RDKDC Final Project Report

Eva Yuqin Pan, Can Kocabalkanli, Mich Fredricks

Method 1: Inverse Kinematics

This involves solving the joint angles from target positions using DH-parameterization via the provided packages. Using DH parameters we can map from the SE(3) workspace to Q, the configuration space.

Workflow:

1. **Teach and record the start pose**
2. **Teach and record the end pose**
3. Calculate intermediate position, which is located in the center of the start and end location with an offset height of 35 cm, and find its inverse kinematics with Inv_Kin.m to find possible joint configurations corresponding to it. Then, the "best" configuration is selected to satisfy the condition where singularities are avoided and the distance to move is minimized
4. Move to intermediate position
5. Show intention by making the end-effector pointing to the start location
6. Apply Inv_Kin.m to find all possible joint configurations that would result in start pose
7. Pick the best one in terms of the least amount of movement (by checking the difference between the norms of q vectors) and also eliminate those that would result in collisions with the table.
8. Move to 10 cm above the start pose and then go down to the desired start pose
9. Apply Inv_Kin.m to find all possible joint configurations that would result in end pose
10. Pick the best one in terms of the least amount of movement, eliminate those that would result in collisions with the table.
11. Move to 10 cm above the end pose and then go down to the desired end pose
12. Move to the joint configuration previously selected for the intermediate step.

Video of the UR5 in action:
https://drive.google.com/open?id=1RHmWxiaA4_gB_2r1PsPorBCTaI780qe

Method 2: Rate Control with Differential Kinematics

Using the inverse Jacobian and finding a small differential Cartesian offset, we find the joint space velocity to produce increments in the correct direction.

Workflow:

1. **Teach and record the start pose**
2. **Teach and record the end pose**
3. Calculate intermediate position
4. In discrete time steps, use resolve rate control to move towards intermediate step reducing position error each time.
5. Stop when error is below threshold, indicating we are at intermediate pose
6. Show intention by pointing the end-effector towards destination
7. In discrete time steps, use resolve rate control to move towards 10 cm above the start pose reducing position error each time.

8. Stop when error is below threshold, indicating we are at the desired pose
9. Repeat 7 - 8 to get to the start pose
10. In discrete time steps, use resolve rate control to move towards 10 cm above the end pose reducing position error each time.
11. Stop when error is below threshold, indicating we are at the desired pose
12. Repeat 10 - 11 to get to the end pose
13. Move to intermediate pose using resolve rate control as in step 4.

Video of the UR5 in action:
https://drive.google.com/open?id=1fPsrddCVyzQDEE_MGf_Txd5kSav1SgAR

Method 3: Gradient-Based Control
Similar to the previous method but uses transpose of the Jacobian instead of the inverse.
Workflow:
1. **Teach and record the start pose**
2. **Teach and record the end pose**
3. Calculate intermediate position.
4. In discrete time steps, use gradient based control to move towards intermediate step reducing position error each time.
5. Stop when error is below threshold, indicating we are at intermediate pose
6. Show intention by pointing the end-effector towards destination
7. In discrete time steps, use gradient based control to move towards 10 cm above the start pose reducing position error each time.
8. Stop when error is below threshold, indicating we are at the desired pose
9. Repeat 7 - 8 to get to the start pose
10. In discrete time steps, use gradient based control to move towards 10 cm above the end pose reducing position error each time.
11. Stop when error is below threshold, indicating we are at the desired pose
12. Repeat 10 - 11 to get to the end pose
13. Move to intermediate pose using gradient based control as in step 4.

Video of the UR5 in action:
https://drive.google.com/file/d/1s-FHFzroWgKfPOLhrvO9Cnr9DSlacrg0/view?usp=sharing

<u>Figures of the Steps Common to All Three Control Methods</u>

Figure 1: Intermediate Pose

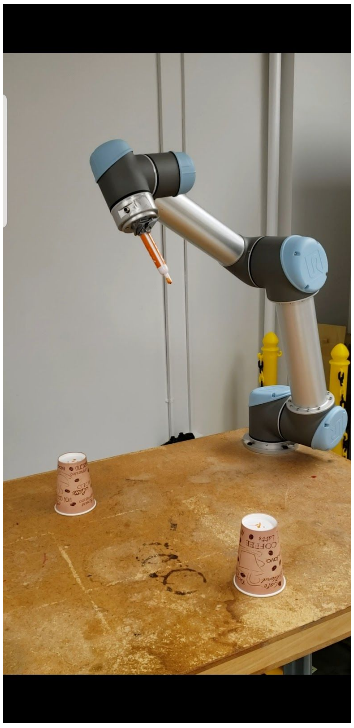

Figure 2: Show Intention

Figure 3: 10 cm Above Start

Figure 4: Star Pose

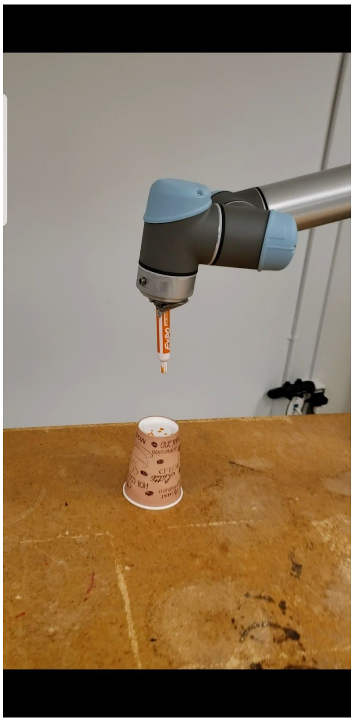

Figure 5: 10 cm Above End

Figure 6: End Pose

Extra Credit: Two Towers

This task involves training the UR5 by recording the locations of the centers of two cups, denoted by blue dots in Figure 7 below. The UR5 then moves the end effector in a zigzag trajectory around the cups, navigating through them without touching.

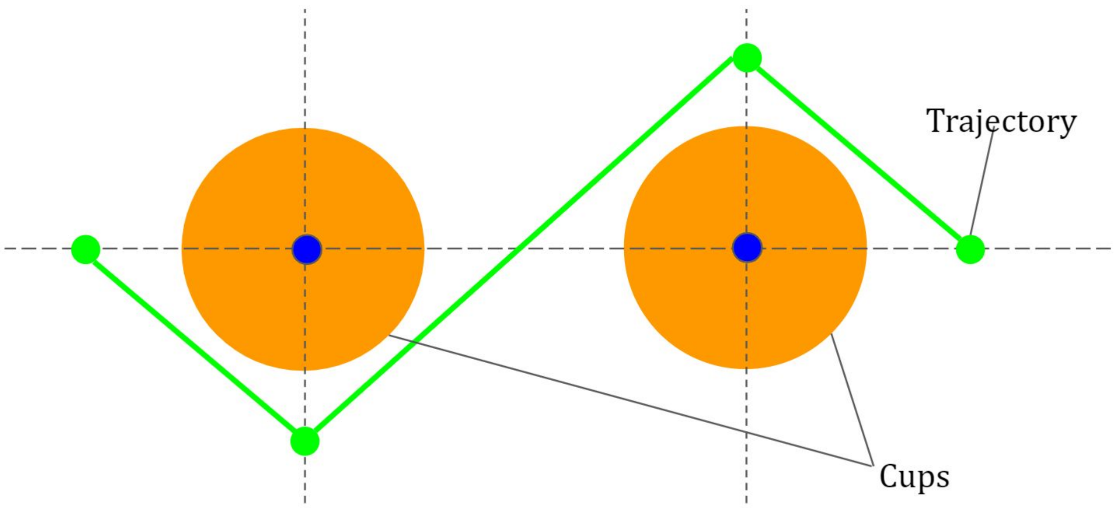

Figure 7: Extra Credit Task Schematic

Video of UR5 navigating through the two cups:
https://drive.google.com/open?id=1blvlu0eJERyeKlH6nreWCb6xab4fZwGz

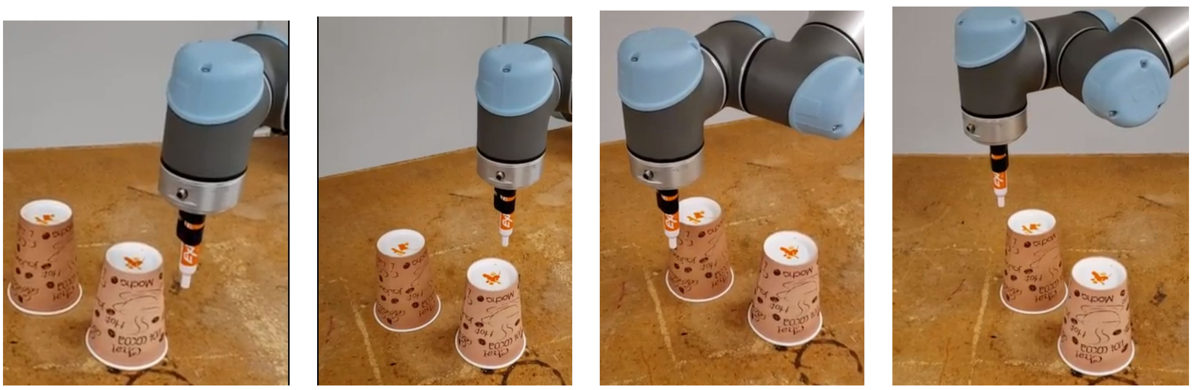

Figure 8: The four green points in the UR5 trajectory in real life

Results:

Inverse Kinematics:

- As expected this was the quickest method as it used UR5's move_joints function (that takes time duration as an input argument) after solving the inverse kinematics. It took a total of 53.80 seconds for the tested path. It is also the smoothest and takes the most efficient path as it goes directly and accurately to the target position, touching the marker exactly on the previous mark left when the UR5 was trained.

Resolved Rate Control:

- This method was the second quickest, taking a total of 93.47 seconds for the same tested path. It had relatively smooth movements in between steps compared to gradient-based control but the pauses during steps were quite clear and abrupt. The efficiency seems similar to that of inverse kinematics with the path being relatively direct. It accurately touched the marker on the previous mark left when the UR5 was trained, which indicates that the error threshold we picked for the algorithm was adequate.

Gradient-Based Control:

- This method took the longest time with a total of 315.31 seconds for the tested path. It was the least smooth with some erratic movements, especially at beginning where the movements go back and forth rather violently. This is admittedly inefficient and is the reason behind the long time. However, overall accuracy is fairly reasonable and ends up within the rim of the bottom of cup.

Work Distribution

We reused the methods we have written for Lab 3 for the most part, and all three worked on putting them together, tweaking parameters, and handling exceptions.