**IVR COURSEWORK**

**Amy Rafferty s1817812 – Alasdair Forbes s1849401**

**Contributions:**

* 2.1: Code – Alasdair, Report – Amy
* 2.2: Code - Both, Report - ??

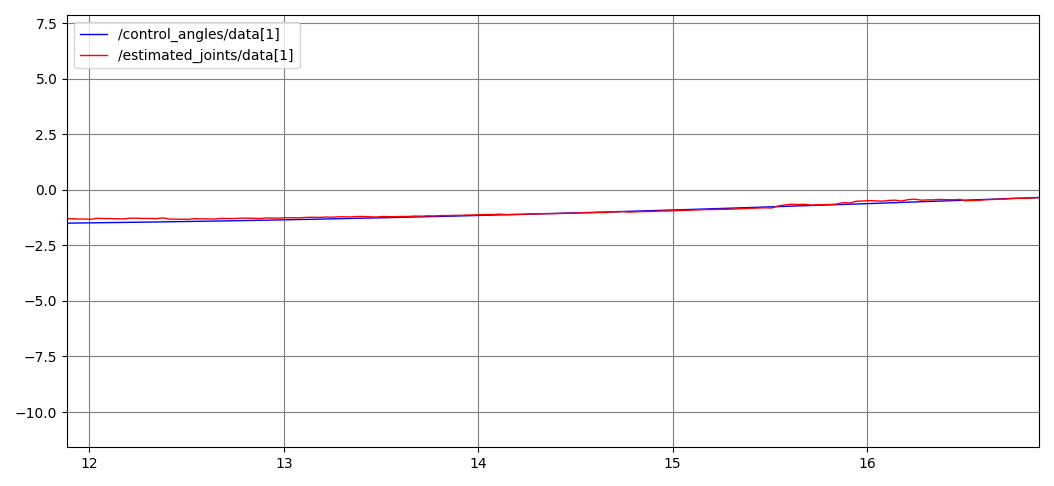
**Repository:** <https://github.com/aforbes23/IVR_assignment>

**Question 2.1**

Our algorithm first gets the coordinates of the centres of each joint and the end effector, by using blob detection for each joint colour (yellow, blue, green, red). During this step, the image is converted to HSV, as it tends to be a more reliable format than the default BGR for colour masks. Before any angles can be calculated, the coordinates of the blob centres must be converted from pixels to meters. The conversion factor between these metrics is found by dividing the length of a link in meters, as given in the coursework specification, by the distance between the centres of the blobs at each end of this link in the image. The coordinates of the centres are then converted to meters, with respect to the position of joint 1, which we take as the centre of the image (0,0).

This is repeated for both cameras, which gives the result of a set of coordinates for each blob for each image. The coordinates for image 2 are published by image2.py and subscribed to by image1.py, which is where our algorithm primarily operates. Then, for each blob, these two sets of coordinates are combined to find the 3d position of the blob (x, y, z). Image 1 provides the x coordinate, image 2 provides the y coordinate, and the z coordinate should be the same for each image, so can be taken from either, providing that it is a legal number (not NaN). Then, trigonometry is used to find the estimated angles for the joints, each calculated on the plane in which the joint operates. For joint 2, this is the y-z plane, as it rotates around the x axis. Similarly, for joint 3, which operates on the x-z plane, and rotates around the y axis. Joint 4 is slightly different, as it does not operate on an axes plane. Instead, we had to calculate the rotation matrix for the plane, then transform the direction vector for the link to get the required angle.

These estimated angles are then published. The actual angles of the joints are calculated and published by image2.py, in the form of a sinusoidal trajectory specified in the coursework specification. Here, we have graphs showing the differences between these actual angles (shown in blue), and our estimated angles (shown in red), as found by our algorithm, each over a 5 second period.

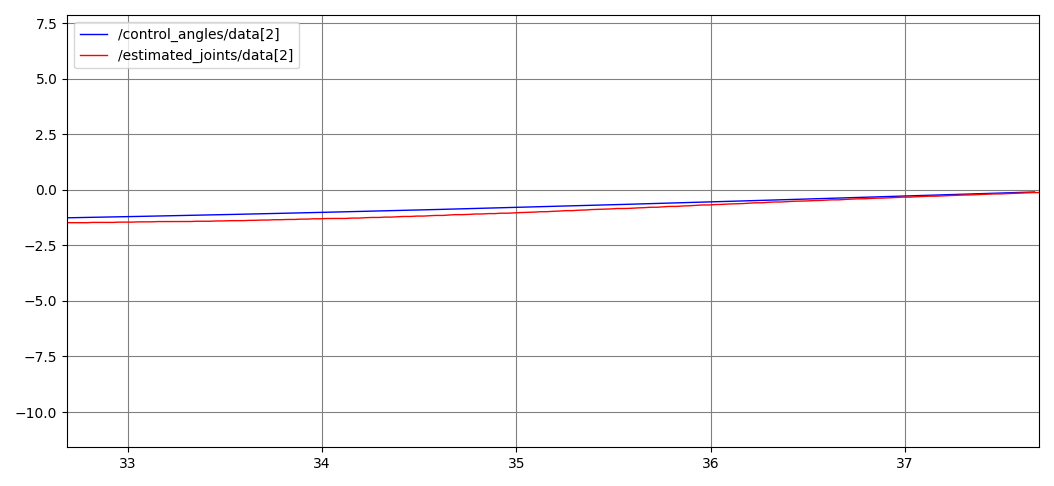


Time (s)

Angle (radians)

Actual and Estimated Trajectories for Joint 2

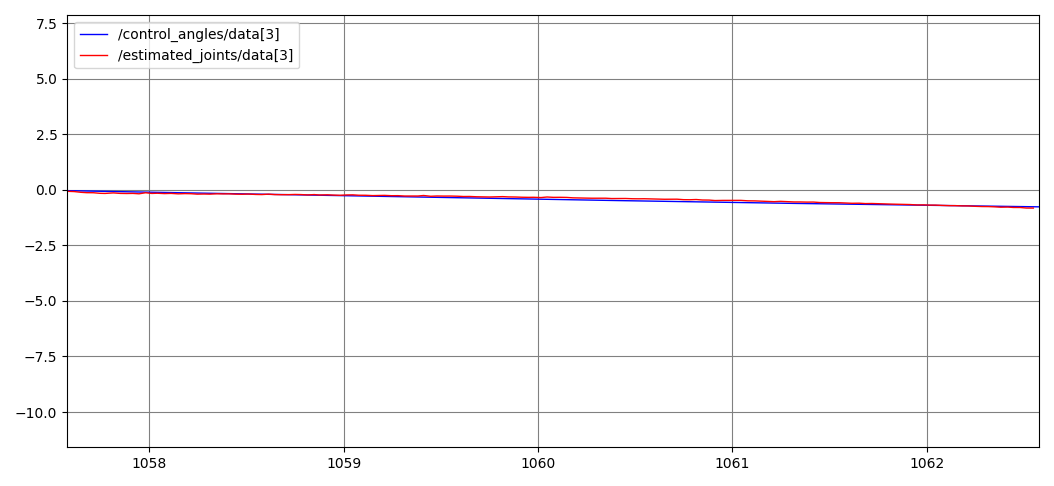
Actual and Estimated Trajectories for Joint 3



Time (s)

Angle (radians)

Angle (radians)



Time (s)

Actual and Estimated Trajectories for Joint 4

**Question 2.2**