation along with obstacle avoidance by the robot manipulator.
- PID Controller implementation for consistent trajectory tracking.

References

1 *ADAM | Robot Bartender | The Exciting Robotic Worker*. (n.d.).

https://www.richtechrobotics.com/

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

3 *ROS2 Tutorials*. (n.d.). https://docs.ros.org/en/galactic/Tutorials.html

THANK YOU

Grayson Gilbert
Shreya Kalyanaraman
Manas Desai
Kunj Golwala