

# IVR Coursework

Matt Timmons-Brown & Neil Weidinger

December 4, 2020

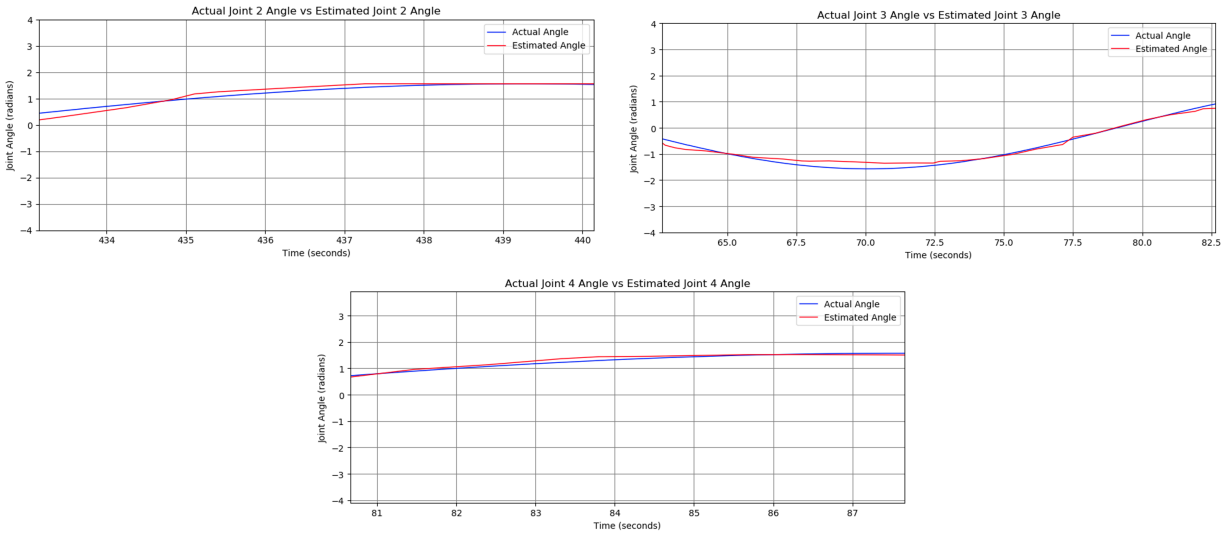
Neil and Matt worked on this coursework collaboratively - answering all questions and coding together. Thus there was an equal contribution from both members of the team.

Access the GitHub link to our code here: <https://github.com/the-raspberry-pi-guy/IVR-Assignment>

## 2 Robot Vision

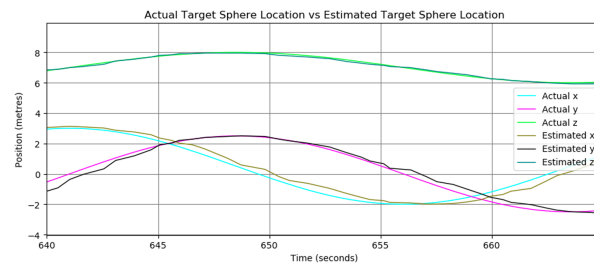
### 2.1 Joint State Estimation

*DESCRIBE ALGORITHM*



### 2.2 Target Detection

*DESCRIBE ALGORITHM & COMMENT ON SOURCES OF ERROR IN MEASUREMENTS*



### 3 Robot Control

#### 3.1 Forward Kinematics

$$\begin{bmatrix} 3s(\theta_1)s(\theta_2)c(\theta_3)c(\theta_4) + 3.5s(\theta_1)s(\theta_2)c(\theta_3) + 3s(\theta_1)s(\theta_4)c(\theta_2) + 3s(\theta_3)c(\theta_1)c(\theta_4) + 3.5s(\theta_3)c(\theta_1) \\ 3s(\theta_1)s(\theta_3)c(\theta_4) + 3.5s(\theta_1)s(\theta_3) - 3s(\theta_2)c(\theta_1)c(\theta_3)c(\theta_4) - 3.5s(\theta_2)c(\theta_1)c(\theta_3) - 3s(\theta_4)c(\theta_1)c(\theta_2) \\ -3s(\theta_2)s(\theta_4) + 3c(\theta_2)c(\theta_3)c(\theta_4) + 3.5c(\theta_2)c(\theta_3) + 2.5 \end{bmatrix}$$

Joint Angle <i>Joint 1,2,3,4 (rad)</i>	Estimated via FK <i>x,y,z (m)</i>	Estimated via Images <i>x,y,z (m)</i>
1,0.5,0.1,-1	0.47,0.31,8.18	0.33,0.33,8.76
-1,-1,-1,1	-1.52,4.15,6.12	-1.36,3.68,6.59
0.25,0.25,0.25,0.25	2.09, -1.79, 8.33	2.24,-2.06,9.16
1,1,0.5,0.5	6.05,-0.39,4.20	6.11,-0.59,4.64
-1,-0.5,-0.1,1	4.20,2.09,5.76	3.68,2.72,6.22
1,1,1,-1	3.14,3.10,6.12	2.54,3.72,6.77
-0.25,-0.25,-0.25,-0.25	-0.98,2.58,8.33	-0.92,2.43,8.65
-1,-1,-0.5,-0.5	2.88,5.34,4.20	2.13,6.33,4.97
$\pi, \pi/2, \pi/4, -0.1$	-4.58,4.59,2.80	-3.72,3.64,3.86
$-\pi, -\pi/2, -\pi/4, 0.1$	4.59,-4.58,2.80	6.07,-6.15,2.43

COMMENT ON ACCURACY

#### 3.2 Closed-Loop Control

$$A =$$

$$\begin{bmatrix} -3s(\theta_1)s(\theta_3)c(\theta_4) - 3.5s(\theta_1)s(\theta_3) + 3s(\theta_2)c(\theta_1)c(\theta_3)c(\theta_4) + 3.5s(\theta_2)c(\theta_1)c(\theta_3) + 3s(\theta_4)c(\theta_1)c(\theta_2) \\ 3s(\theta_1)s(\theta_2)c(\theta_3)c(\theta_4) + 3.5s(\theta_1)s(\theta_2)c(\theta_3) + 3s(\theta_1)s(\theta_4)c(\theta_2) + 3s(\theta_3)c(\theta_1)c(\theta_4) + 3.5s(\theta_3)c(\theta_1) \\ 0 \end{bmatrix}$$

$$B =$$

$$\begin{bmatrix} -3s(\theta_1)s(\theta_2)s(\theta_4) + 3s(\theta_1)c(\theta_2)c(\theta_3)c(\theta_4) + 3.5s(\theta_1)c(\theta_2)c(\theta_3) \\ 3s(\theta_2)s(\theta_4)c(\theta_1) - 3c(\theta_1)c(\theta_2)c(\theta_3)c(\theta_4) - 3.5c(\theta_1)c(\theta_2)c(\theta_3) \\ -3s(\theta_2)c(\theta_3)c(\theta_4) - 3.5s(\theta_2)c(\theta_3) - 3s(\theta_4)c(\theta_2) \end{bmatrix}$$

$$C =$$

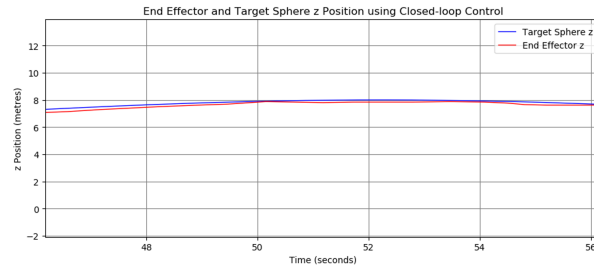
$$\begin{bmatrix} -3s(\theta_1)s(\theta_2)s(\theta_3)c(\theta_4) - 3.5s(\theta_1)s(\theta_2)s(\theta_3) + 3c(\theta_1)c(\theta_3)c(\theta_4) + 3.5c(\theta_1)c(\theta_3) \\ 3s(\theta_1)c(\theta_3)c(\theta_4) + 3.5s(\theta_1)c(\theta_3) + 3s(\theta_2)s(\theta_3)c(\theta_1)c(\theta_4) + 3.5s(\theta_2)s(\theta_3)c(\theta_1) \\ -3s(\theta_3)c(\theta_2)c(\theta_4) - 3.5s(\theta_3)c(\theta_2) \end{bmatrix}$$

$$D =$$

$$\begin{bmatrix} -3s(\theta_1)s(\theta_2)s(\theta_4)c(\theta_3) + 3s(\theta_1)c(\theta_2)c(\theta_4) - 3s(\theta_3)s(\theta_4)c(\theta_1) \\ -3s(\theta_1)s(\theta_3)s(\theta_4) + 3s(\theta_2)s(\theta_4)c(\theta_1)c(\theta_3) - 3c(\theta_1)c(\theta_2)c(\theta_4) \\ -3s(\theta_2)c(\theta_4) - 3s(\theta_4)c(\theta_2)c(\theta_3) \end{bmatrix}$$

Where  $c(\theta_i) = \cos(\theta_i)$ ,  $s(\theta_i) = \sin(\theta_i)$  and A, B, C, D are column vectors that form the Jacobian when arranged like (formatted to save space):

$$\begin{bmatrix} A & B & C & D \end{bmatrix}$$



## 4 Final Task

### 4.2 Null-space Control

*DISCUSS ALGORITHM AND HOW IT IS DIFFERENT FROM PREVIOUS CONTROLLER.*

