1 Robot Vision

1.1 Joint State Estimation 1

1.1.1 Algorithm Overview

This section briefly describes the algorithm for estimating the robot joint angles. Two separate, supporting processes are invoked to capture the images from camera 1 (orthogonal to the y axis), camera 2 (orthogonal to the x axis). The images are published to image_topic1 and image_topic2. An additional process is setup to move robot joints 2, 3 and 4 in a sinusodial trajectory. The joints follow the following wavelengths. TD is the time delta in float seconds from the process beginning and the clock time which is determined by the **rospy.get_time()** method.

- Joint 1: $\frac{\pi}{2}\sin(\frac{pi}{15} * TD)$
- Joint 1: $\frac{\pi}{2}\sin(\frac{pi}{18} * TD)$
- Joint 1: $\frac{\pi}{2}\sin(\frac{pi}{18} * TD)$

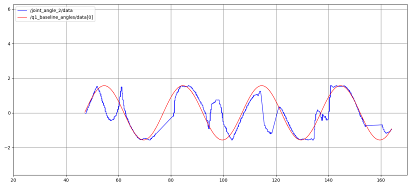
Once these processes are calculated the joint angle detection algorithm can be ran. The joint angle detection algorithm is based on the following algorithm:

```
procedure JOINT ANGLE DETECTION(camera_x, camera_y)
  Ensure: image_x = camera_x
  Ensure: image_y = camera_y
    blob_yellow_x, blob_blue_x, blob_red_x = blob_detection(image_x)
    blob_yellow_y, blob_blue_y, blob_red_y = blob_detection(image_y)
    if blob_detection_successful then
        distance_yellow_y = $3.2 * \sqrt(blob\_yellow\_y - blob\_blue\_y)^2$
        distance_yellow_x = $3.2 * \sqrt(blob\_yellow\_x - blob\_blue\_x)^2$
        distance_blue_x = $3.2 * \sqrt(blob\_blue\_x - blob\_red\_x)^2$
        distance_blue_y = $3.2 * \sqrt(blob\_blue\_y - blob\_red\_y)^2$
        distance_red_y = $2.8 * \sqrt(blob\_red\_y - blob\_blue\_y)^2$
        determine pixel to meter; each blob = $blob * distance$
        ja2 = $\tan(blob\_yellow\_y - blob\_blue\_y)$
        ja3 = $\tan(blob\_yellow\_x - blob\_blue\_x)$
        ja4 = $\tan(blob\_blue\_x - blob\_red\_x)$
        publish joint2, joint3, joint4
    else
        Write image to disk for analysis
        Print message joint angle could not be calculated
    end if
end procedure
```

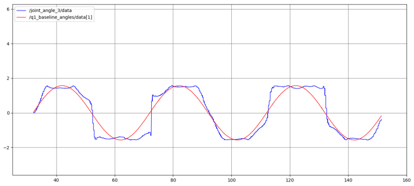
In working out the tangent the numpy atan2 library is utilized to ensure the correct angle quadrant is calculated. The absolute distance in y co-ordinates is also used rather than relative distance due to the fact that an upstream joint can, depending on the joint angles end in a lower position to a downstream joint. The blob detection is calculated using simple colour masking techniques to isolate the links, and then the center of each link is estimated via the moments of the image. For estimating changes to the joint angles the cameras facing the axis are used independently of one another, so changes to the joints in the y axis are tracked via the camera facing that plane and vice versa for camera x.

1.1.2 Results

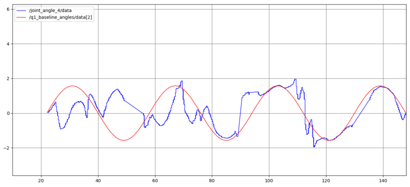
The algorithm was ran for 120 seconds with the recorded change in joint angle plotted against the estimated change in joint angle from the vision algorithm. Separate graphs are used to plot the deltas between the different joint angles and the vision algorithm estimations. The results are shown in the plots below:



(a) legend



(b) legend



(c) legend

1.1.3 Discussion