IVR Report

S1875587 focused on part 2 (Vision) of the coursework and S1828233 focused on part 3 (Control) of the coursework. We both helped each other in our respective parts. Part 4 was worked on by both members.

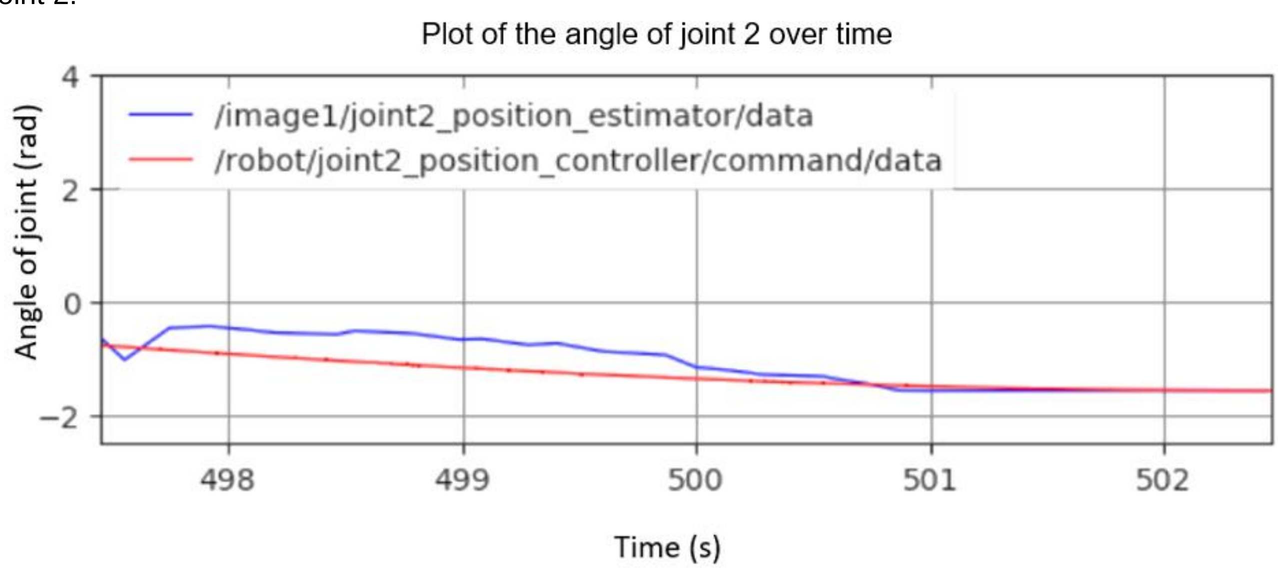
Github link: https://github.com/Ziemniok352/IVR-Assignment/

Joint State Estimation 2.1:
Algorithm

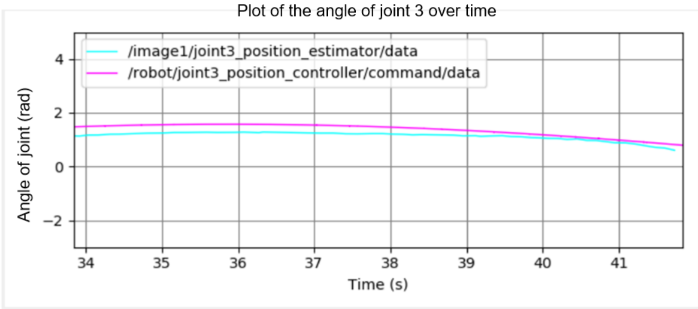
1) Receive the raw image from each camera.
2) Mask each image to show only red, and repeat for each joint color (red, blue, yellow, and green).
3) Dilate the masked images to smooth them out and make detection easier.
4) Get the pixel coordinates of the center of each blob using moments: x = m10/m00 and y = m01/m00.
5) Change these coordinates to meter scale using the length of a link as reference.
6) Change coordinate reference frame to use the yellow joint as origin: $x = (x_{yellow} – x) * -1$, $y = (y_{yellow} – y)$.
7) Combine these two sets of 2d coordinates for each joint into one set of 3d coordinates: Joint one is fixed, so the positions of the yellow and blue joints are also fixed at [0,0,0] and [0,0,2] respectively. For the green and red joints, in the general case, the x-coordinate is equal to the x-coordinate from the second image, the y-coordinate the x-coordinate from the first image, and the z-coordinate is equal to the mean between the two y-coordinates.
    a. If the joint is occluded from one camera view enough that it could not be detected, coordinates which would be taken from that camera view are instead replaced with the coordinates from the previous joint. The z-coordinate is estimated as equal to the y-coordinate from whichever camera still sees the joint.
8) Calculate an estimate for the angle of each joint by using a least squares regression between the bounds of -pi/2 and pi/2 with each joint's elementary rotation matrix.

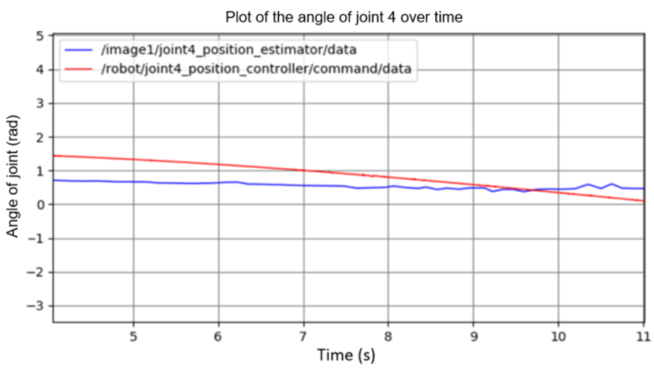
Plots

Joint 2:



IVR Report



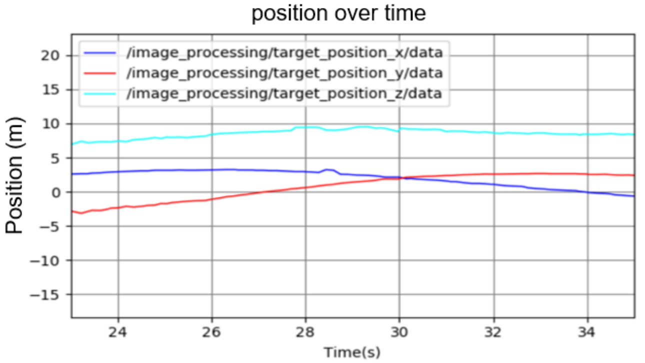


2.2: Target Detection
Algorithm and Sources of Error

1) Receive the raw image from each camera.
2) Mask each image to show only orange.
3) Dilate the masked images for easier detection of shapes.
4) Conduct Chamfer matching on each image using a circle as a template.
5) Calculate the coordinates for the point in each image which matches the template the best as was done previously for the joints (see steps 4 through 7).

There are two main sources of error in this target detection method: First, it does very little to account for when the target is occluded, because unlike for the joints, there is no way to guess what the target will be occluded behind. Second, at certain angles, the silhouette of the orange box can be mistaken by the Chamfer matching program for a sphere.

Plot of the spherical target's position over time



Forward Kinematics 3.1

Forward Kinematics:

IVR Report

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} 3.5cos\theta_1 cos\theta_2 cos\theta_3 + 3.5sin\theta_1 sin\theta_3 + 3cos\theta_4(cos\theta_1 cos\theta_2 cos\theta_3 + sin\theta_1 sin\theta_3) - 3cos\theta_1 sin\theta_2 sin\theta_4 \\ 3.5sin\theta_1 cos\theta_2 cos\theta_3 + 3cos\theta_4(sin\theta_1 cos\theta_2 cos\theta_3 - cos\theta_1 sin\theta_3) - 3sin\theta_1 sin\theta_2 sin\theta_4 - 3.5cos\theta_1 sin\theta_3 \\ 3.5sin\theta_2 cos\theta_3 + 3sin\theta_2 cos\theta_3 cos\theta_4 + 3cos\theta_2 sin\theta_4 + 2.5 \end{bmatrix}$$

Joint angles ($\theta_1, \theta_2, \theta_3, \theta_4$)	**Est. end-effector from image**	**Est. end-effector from FK**
(0.1,0.1,0.1,0.1)	x: -15.674, y: -15.932, z: 11.548	x: 0.738, y: -0.873, z: 8.890
(0.5,-0.5,0.5,-0.5)	x: -16.182, y: -21.883,z: 14.575	x: 0.738, y: 4.782, z: 6.534
(2.7, 0.3, 1.3, 0.9)	x: 3.073, y: -20.656, z: 18.133	x: -3.533, y: 4.622, z: 3.177
(1, -0.8, 1.2, 1.4)	x: 18.986, y: 19.598, z: -15.594	x: 2.875, y: 2.595, z: 5.633
(pi, pi/2, pi/2, pi/2)	x: 12.926, y: 15.830, z: -23.747	x: -3.500, y: 0.000, z: -0.500
(-pi, pi/2, -pi/2, pi/2)	x: 20.431, y: 19.149, z: -24.309	x: 3.500, y: 0.000, z: -0.500
(-1.6, 0.4, 1.1, -1.4)	x: 17.304, y: 21.344, z: -16.169	x: 1.909, y: -3.631, z: 5.327
(0.25, -0.35, 0.8, -1.1)	x: 10.544, y: 14.366, z: -16.286	x: 2.470, y: 4.421, z: 4.764
(-0.3, 1.5, 0.2, 1.5)	x: 20.448, y: 22.562, z: -24.966	x: -0.430, y: -3.887, z: -0.228
(-2.15, -0.6, -1.5, - 0.3)	x: 24.179, y: 11.752, z: -20.022	x: 4.301, y: 4.775, z: 2.371

As can be seen from the table above, there is a huge difference between the two columns. Since the FK is most likely correct as it was verified using rostopic and the angle results are also relatively accurate, the error in the image column when detecting the end-effector from the image is most likely a calculation error and not due to the vision detection aspect itself.

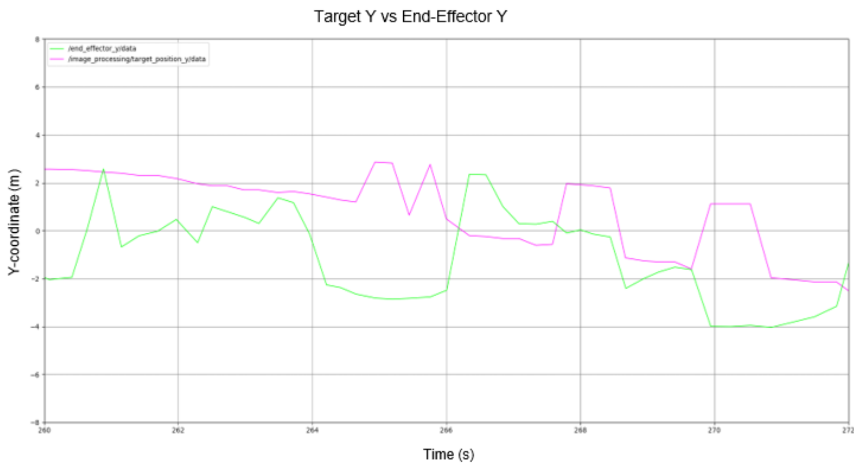
Closed-loop Control 3.2

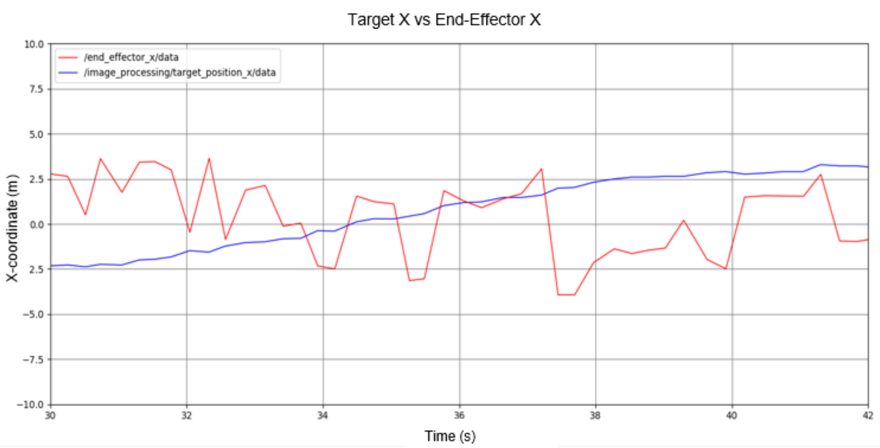
Jacobian:

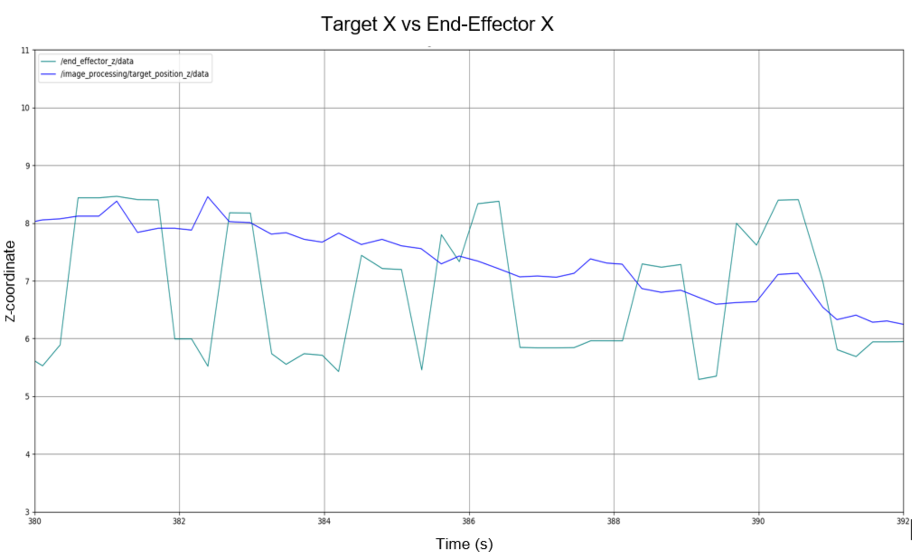
$\mathrm{J} = [[3.5c_1c_3s_2 + 3c_1c_2s_4 + 3(c_1c_3s_2 - s_1s_3)c_4 - 3.5s_1s_3, 3c_2c_3c_4s_1 + 3.5c_2c_3s_1 - 3s_1s_2s_4, -3.5s_1s_2s_3 + 3.5c_1c_3 - 3(s_1s_2s_3 - c_1c_3)c_4, 3c_2c_4s_1 - 3(c_3s_1s_2 + c_1s_3)s_4], [3.5c_3s_1s_2 + 3c_2s_1s_4 + 3(c_3s_1s_2 + c_1s_3)c_4 + 3.5c_1s_3, -3c_1c_2c_3c_4 - 3.5c_1c_2c_3 + 3c_1s_2s_4, 3.5c_1s_2s_3 + 3(c_1s_2s_3 + c_3s_1)c_4 + 3.5c_3s_1, -3c_1c_2c_4 + 3(c_1c_3s_2 - s_1s_3)s_4], [0, -3c_3c_4s_2 - 3.5c_3s_2 - 3c_2s_4, -3c_2c_4s_3 - 3.5c_2s_3, -3c_2c_3s_4 - 3c_4s_2]]$

where $\mathrm{s}_1, s_2, s_3, s_4 = sin\theta_1, sin\theta_2, sin\theta_3, sin\theta_4,$
$c_1, c_2, c_3, c_4 = cos\theta_1, cos\theta_2, cos\theta_3, cos\theta_4$

3 graphs comparing x,y,z position of end effector with x,y,z position of target:







4.2: Null-Space Control

First, in order to avoid the box, we must be able to track the box. The algorithm for this is exactly the same as the algorithm which calculates the coordinates for the sphere, except for the template used in the Chamfer matching. Instead of a template of a sphere, it instead uses a template of a box—a

rectangle. The control part would try to use the box's position and use a cost function which minimises the distance of the robot’s end effector to the orange box.
We were not able to make either part work correctly. Although, when plotting the boxes actual position, against the end-effector and target, it can be seen that for the most part the robot does avoid the target.