

MTRX5700
EXPERIMENTAL ROBOTICS

Assignment 1

Author(s):

KAUSTHUB KRISHNAMURTHY
JAMES FERRIS
SACHITH GUNAWARDHANA

SID:

312086040
311220045
440623630

Due: March 25, 2015

1 Question 1

1.a

The homogeneous transformation matrix for
 $\alpha = 10^\circ$, $\beta = 20^\circ$, $\gamma = 30^\circ$, ${}^A P_B = \{1 \ 2 \ 3\}^T$
looks like:

$$A = \begin{pmatrix} 0.9254 & 0.0180 & 0.3785 & 1.0000 \\ 0.1632 & 0.8826 & -0.4410 & 2.0000 \\ -0.3420 & 0.4698 & 0.8138 & 3.0000 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.b

The homogeneous transformation matrix for
 $\alpha = 10^\circ$, $\beta = 30^\circ$, $\gamma = 30^\circ$, ${}^A P_B = \{3 \ 0 \ 0\}^T$
looks like:

$$B = \begin{pmatrix} 0.8529 & 0.0958 & 0.5133 & 3.0000 \\ 0.1504 & 0.8963 & -0.4172 & 0 \\ -0.5000 & 0.4330 & 0.7500 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.c

The homogeneous transformation matrix for
 $\alpha = 90^\circ$, $\beta = 180^\circ$, $\gamma = -90^\circ$, ${}^A P_B = \{0 \ 0 \ 1\}^T$
looks like:

$$C_1 = \begin{pmatrix} -0 & -0 & -1 & 0 \\ -1 & -0 & 0 & 0 \\ -0 & 1 & -0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In Comparison: The homogeneous transformation matrix for
 $\alpha = 90^\circ$, $\beta = 180^\circ$, $\gamma = 270^\circ$, ${}^A P_B = \{0 \ 0 \ 1\}^T$
looks like:

$$C_2 = \begin{pmatrix} -0 & 0 & -1 & 0 \\ -1 & -0 & 0 & 0 \\ -0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Comparing these two we find that there is no functional difference between the two generated matrices. However we can notice that two of the zeroes are negative in C_1 but are positive in C_2 meaning that the approach to the orientation is different in C_1 and C_2 even if they result in the same thing.

Code Listing

See Appendix A [9.1]

2 Question 2

Refer to the MATLAB code in Appendix 9.a which was used to generate the homogeneous transformation matrices required.

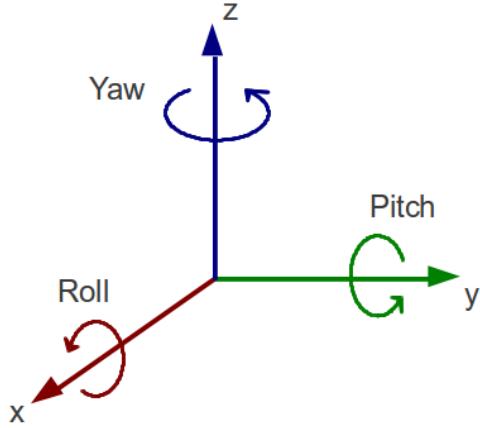


Figure 1: Roll Pitch and Yaw definitions (relative to axis names)

2.a Validity Check

We need to prove that the determinant of the matrix is equal to 1 and that the inverse of the matrix is equal to the transpose to prove validity.

2.a.i R_1

$$R_1 = \begin{pmatrix} 0.7500 & -0.4330 & -0.5000 \\ 0.2165 & 0.8750 & -0.4330 \\ 0.6250 & 0.2165 & 0.7500 \end{pmatrix}$$

$$\det(R_1) = 1.0000$$

$$R_1^T = \begin{pmatrix} 0.7500 & 0.2165 & 0.6250 \\ -0.4330 & 0.8750 & 0.2165 \\ -0.5000 & -0.4330 & 0.7500 \end{pmatrix}$$

$$R_1^{-1} = R_1^T \therefore R_1 \text{ is valid}$$

2.a.ii R_2

$$R_2 = \begin{pmatrix} 0.7725 & -0.4460 & -0.5150 \\ 0.2165 & 0.8750 & -0.4330 \\ 0.6000 & 0.2078 & 0.7200 \end{pmatrix}$$

$$\det(R_2) = 0.9888$$

$$R_2^{-1} = \begin{pmatrix} 0.7281 & 0.2165 & 0.6510 \\ -0.4204 & 0.8750 & 0.2255 \\ -0.4885 & -0.4330 & 0.7813 \end{pmatrix}$$

$$R_2^T = \begin{pmatrix} 0.7725 & 0.2165 & 0.6000 \\ -0.4460 & 0.8750 & 0.2078 \\ -0.5150 & -0.4330 & 0.7200 \end{pmatrix}$$

$R_2^{-1} \cong R_2^T \therefore R_2$ is valid within the limits of practical numerical applications.

2.a.iii R_3

$$R_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0.8660 & 0.5000 & 0 \\ -0.5000 & 0.8660 & 0 \end{pmatrix}$$

$$\det(R_3) = 1$$

$$R_3^{-1} = \begin{pmatrix} 0 & 0.8660 & -0.5000 \\ 0 & 0.5000 & 0.8660 \\ 1 & 0 & 0 \end{pmatrix}$$

$$R_3^T = \begin{pmatrix} 0 & 0.8660 & -0.5000 \\ 0 & 0.5000 & 0.8660 \\ 1 & 0 & 0 \end{pmatrix}$$

$R_3^{-1} = R_3^T \therefore R_3$ is valid

2.a.iv R_4

$$R_4 = \begin{pmatrix} -0.7500 & -0.2165 & -0.6250 \\ 0.4330 & -0.8750 & -0.2165 \\ 0.5000 & 0.4330 & -0.7500 \end{pmatrix}$$

$$\det(R_4) = -1$$

$$R_4^{-1} = \begin{pmatrix} -0.7500 & 0.4330 & 0.5000 \\ -0.2165 & -0.8750 & 0.4330 \\ -0.6250 & -0.2165 & -0.7500 \end{pmatrix}$$

$$R_4^T = \begin{pmatrix} -0.7500 & 0.4330 & 0.5000 \\ -0.2165 & -0.8750 & 0.4330 \\ -0.6250 & -0.2165 & -0.7500 \end{pmatrix}$$

Although $R_4^{-1} = R_4^T$, $\det(R_4) \neq 1 \therefore R_4$ is invalid

2.b Roll/Pitch/Yaw Angles

Roll Angle is α , Pitch Angle is β , Yaw Angle is γ .

Assumption: The believability of the angles can be determined purely mathematically. In reality these angles may not necessarily be believable depending on the application at hand. However, we cannot account for this without further information about the system.

2.b.i R_1

The following values are believable

$$\alpha_1 = 16.1021^\circ$$

$$\beta_1 = -36.6822^\circ$$

$$\gamma_1 = 16.1016^\circ$$

2.b.ii R_2

The following values are believable

$$\alpha_2 = 15.0665^\circ$$

$$\beta_2 = -36.8699^\circ$$

$$\gamma_2 = 15.0552^\circ$$

2.b.iii R_3

The following values are believable

$$\alpha_3 = 90^\circ$$

$$\beta_3 = 30^\circ$$

$$\gamma_3 = 89.5611^\circ$$

2.b.iv R_4

The following values are not believable as R_4 is an invalid rotational matrix

$$\alpha_4 = 150^\circ$$

$$\beta_4 = -30^\circ$$

$$\gamma_4 = 29.9990^\circ$$

2.c Angle Estimation

For a matrix such as R_2 we can adjust our matrix values slightly (making them less accurate i.e. to fewer decimal values) to still get a reasonable estimation of our angles. We can realise this by seeing that the determinant is so close to 1. R_4 , however, cannot give us a reasonable estimate for the angles as the determinant is too far from the required value of 1.

Code Listing

See Appendix A [9.2]

3 Question 3

DH Notation

i	θ	d	r	α
1	θ_1	67	100	$\pi/2$
2	θ_2	0	250	0
3	θ_2	0	0	$\pi/2$
4	θ_4	0	250	$-\pi/2$
5	θ_5	0	0	$-\pi/2$
6	θ_6	0	245	0

Table 1: D-H Variable Notation

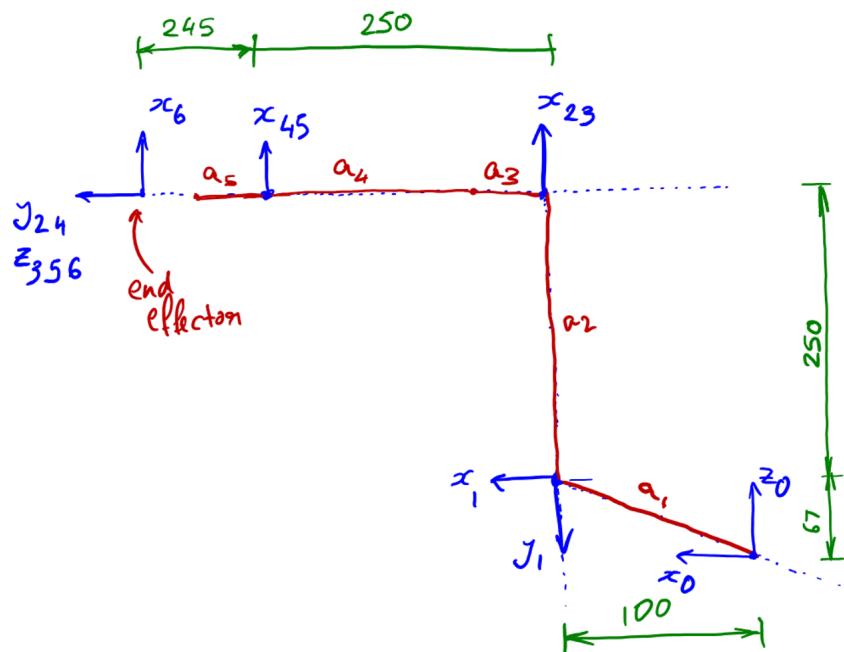


Figure 2: D-H Standard for Forward Kinematics Coordinate Systems

Assumption

Effectively taking the zero coordinate system above joint J1 allows us to eliminate one transformation from the system without drastically altering the kinematic solution behind it. The solution I provide below will be the transformation matrix with respect to my coordinate system 0.

Code Listing

See Appendix A [9.3]

Resulting Transformation Matrix

The resulting transformation matrix can be represented as:

$${}^0\mathbf{T}_6 = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^0\mathbf{A}_1 {}^1\mathbf{A}_2 {}^2\mathbf{A}_3 {}^3\mathbf{A}_4 {}^4\mathbf{A}_5 {}^5\mathbf{A}_6$$

$$\mathbf{n}_x = -\sin \theta_6 (\cos \theta_4 \sin \theta_1 - \sin \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) - \cos \theta_6 (\cos \theta_5 (\sin \theta_1 \sin \theta_4 + \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) + \sin \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2))$$

$$\mathbf{s}_x = \sin \theta_6 (\cos \theta_5 (\sin \theta_1 \sin \theta_4 + \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) + \sin \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2)) - \cos \theta_6 (\cos \theta_4 \sin \theta_1 - \sin \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3))$$

$$\mathbf{a}_x = \sin \theta_5 (\sin \theta_1 \sin \theta_4 + \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) - \cos \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2)$$

$$\mathbf{p}_x = 100 \cos \theta_1 + 250 \cos \theta_1 \cos \theta_2 - 250 \sin \theta_1 \sin \theta_4 - 250 \sin \theta_6 (\cos \theta_4 \sin \theta_1 - \sin \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) - 250 \cos \theta_6 (\cos \theta_5 (\sin \theta_1 \sin \theta_4 + \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) + \sin \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2)) - 250 \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)$$

$$\mathbf{n}_y = \sin \theta_6 (\cos \theta_1 \cos \theta_4 + \sin \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) + \cos \theta_6 (\cos \theta_5 (\cos \theta_1 \sin \theta_4 - \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \sin \theta_5 (\cos \theta_2 \sin \theta_1 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \sin \theta_2))$$

$$\mathbf{s}_y = \cos \theta_6 (\cos \theta_1 \cos \theta_4 + \sin \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \sin \theta_6 (\cos \theta_5 (\cos \theta_1 \sin \theta_4 - \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \sin \theta_5 (\cos \theta_2 \sin \theta_1 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \sin \theta_2))$$

$$\mathbf{a}_y = -\sin \theta_5 (\cos \theta_1 \sin \theta_4 - \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \cos \theta_5 (\cos \theta_2 \sin \theta_1 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \sin \theta_2)$$

$$\mathbf{p}_y = 100 \sin \theta_1 + 250 \cos \theta_2 \sin \theta_1 + 250 \cos \theta_1 \sin \theta_4 + 250 \sin \theta_6 (\cos \theta_1 \cos \theta_4 + \sin \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - 250 \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1) + 250 \cos \theta_6 (\cos \theta_5 (\cos \theta_1 \sin \theta_4 - \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \sin \theta_5 (\cos \theta_2 \sin \theta_1 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \sin \theta_2))$$

$$\mathbf{n}_z = \cos \theta_6 (\sin \theta_5 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) + \cos \theta_4 \cos \theta_5 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)) - \sin \theta_4 \sin \theta_6 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)$$

$$\mathbf{s}_z = -\sin \theta_6 (\sin \theta_5 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) + \cos \theta_4 \cos \theta_5 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)) - \cos \theta_6 \sin \theta_4 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)$$

$$\mathbf{a}_z = \cos \theta_5 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) - \cos \theta_4 \sin \theta_5 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)$$

$$\mathbf{p}_z = 250 \sin \theta_2 + 250 \cos \theta_6 (\sin \theta_5 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) + \cos \theta_4 \cos \theta_5 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)) + 250 \cos \theta_4 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2) - 250 \sin \theta_4 \sin \theta_6 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2) + 67$$

World Coordinate Transformation

If it is necessary to calculate the transformation matrix with respect to the "real" (0, 0, 0) world coordinate system (as defined by the assignment document) we can take the following matrix:

$${}^W \mathbf{A}_0 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 253 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and multiply it with our Transformation Matrix ${}^0 T_6$:

$${}^W T_6 = {}^W A_0 \cdot {}^0 T_6 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 253 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^W T_6 = \begin{pmatrix} -n_y & -s_y & a_y & p_y \\ n_x & s_x & a_x & p_x + 253 \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This can be used for inverse kinematics problems by substituting our end effector position values into ${}^W T_6$ in order to calculate the joint angles. We will see more about this in Question 6.

4 Question 4

4.a Modified DH Notation

i	r	α	d	θ
1	0	0	0	θ_1
2	0	$\pi/2$	0	θ_1
3	10	$\pi/2$	0	θ_1
4	12	$-\pi/2$	0	θ_4
5	0	$\pi/2$	10.5	θ_1

Table 2: Modified D-H Variable Notation

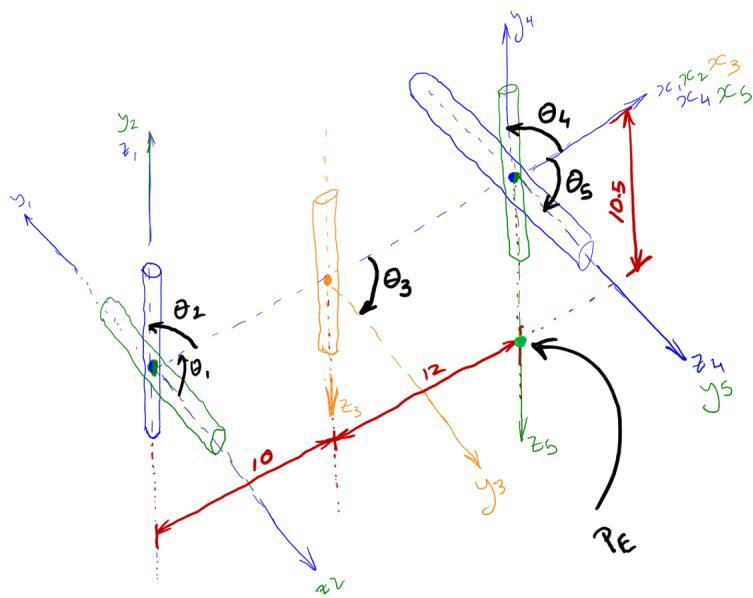


Figure 3: Derived Coordinate Systems

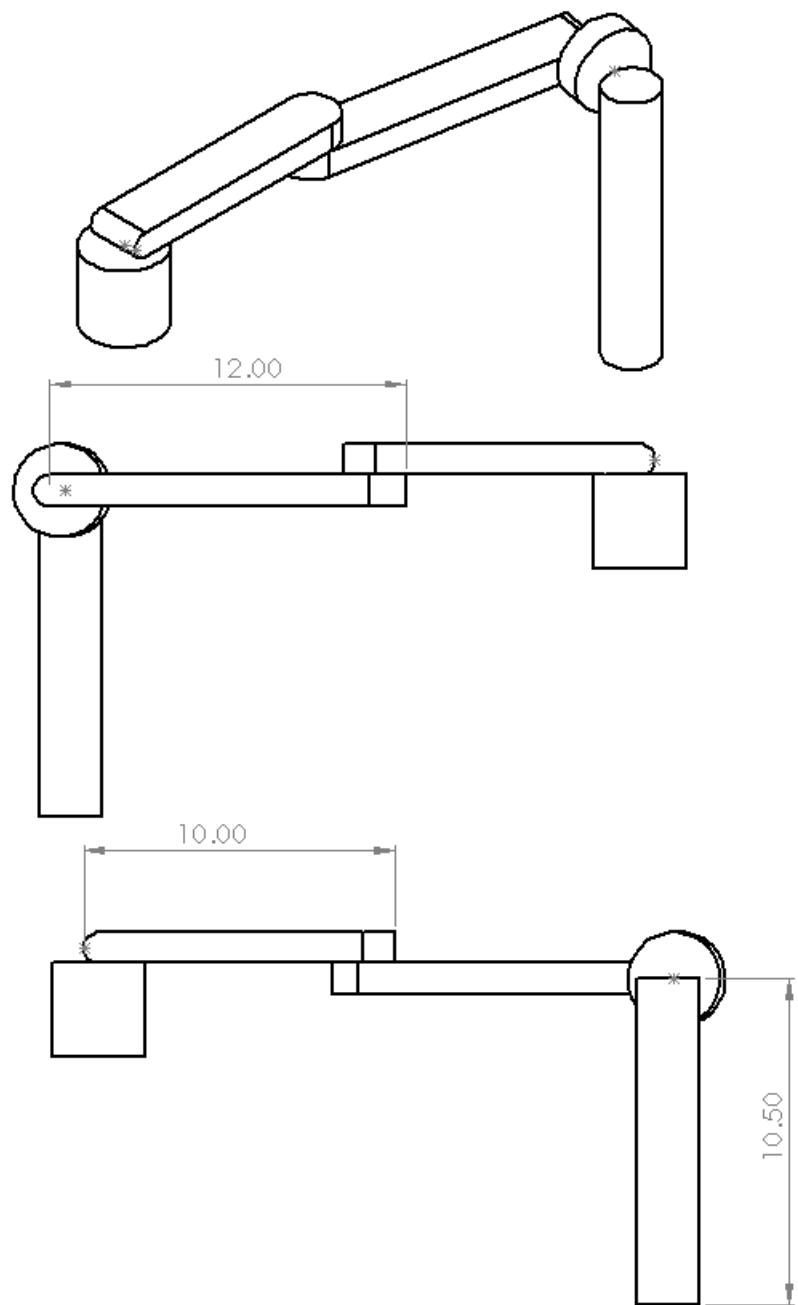


Figure 4: Derived Robot Arm Geometry

5 Question 5

5.a

5.a.i Workspace

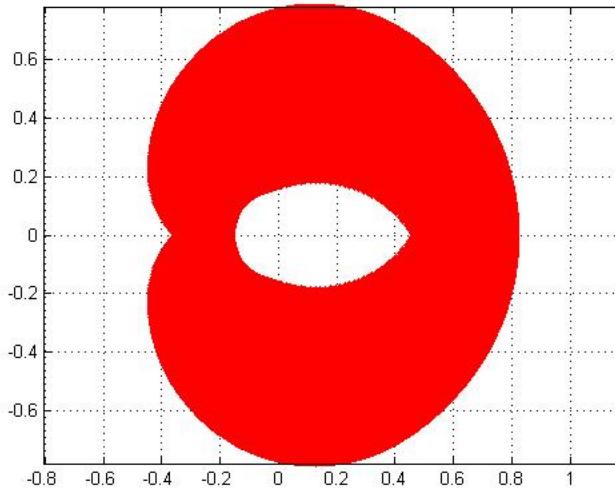


Figure 5: Workspace Plot (See Appendix A [9.4.i])

5.a.ii Configuration Space

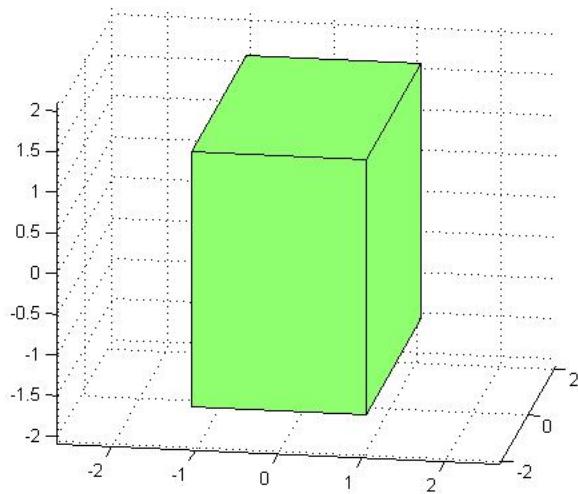


Figure 6: Configuration Space Plot (See Appendix A [9.4.ii])

5.b Singularities

Considering link L1 and link L2, take end point of link L2 as P(x,y)

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

$$x' = (-L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2))\theta' - L_2 \sin(\theta_1 + \theta_2)\theta'_2$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

$$y' = (L_1 \cos \theta_1 - L_2 \cos(\theta_1 + \theta_2))\theta' - L_2 \cos(\theta_1 + \theta_2)\theta'_2$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \cdot \begin{pmatrix} \theta'_1 \\ \theta'_2 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = (J_1) \cdot \begin{pmatrix} \theta'_1 \\ \theta'_2 \end{pmatrix}$$

where $J_1 =$

$$\begin{pmatrix} -L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{pmatrix}$$

$$\det(J_1) = (-L_1 \sin \theta_1 - L_2 \sin(\theta_1 + \theta_2)) \cdot (L_2 \cos(\theta_1 + \theta_2)) + (L_2 \sin(\theta_1 + \theta_2)) \cdot (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2))$$

$$\det(J_1) = L_1 L_2 \sin \theta_3$$

Similarly we can find that

$$\therefore \det(J_1) = L_2 L_3 \sin \theta_3$$

Finding singularities Singularities occurs when $\det(J) = 0$

For J_1 :

$$\det(J_1) = L_2 L_3 \sin \theta_3 = 0$$

\therefore when $\theta_3 = 0$ or π

However, the range of θ_3 must be between $(-\pi/2)$ and $(\pi/2)$

$$\therefore \theta_3 = 0$$

Similarly for J_2 :

$$\det(J_2) = L_1 L_2 \sin \theta_2 = 0$$

\therefore when $\theta_2 = 0$ or π

However, the range of θ_2 must be between $(-2\pi/3)$ and $(2\pi/3)$

$$\therefore \theta_2 = 0$$

So for the whole system singularities occurs according to the intersection of results of A B:
i.e. for all points where $\theta_2 = 0$ and $\theta_3 = 0$

6 Question 6

6.a Inverse Kinematics

6.a.i Theoretical Method

In Question 3 we used forward kinematics to derive the Transformation matrix (0T_6) for this robot arm with respect to our assumed "zero" coordinate system (see Figure 2).

Our aim for this question is to derive the transformation matrix specific to two separate end effector points (P_{e_a} P_{e_b}) which can be done in two ways.

One method (as mentioned in Question 3) could be to apply another transformation matrix wA_0 and then continue simultaneously equating.
newline

According to our method we know that our Z_0 points are based 253 higher than the Z_w points in space and rotated about the Z axis by $\pi/2$. We need to account for this by modifying our end effector point values from:

newline

$$P_{e_a} = (0, 490, 40)$$

$$P_{e_b} = (0, 490, 120)$$

newline

To:

$$P_{e_a} = (0, 490, -213)$$

$$P_{e_b} = (0, 490, -133)$$

Since our end effector must point vertically downwards our $\sum \theta = -\pi/2$ in order to achieve this. We then simultaneously equate this with our equations for P_{e_x} and P_{e_y} with their equivalent values from our transformation matrix 0T_6 . Substituting $\theta_1 = \theta_4 = \theta_6 = 0$ (as we do not rotate around any axis parallel to the Z_0 axis) we can solve for θ_2, θ_3 , and θ_5 .

We do this once to find all the angles for Point A and again for all the angles for point B. We can then compare these angle values with the simulator jkoint angle readings in Q6 Part B.

6.a.ii Results

Our Matlab output looks as follows:

```
sa2 =
-133.57970015587177746184941710562
-56.676681927833899035216072316143

sa3 =
166.90301822803787842663334478947
13.096981771962121573366655210527

sa5 =
-123.32331807216610408729072583881
-46.420299844128225660657381049333

sb2 =
-121.68117590050949105729393864272
-45.154934789134486545751233766901

sb3 =
166.52624111137500451154270487582
13.47375888624995488457295124181

sb5 =
-134.84506521086551657675556438805
-58.318824099490512065212859512228
```

We can interpret this as Points A and B can each have two theoretical values for the angles (i.e. two possible configurations) to reach the desired end effector pose.

Point	θ_2	θ_3	θ_5
P_{e_a} Path 1	-56.677	13.096	-46.420
P_{e_a} Path 2	-133.579	166.903	-123.323
P_{e_b} Path 1	-45.155	13.473	-58.319
P_{e_b} Path 2	-121.681	166.526	-134.845

Table 3: Potential Angle Values for End Effector Positions A and B

6.a.iii Code Listing

See Appendix A [9.5]

6.b Simulation

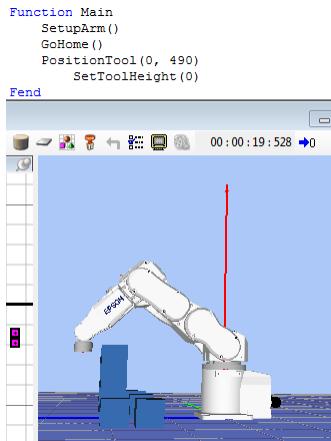


Figure 7: Simulator Point A

Current Position					
J1 (deg)	J2 (deg)	J3 (deg)	J4 (deg)	J5 (deg)	J6 (deg)
0.000	-56.677	13.097	0.000	-46.420	0.000
<input type="radio"/> World			<input checked="" type="radio"/> Joint		
<input type="radio"/> Pulse					

Figure 8: Simulator Point A Angles

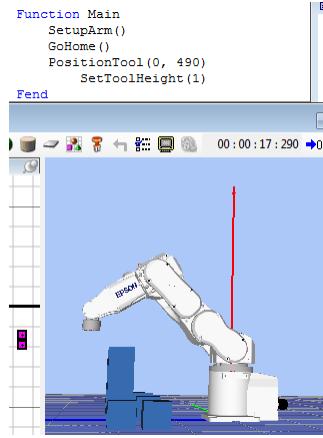


Figure 9: Simulator Point B

Current Position			
J1 (deg)	J2 (deg)	J3 (deg)	<input type="radio"/> World
0.000	-51.349	12.521	<input checked="" type="radio"/> Joint
0.000	-51.172	0.000	<input type="radio"/> Pulse

Figure 10: Simulator Point B Angles

7 Question 7

7.a Qualitative Analysis of Block Stacking Method

The following code describes a general function which can move a block from any location and place it as part of the tower. The function takes as input the x and y coordinates of the block, and the block number. As long as it is known where the block is and how many blocks have already been placed, this function will place the block in its appropriate position of the tower (within limits of the robotic arm, of course).

This function is not optimised - it will not give the tower building time, which requires previous knowledge of the blocks and their locations. This function was designed with the mentality that nothing is previously known about the system - when a block is found, it must be added to the tower, no matter where that block is or how many blocks there are.

7.b CODE

See the Appendix for the Epson robotic arm code.

8 Question 8

8.a

For complete working out see the Appendix.

Brief overview of methodology:

By differentiating the position of link 2 we obtain its velocity. From there we can determine the systems Kinetic Energy.

Potential Energy can be found straight from the effect of gravity.

From there, we can find the Lagrangian $L = K - P$.

We can then determine T_1 and T_2 from the Lagrangian.

Final Results:

After substituting in for m_i , l_i , I_i , we obtain the following:

$$\begin{aligned} T_1 = & \alpha_1[5.1625 + 3 \cos \theta_2] + \alpha_2[0.7625 + 1.5 \cos \theta_2] \\ & -\omega_1 \omega_2 [3 \sin \theta_2] - \omega_2^2 [3 \sin \theta_2] \\ & + 6.5g \cos \theta_1 + 1.5g \cos(\theta_1 + \theta_2) \end{aligned}$$

and

$$\begin{aligned} T_2 = & \alpha_1[0.7625 + 1.5 \cos \theta_2] + \alpha_2[0.2] \\ & + \omega_1^2 [1.5 \sin \theta_2] + 1.5g \cos(\theta_1 + \theta_2) \end{aligned}$$

where

$g = 9.81$ is the acceleration due to gravity.

$\omega_i = \frac{d}{dt} \theta_i$ is the angular velocity.

$\alpha_i = \frac{d}{dt} \omega_i$ is the angular acceleration.

8.b

Rearranging the above equations so that α_i is the subject we obtain the following:

$$\alpha_1 = \frac{T_1 - \alpha_2[0.7625 + 1.5 \cos \theta_2] + \omega_1 \omega_2 [3 \sin \theta_2] + \omega_2^2 [3 \sin \theta_2] - 6.5g \cos(\theta_1 + \theta_2)}{5.1625 + 3 \cos \theta_2}$$

\Rightarrow Equation 1

$$\alpha_2 = \frac{T_2 - \alpha_1[0.7625 + 1.5 \cos \theta_2] - \omega_1^2 [1.5 \sin \theta_2] - 1.5g \cos(\theta_1 + \theta_2)}{0.2}$$

\Rightarrow Equation 2

Through matlab we can simultaneously solve Equations 1 and 2 to obtain a description of α_i without depending on the other α_i .

See the Appendix for Question 8 for the matlab code.

This enables us to utilise the following simulink model:

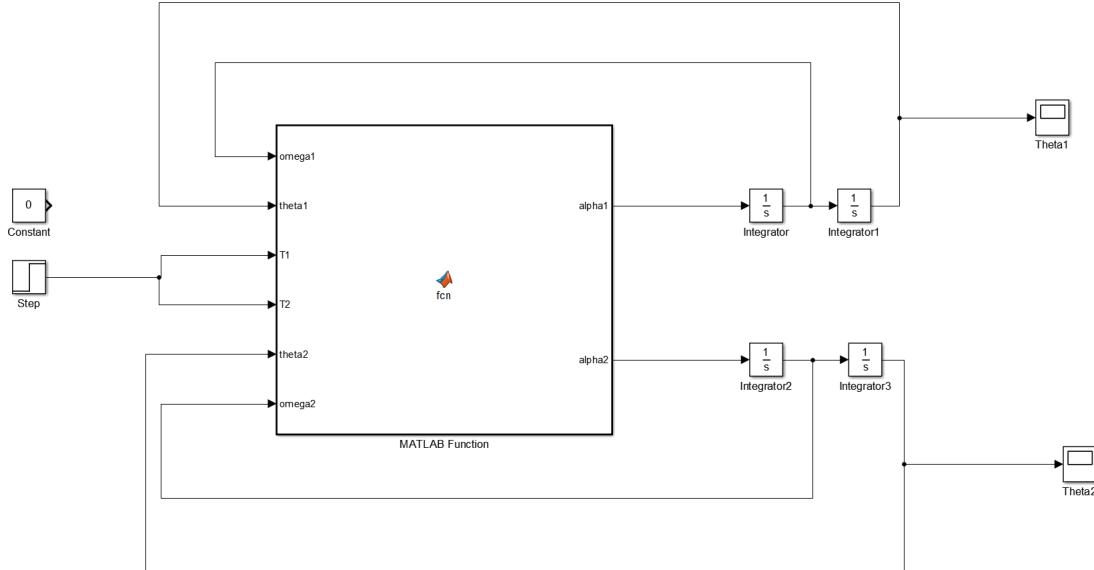


Figure 11: Simulink Model

For a constant zero input torque, we obtain for θ_1 the following output:

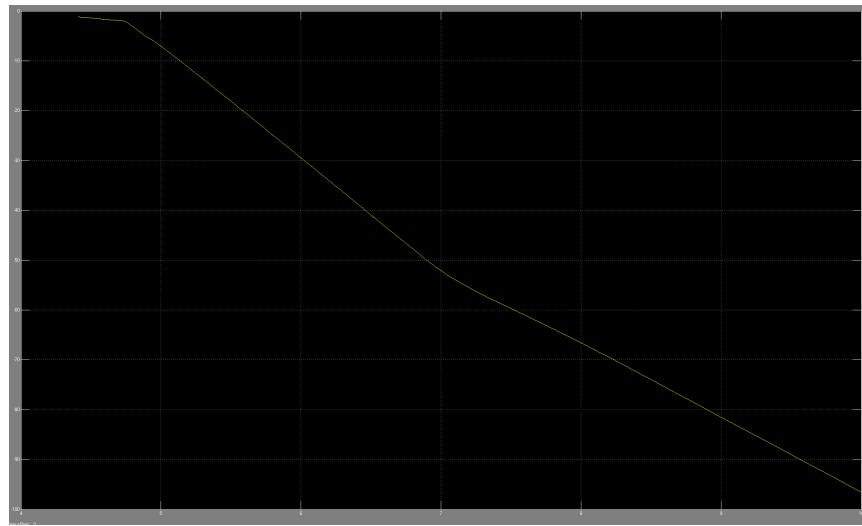


Figure 12: Theta 1 for constant zero torque

and for θ_2 :

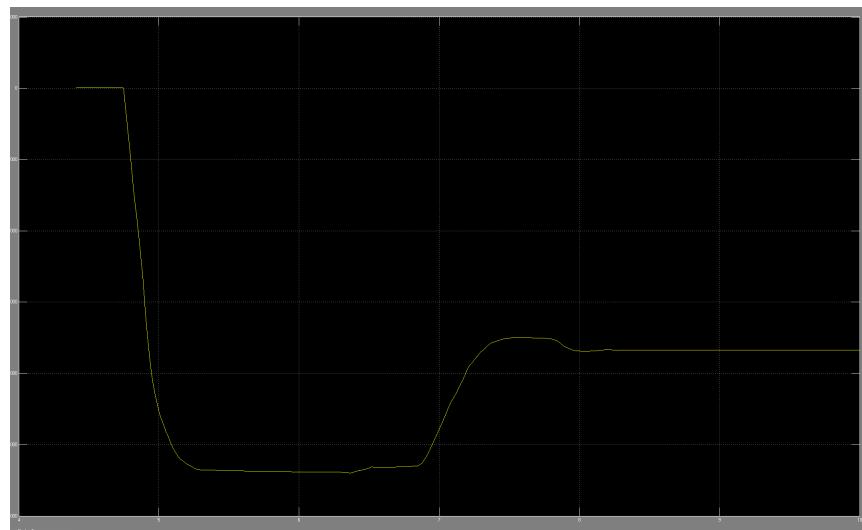


Figure 13: Theta 2 for constant zero torque

For a step input torque, we obtain for θ_1 :

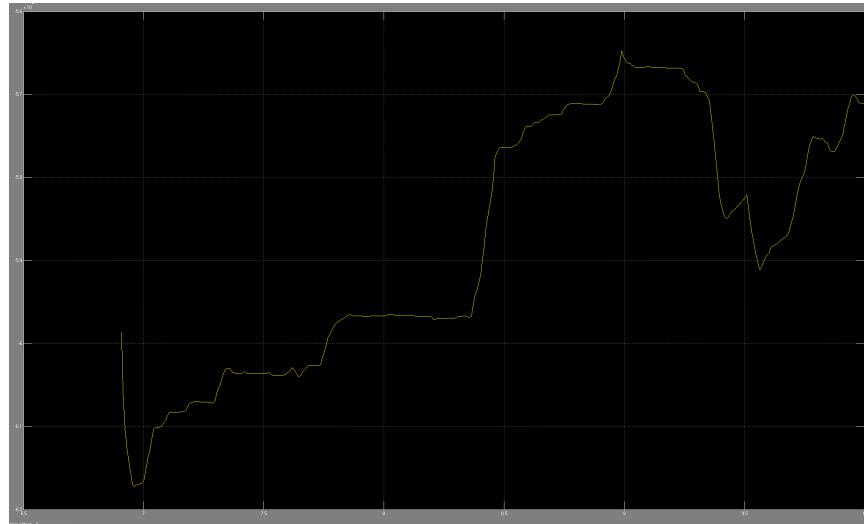


Figure 14: Theta 1 for step torque

and for θ_2 :

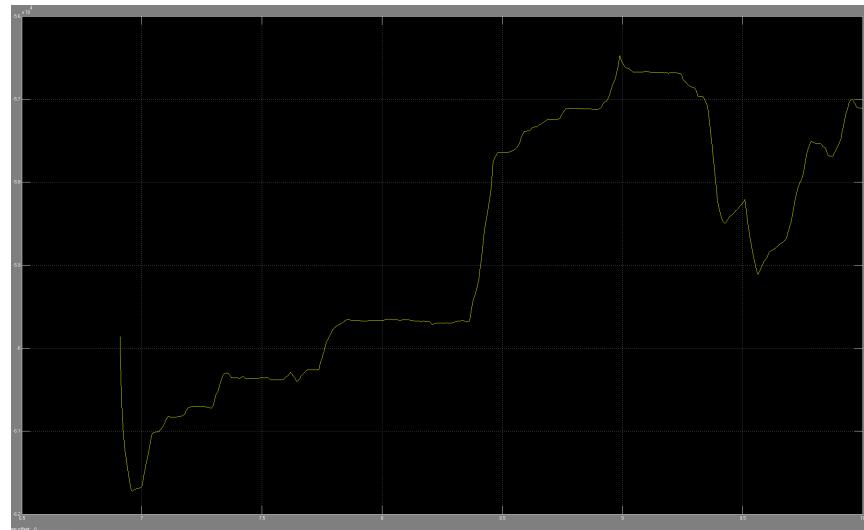


Figure 15: Theta 2 for step torque

8.c

NA

8.d

NA

9 Appendix

9.1 Question 1 Code Listings

```
1 close all
2 clear
3 clc
4
5 DEGREES = pi/180;
6 RADIANS = 180/pi;
7
8 fprintf('a)\n');
9 roll = 10*DEGREES; %alpha
10 pitch = 20*DEGREES; %beta
11 yaw = 30*DEGREES; %gamma
12
13 aPb = [1 2 3];
14 aRb = [cos(roll)*cos(pitch), cos(roll)*sin(pitch)*sin(yaw)-sin(roll)*cos(yaw), cos(roll)*sin(pitch)*
    ↪ cos(yaw)+sin(roll)*sin(yaw);
15     sin(roll)*cos(pitch), sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(yaw), sin(roll)*sin(pitch)*cos
    ↪ (yaw)-cos(roll)*sin(yaw);
16     -sin(pitch), cos(pitch)*sin(yaw), cos(pitch)*cos(yaw)];
17 %transpose(aPb)
18
19 aTb = [aRb transpose(aPb); 0 0 0 1]
20
21 fprintf('b)\n');
22
23 roll = 10*DEGREES; %alpha
24 pitch = 30*DEGREES; %beta
25 yaw = 30*DEGREES; %gamma
26
27 aPb = [3 0 0];
28 aRb = [cos(roll)*cos(pitch), cos(roll)*sin(pitch)*sin(yaw)-sin(roll)*cos(yaw), cos(roll)*sin(pitch)*
    ↪ cos(yaw)+sin(roll)*sin(yaw);
29     sin(roll)*cos(pitch), sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(yaw), sin(roll)*sin(pitch)*cos
    ↪ (yaw)-cos(roll)*sin(yaw);
30     -sin(pitch), cos(pitch)*sin(yaw), cos(pitch)*cos(yaw)];
31 %transpose(aPb)
32
33 aTb = [aRb transpose(aPb); 0 0 0 1]
34
35 fprintf('c)\n');
36
37 roll = 90*DEGREES; %alpha
38 pitch = 180*DEGREES; %beta
39 yaw = -90*DEGREES; %gamma
40
41 aPb = [0 0 1];
42 aRb = [cos(roll)*cos(pitch), cos(roll)*sin(pitch)*sin(yaw)-sin(roll)*cos(yaw), cos(roll)*sin(pitch)*
    ↪ cos(yaw)+sin(roll)*sin(yaw);
43     sin(roll)*cos(pitch), sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(yaw), sin(roll)*sin(pitch)*cos
    ↪ (yaw)-cos(roll)*sin(yaw);
44     -sin(pitch), cos(pitch)*sin(yaw), cos(pitch)*cos(yaw)];
45 %transpose(aPb)
46
47 aTb = [aRb transpose(aPb); 0 0 0 1]
48
49 roll = 90*DEGREES; %alpha
50 pitch = 180*DEGREES; %beta
51 yaw = 270*DEGREES; %gamma
52 aPb = [0 0 1];
53 aRb = [cos(roll)*cos(pitch), cos(roll)*sin(pitch)*sin(yaw)-sin(roll)*cos(yaw), cos(roll)*sin(pitch)*
    ↪ cos(yaw)+sin(roll)*sin(yaw);
54     sin(roll)*cos(pitch), sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(yaw), sin(roll)*sin(pitch)*cos
    ↪ (yaw)-cos(roll)*sin(yaw);
55     -sin(pitch), cos(pitch)*sin(yaw), cos(pitch)*cos(yaw)];
56 %transpose(aPb)
57 aTb = [aRb transpose(aPb); 0 0 0 1]
```

9.2 Question 2 Code Listings

```
1 close all
2 clear
3 clc
4
5 DEGREES = pi/180;
6 RADIANS = 180/pi;
7
8 R1 = [0.7500, -0.4330, -0.5000; 0.2165, 0.8750, -0.4330; 0.6250, 0.2165, 0.7500]
9
10
11 determinantR1 = det(R1)
12 inverseR1 = inv(R1)
13 transposeR1 = transpose(R1)
14
15 betal = asin(-1*R1(3,1));
16 gammal = asin(R1(3,2)/cos(betal));
17 alphal = acos(R1(1,1)/cos(betal));
18
19 alphal = alphal*RADIANS
20 betal = betal*RADIANS
21 gammal = gammal*RADIANS
22
23 R2 = [0.7725, -0.4460, -0.5150; 0.2165, 0.8750, -0.4330; 0.6000, 0.2078, 0.7200]
24 determinantR2 = det(R2)
25 inverseR2 = inv(R2)
26 transposeR2 = transpose(R2)
27
28 beta2 = asin(-1*R2(3,1));
29 gamma2 = asin(R2(3,2)/cos(beta2));
30 alpha2 = acos(R2(1,1)/cos(beta2));
31
32 alpha2 = alpha2*RADIANS
33 beta2 = beta2*RADIANS
34 gamma2 = gamma2*RADIANS
35
36 R3 = [0, 0, 1; 0.8660, 0.500, 0; -0.500, 0.8660, 0]
37 determinantR3 = det(R3)
38 inverseR3 = inv(R3)
39 transposeR3 = transpose(R3)
40
41 beta3 = asin(-1*R3(3,1));
42 gamma3 = asin(R3(3,2)/cos(beta3));
43 alpha3 = acos(R3(1,1)/cos(beta3));
44
45 alpha3 = alpha3*RADIANS
46 beta3 = beta3*RADIANS
47 gamma3 = gamma3*RADIANS
48
49 R4 = [-0.7500, -0.2165, -0.6250; 0.4330, -0.8750, -0.2165; 0.500, 0.4330, -0.7500]
50 determinantR4 = det(R4)
51 inverseR4 = inv(R4)
52 transposeR4 = transpose(R4)
53
54 beta4 = asin(-1*R4(3,1));
55 gamma4 = asin(R4(3,2)/cos(beta4));
56 alpha4 = acos(R4(1,1)/cos(beta4));
57
58 alpha4 = alpha4*RADIANS
59 beta4 = beta4*RADIANS
60 gamma4 = gamma4*RADIANS
```

9.3 Question 3 Code Listings

```

1 close all
2 clear all
3 clc
4
5 %%uninitialised symbolic values
6 th = sym('th',[1 6]);
7 d = sym('d', [1 6]);
8 al = sym('al',[1 6]);
9 r = sym('a',[1 6]);
10
11 %%initialised known values
12 %%d_i
13 d(1) = 67;
14 d(2) = 0;
15 d(3) = 0;
16 d(4) = 0;
17 d(5) = 0;
18 d(6) = 0;
19
20 %%alpha_i
21 al(1) = pi()/2;
22 al(2) = 0;
23 al(3) = -pi()/2;
24 al(4) = pi()/2;
25 al(5) = -pi()/2;
26 al(6) = 0;
27
28 %%r_i
29 r(1) = 100;
30 r(2) = 250;
31 r(3) = 0;
32 r(4) = 250;
33 r(5) = 0;
34 r(6) = 245;
35
36 %%individual link to link transformation matrices
37 A_01 = [cos(th(1)) -sin(th(1))*cos(al(1)) sin(th(1))*sin(al(1)) r(1)*cos(th(1));sin(th(1)) cos(th(1)
   ↪ )*cos(al(1)) -cos(th(1))*sin(al(1)) r(1)*sin(th(1));0 sin(al(1)) cos(al(1)) d(1);0 0 0 1];
38 A_12 = [cos(th(2)) -sin(th(2))*cos(al(2)) sin(th(2))*sin(al(2)) r(2)*cos(th(2));sin(th(2)) cos(th(2)
   ↪ )*cos(al(2)) -cos(th(2))*sin(al(2)) r(2)*sin(th(2));0 sin(al(2)) cos(al(2)) d(2);0 0 0 1];
39 A_23 = [cos(th(3)) -sin(th(3))*cos(al(3)) sin(th(3))*sin(al(3)) r(3)*cos(th(3));sin(th(3)) cos(th(3)
   ↪ )*cos(al(3)) -cos(th(3))*sin(al(3)) r(3)*sin(th(3));0 sin(al(3)) cos(al(3)) d(3);0 0 0 1];
40 A_34 = [cos(th(4)) -sin(th(4))*cos(al(4)) sin(th(4))*sin(al(4)) r(4)*cos(th(4));sin(th(4)) cos(th(4)
   ↪ )*cos(al(4)) -cos(th(4))*sin(al(4)) r(4)*sin(th(4));0 sin(al(4)) cos(al(4)) d(4);0 0 0 1];
41 A_45 = [cos(th(5)) -sin(th(5))*cos(al(5)) sin(th(5))*sin(al(5)) r(5)*cos(th(5));sin(th(5)) cos(th(5)
   ↪ )*cos(al(5)) -cos(th(5))*sin(al(5)) r(5)*sin(th(5));0 sin(al(5)) cos(al(5)) d(5);0 0 0 1];
42 A_56 = [cos(th(6)) -sin(th(6))*cos(al(6)) sin(th(6))*sin(al(6)) r(6)*cos(th(6));sin(th(6)) cos(th(6)
   ↪ )*cos(al(6)) -cos(th(6))*sin(al(6)) r(6)*sin(th(6));0 sin(al(6)) cos(al(6)) d(6);0 0 0 1];
43
44 %%transformation matrix for end effector pose with respect to coordinate
45 %%system zero
46 T_06 = A_01*A_12*A_23*A_34*A_45*A_56

```

9.4 Question 5 Code Listings

9.4.i Workspace Plot

```
1 %%Q5 - Determine the workspace of the system
2 close all;clear;clc
3
4 %Declare constants
5 mm = 10^-3;
6 L1 = 250*mm;
7 L2= 250*mm;
8 L3 = 315*mm;
9
10 %Generate an array of 100 values for each boundary conditions
11 theta1 = linspace(-pi/3,pi/3,100);
12 theta2 = linspace(-2*pi/3, 2*pi/3,100);
13 theta3 = linspace(-pi/2, pi/2,100);
14
15 %Create a grid matrix of the boundary values
16 [THETA1, THETA2, THETA3] = meshgrid(theta1,theta2,theta3);
17
18 %Using the above grid matrix, find X and Y
19 X = L1*cos(THETA1)+L2*cos(THETA1+THETA2)+L3*cos(THETA1+THETA2+THETA3);
20 Y = L1*sin(THETA1)+L2*sin(THETA1+THETA2)+L3*sin(THETA1+THETA2+THETA3);
21
22 %Take the final column of the resulting X and Y matrices
23 %These are the coordinates for a scatter plot
24 plot(X(:,1), Y(:,1), 'r.');
25 axis equal
26 grid
27 xlabel('x');
28 ylabel('y');
29 title('Workspace of the System');
30
31 %http://au.mathworks.com/help/fuzzy/examples/modeling-inverse-kinematics-in-a-robotic-arm.html
```

9.4.ii Configuration Space Plot

```
1 %%Q5 - Determine the Configuration Space of the System
2 close all;clear;clc
3
4 %Declare the boundaries of the angles (theta1, theta2, theta3)
5 theta1 = [-pi/3, pi/3];
6 theta2 = [-2*pi/3, 2*pi/3];
7 theta3 = [-pi/2, pi/2];
8
9 %Send to a function that will generate the graph
10 config.space_3dof(theta1, theta2, theta3);
11 title('Configuration Space of the System');
12 xlabel('-pi/3 < theta1 < pi/3');
13 ylabel('-2pi/3 < theta2 < 2pi/3');
14 zlabel('-pi/2 < theta3 < pi/2');
```

```
1 %%Function to plot the configuration space of a 3DOF system
2 %Creates a rectangular polygon with corners defined by the boundary values
3 %of the system
4
5 %Takes as input 3 arrays with 2 elements in each - these
6 %are the boundaries of the x, y, and z values respectively
7 function config.space_3dof(X,Y,Z)
8
9     hold on
10    grid
11    axis equal
```

```

12
13 %Generate one face of the rectangular polygon, with corners described
14 %by the system boundaries.
15 cornersX = [X(2), X(2), X(1), X(1)];
16 cornersY = [Y(2), Y(1), Y(1), Y(2)];
17 cornersZ = [Z(1), Z(1), Z(1), Z(1)];
18 fill3(cornersX, cornersY, cornersZ,1);
19
20 %Generate the rest of the faces, building the polygon
21 cornersX = [X(2), X(2), X(1), X(1)];
22 cornersY = [Y(2), Y(1), Y(1), Y(2)];
23 cornersZ = [Z(2), Z(2), Z(2), Z(2)];
24 fill3(cornersX, cornersY, cornersZ,1);
25
26 cornersX = [X(2), X(2), X(2), X(2)];
27 cornersY = [Y(2), Y(1), Y(1), Y(2)];
28 cornersZ = [Z(1), Z(1), Z(2), Z(2)];
29 fill3(cornersX, cornersY, cornersZ,1);
30
31 cornersX = [X(1), X(1), X(1), X(1)];
32 cornersY = [Y(2), Y(1), Y(1), Y(2)];
33 cornersZ = [Z(1), Z(1), Z(2), Z(2)];
34 fill3(cornersX, cornersY, cornersZ,1);
35
36 cornersX = [X(1), X(2), X(2), X(1)];
37 cornersY = [Y(2), Y(2), Y(2), Y(2)];
38 cornersZ = [Z(1), Z(1), Z(2), Z(2)];
39 fill3(cornersX, cornersY, cornersZ,1);
40
41 cornersX = [X(1), X(2), X(2), X(1)];
42 cornersY = [Y(1), Y(1), Y(1), Y(1)];
43 cornersZ = [Z(1), Z(1), Z(2), Z(2)];
44 fill3(cornersX, cornersY, cornersZ,1);
45
46 hold off
47 end

```

9.5 Question 6 Code Listings

```

1 close all
2 clear all
3 clc
4
5 %uninitialised symbolic values
6 th = sym('th',[1 6]);
7 d = sym('d', [1 6]);
8 al = sym('al',[1 6]);
9 r = sym('a',[1 6]);
10
11 %%initialised known values
12 %%initializing thetas 1, 4, and 6 as we know the robot cannot turn about
13 %%those angles for the specified end effector position
14 th(1) = 0;
15 %th(2) = 0;
16 % th(3) = 0;
17 th(4) = 0;
18 % th(5) = 0;
19 th(6) = 0;
20
21 d(1) = 67;
22 d(2) = 0;
23 d(3) = 0;
24 d(4) = 0;
25 d(5) = 0;
26 d(6) = 0;
27
28 al(1) = pi()/2;
29 al(2) = 0;
30 al(3) = -pi()/2;
31 al(4) = pi()/2;
32 al(5) = -pi()/2;
33 al(6) = 0;
34
35 r(1) = 100;
36 r(2) = 250;
37 r(3) = 0;
38 r(4) = 250;
39 r(5) = 0;
40 r(6) = 245;
41
42 %%individual link to link transformation matrices
43
44 A_01 = [cos(th(1)) -sin(th(1))*cos(al(1)) sin(th(1))*sin(al(1)) r(1)*cos(th(1));sin(th(1)) cos(th(1)
45 %           ↪)*cos(al(1)) -cos(th(1))*sin(al(1)) r(1)*sin(th(1));0 sin(al(1)) cos(al(1)) d(1);0 0 0 1];
46 A_12 = [cos(th(2)) -sin(th(2))*cos(al(2)) sin(th(2))*sin(al(2)) r(2)*cos(th(2));sin(th(2)) cos(th(2)
47 %           ↪)*cos(al(2)) -cos(th(2))*sin(al(2)) r(2)*sin(th(2));0 sin(al(2)) cos(al(2)) d(2);0 0 0 1];
48 A_23 = [cos(th(3)) -sin(th(3))*cos(al(3)) sin(th(3))*sin(al(3)) r(3)*cos(th(3));sin(th(3)) cos(th(3)
49 %           ↪)*cos(al(3)) -cos(th(3))*sin(al(3)) r(3)*sin(th(3));0 sin(al(3)) cos(al(3)) d(3);0 0 0 1];
50 A_34 = [cos(th(4)) -sin(th(4))*cos(al(4)) sin(th(4))*sin(al(4)) r(4)*cos(th(4));sin(th(4)) cos(th(4)
51 %           ↪)*cos(al(4)) -cos(th(4))*sin(al(4)) r(4)*sin(th(4));0 sin(al(4)) cos(al(4)) d(4);0 0 0 1];
52 A_45 = [cos(th(5)) -sin(th(5))*cos(al(5)) sin(th(5))*sin(al(5)) r(5)*cos(th(5));sin(th(5)) cos(th(5)
53 %           ↪)*cos(al(5)) -cos(th(5))*sin(al(5)) r(5)*sin(th(5));0 sin(al(5)) cos(al(5)) d(5);0 0 0 1];
54 A_56 = [cos(th(6)) -sin(th(6))*cos(al(6)) sin(th(6))*sin(al(6)) r(6)*cos(th(6));sin(th(6)) cos(th(6)
55 %           ↪)*cos(al(6)) -cos(th(6))*sin(al(6)) r(6)*sin(th(6));0 sin(al(6)) cos(al(6)) d(6);0 0 0 1];
56
57 %%transformation matrix for end effector pose with respect to coordinate
58 %%system zero
59
60 T_06 = A_01*A_12*A_23*A_34*A_45*A_56
61
62 % T11 = T_06(1, 1);
63 % T12 = T_06(1, 2);
64 % T13 = T_06(1, 3);
65 % T14 = T_06(1, 4);
66 % T21 = T_06(2, 1);
67 % T22 = T_06(2, 2);

```

```

62 % T23 = T_06(2, 3);
63 % T24 = T_06(2, 4);
64 % T31 = T_06(3, 1);
65 % T32 = T_06(3, 2);
66 % T33 = T_06(3, 3);
67 % T34 = T_06(3, 4);
68 % T41 = T_06(4, 1);
69 % T42 = T_06(4, 2);
70 % T43 = T_06(4, 3);
71 % T44 = T_06(4, 4);
72
73 %%rectifies against unrealistic solutions.
74 %%included this line prior to running the solve function
75 T = vpa(T_06);
76
77 %%implement the solve function
78 %%solve simultaneously
79 %%Theta2 + 3 + 4 = -90 degrees %%-->Equation 1
80 %%Px = 490 %%-->Equation 2
81 %%Pz = -213 %%-->Equation 3
82 %%Solve for th2, 3, 5 values for block 1 (designated by a) and block 2
83 %%(designated by b)
84 sa = solve(th(2) + th(3) + th(5) == -pi/2, T_06(1,4) == 490, T_06(3,4) == -213, th(2),th(3),th(5));
85 sb = solve(th(2) + th(3) + th(5) == -pi/2, T_06(1,4) == 490, T_06(3,4) == -133, th(2),th(3),th(5));
86 %%convert to degrees and correct theta
87 sa2 = vpa(radtodeg(sa.th2)) - 90
88 sa3 = vpa(radtodeg(sa.th3)) + 90
89 sa5 = vpa(radtodeg(sa.th5))
90 sb2 = vpa(radtodeg(sb.th2)) - 90
91 sb3 = vpa(radtodeg(sb.th3)) + 90
92 sb5 = vpa(radtodeg(sb.th5))

```

9.6 Question 7 Code Listings

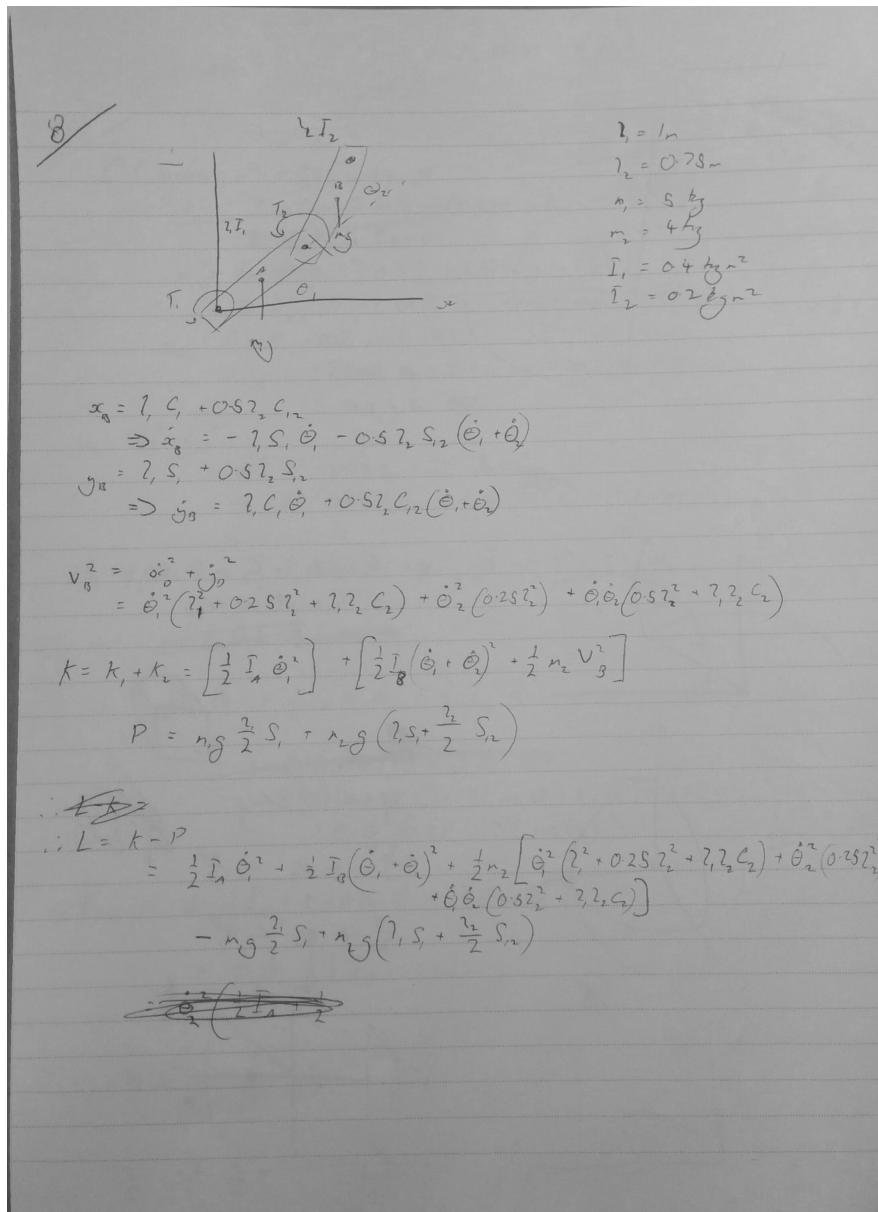
9.6.i Epson Arm Code

```
1 .....  
2 ' Wrapper Functions...'  
3 ' Don 't change anything in this section...'  
4 ..  
5 #define MoveBlockHeight (40)  
6 #define BlockWidth (80)  
7 Function SetupArm()  
8     Motor On  
9     If (0) Then  
10        Power High  
11        SpeedS 2000 ' 80  
12        AccelS 500 ' 50  
13        Accl 50, 50 ' 15, 15  
14        Speed 15  
15        SpeedR 75  
16        AccelR 200  
17    Else  
18        Power Low  
19        SpeedS 80  
20        AccelS 50  
21        Accl 15, 15  
22        Speed 3  
23    EndIf  
24    TLSet 1, XY(0, 0, 180, 0, 0, 0)  
25    Tool 1  
26 Fend  
27 Function GoHome()  
28     Move LJM(Here :Z(MoveBlockHeight + BlockWidth))  
29     Move LJM(Here :U(90) :V(0) :W(180)) ROT  
30     Move LJM(XY(0, 360, MoveBlockHeight + BlockWidth, 90, 0, 180))  
31 Fend  
32 Function CloseGripper  
33     On 10  
34     Wait 0.2  
35 Fend  
36 Function OpenGripper  
37     Off 10  
38     Wait 0.2  
39 Fend  
40 Function SetToolHeight(Height As Real)  
41     If (Height < 0) Then  
42         Height = 0  
43     EndIf  
44     'Go Here :Z(MoveBlockHeight + Height * BlockWidth) LJM  
45     Move LJM(Here :Z(MoveBlockHeight + Height * BlockWidth))  
46 Fend  
47 Function PositionTool(XPos As Real, YPos As Real)  
48     P1 = Here  
49     Real z  
50     z = CZ(P1)  
51     'Go LJM(Here :X(XPos) :Y(YPos))  
52     Move LJM(Here :X(XPos) :Y(YPos))  
53 Fend  
54 Function SetToolAngle(Angle As Real)  
55     If (Angle < 0) Then  
56         Angle = 0  
57     EndIf  
58     If (Angle > 180) Then  
59         Angle = 180  
60     EndIf  
61     Move LJM(Here :U(90 + Angle) :V(0) :W(180)) ROT  
62 Fend  
63 ' ... end wrapper functions '  
64 ..
```

```
66 Function BuildBlock(x As Real, y As Real, BlockNumber As Integer)
67     SetToolHeight(4)
68     PositionTool(x, y)
69     SetToolHeight(0)
70     CloseGripper
71     SetToolHeight(4 + 0.1)
72     PositionTool(0, 490)
73     SetToolHeight((BlockNumber - 1))
74     OpenGripper
75 Fend
76
77 Function main
78     SetupArm()
79     GoHome
80     BuildBlock(0, 290, 1)
81     BuildBlock(300, 290, 2)
82     BuildBlock(300, 390, 3)
83     BuildBlock(-300, 390, 4)
84     BuildBlock(-300, 290, 5)
85
86 Fend
```

9.7 Question 8 Working and Code Listings

9.7.i Part a Working



$$\begin{aligned}
T_i &= \frac{\partial}{\partial \dot{\theta}_i} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) = \frac{\partial L}{\partial \ddot{\theta}_i} \quad F_i = \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) = -\frac{\partial L}{\partial \dot{\theta}_i} \\
k &= \frac{1}{2} I_{\text{ext}} \dot{\theta}_i^2 + \left[\frac{1}{2} I_a (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} J_a V_o^2 \right] \\
&\quad + \frac{1}{2} I_a \dot{\theta}_1^2 + \frac{1}{2} I_a \dot{\theta}_2^2 + I_a \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} J_a \dot{\theta}_2^2 \\
&\quad + \frac{1}{2} m_2 \left[\dot{\theta}_1^2 (I_1^2 + 0.25 I_2^2 + 7.7 C_2) + \dot{\theta}_2^2 (0.25 I_1^2) \right. \\
&\quad \left. + \dot{\theta}_1 \dot{\theta}_2 (0.5 I_1^2 + 7.7 C_2) \right] \\
&= \dot{\theta}_1^2 \left(\frac{1}{2} I_{\text{ext}} + \frac{1}{2} I_a \right) + \dot{\theta}_1 \dot{\theta}_2 (I_a) + \dot{\theta}_2^2 \left(\frac{1}{2} I_a \right) \\
&\quad + \dot{\theta}_1^2 \left[\frac{1}{2} I_{\text{ext}} + \frac{1}{2} I_a + \frac{1}{2} m_2 (I_1^2 + 0.25 I_2^2 + 7.7 C_2) \right] \\
&\quad + \dot{\theta}_1 \dot{\theta}_2 \left[I_a + \frac{1}{2} m_2 (0.5 I_1^2 + 7.7 C_2) \right] \\
&\quad + \dot{\theta}_2^2 \left[\frac{1}{2} I_a \right] \\
P &= m_1 g \frac{I_1}{2} S_1 + m_2 g \left(I_1 S_1 + \frac{I_1}{2} S_{12} \right) \\
L &= T - P \\
&= \dot{\theta}_1^2 \left[\frac{1}{2} I_{\text{ext}} + \frac{1}{2} I_a + \frac{1}{2} m_2 (I_1^2 + 0.25 I_2^2 + 7.7 C_2) \right] \\
&\quad + \dot{\theta}_1 \dot{\theta}_2 \left[I_a + \frac{1}{2} m_2 (0.5 I_1^2 + 7.7 C_2) \right] \\
&\quad + \dot{\theta}_2^2 \left[\frac{1}{2} I_a \right] - m_1 g \frac{I_1}{2} S_1 - m_2 g \left(I_1 S_1 + \frac{I_1}{2} S_{12} \right)
\end{aligned}$$

$$T_i = \frac{\partial}{\partial \dot{\theta}_i} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = -m_1 g \frac{\gamma_1}{2} C_1 - m_2 g \frac{\gamma_1}{2} C_1 - m_2 g \frac{\gamma_2}{2} C_{12}$$

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_2} &= \ddot{\theta}_1 [I_1 + I_2 + m_2 (\gamma_1^2 + 0.25 \gamma_2^2 + \gamma_1 \gamma_2 C_2)] \\ &\quad + \ddot{\theta}_2 [I_3 + \frac{1}{2} m_2 (0.5 \gamma_1^2 + \gamma_1 \gamma_2 C_2)] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \dot{\theta}_1} \frac{\partial L}{\partial \dot{\theta}_2} &= \ddot{\theta}_1 [I_1 + I_2 + m_2 (\gamma_1^2 + 0.25 \gamma_2^2 + \gamma_1 \gamma_2 C_2)] \\ &\quad - m_2 \gamma_1 \gamma_2 S_2 \dot{\theta}_1 \dot{\theta}_2 \\ &\quad + \ddot{\theta}_2 [I_3 + \frac{1}{2} m_2 (0.5 \gamma_1^2 + \gamma_1 \gamma_2 C_2)] \\ &\quad - m_2 \gamma_1 \gamma_2 S_2 \dot{\theta}_2^2 \end{aligned}$$

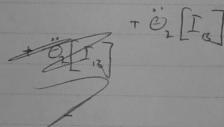
$$\begin{aligned} T_1 &= \ddot{\theta}_1 [I_1 + I_3 + m_2 (\gamma_1^2 + 0.25 \gamma_2^2 + \gamma_1 \gamma_2 C_2)] \\ &\quad + \ddot{\theta}_2 [I_3 + \frac{1}{2} m_2 (0.5 \gamma_1^2 + \gamma_1 \gamma_2 C_2)] \\ &\quad - m_2 \gamma_1 \gamma_2 S_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 \gamma_1 \gamma_2 S_2 \dot{\theta}_2^2 \\ &\quad + m_2 g \frac{\gamma_1}{2} C_1 + m_2 g \gamma_1 C_1 + m_2 g \frac{\gamma_2}{2} C_{12} \end{aligned}$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = -\frac{1}{2} m_2 \dot{\gamma}_2 \sin \dot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_2} = -\frac{1}{2} m_2 \dot{\gamma}_2 \sin \dot{\theta}_2 + \frac{1}{2} m_2 \dot{\gamma}_2 \sin \dot{\theta}_2 = -m_2 \dot{\gamma}_2 \sin \dot{\theta}_2$$

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} &= \ddot{\theta}_1 [I_1 + \frac{1}{2} m_2 (\cos \theta_1^2 + \dot{\gamma}_1 \dot{\gamma}_2)] \\ &\quad + \dot{\theta}_2 [I_2] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{\theta}_1} &= \ddot{\theta}_1 [I_1 + \frac{1}{2} m_2 (\cos \theta_1^2 + \dot{\gamma}_1 \dot{\gamma}_2)] \\ &\quad - \frac{1}{2} m_2 \dot{\gamma}_1 \dot{\gamma}_2 \sin \dot{\theta}_2 \dot{\theta}_2 \end{aligned}$$



$$\begin{aligned} T_1 &= \ddot{\theta}_1 [I_1 + \frac{1}{2} m_2 (\cos \theta_1^2 + \dot{\gamma}_1 \dot{\gamma}_2)] + \dot{\theta}_2 [I_2] \\ &\quad - \frac{1}{2} m_2 \dot{\gamma}_1 \dot{\gamma}_2 \sin \dot{\theta}_2 \dot{\theta}_2 \\ &\quad + \frac{1}{2} m_2 \dot{\gamma}_1 \dot{\gamma}_2 \sin \dot{\theta}_2 \dot{\theta}_2 + m_2 \dot{\gamma}_2 \frac{\dot{\gamma}_1}{2} C_2 \end{aligned}$$

$$T_1 = \ddot{\theta}_1 [I_A + I_B + m_2 (r_1^2 + 0.25 r_2^2 + 2r_1 C_2)]$$

$$+ \ddot{\theta}_2 [I_B + \frac{1}{2} m_2 (0.5 r_2^2 + 2r_1 C_2)]$$

$$+ -m_2 r_1 r_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 r_1 \dot{C}_2 \dot{\theta}_2^2$$

$$+ mg \frac{r_1}{2} C_1 + m_2 g r_1 C_2 - m_2 g \frac{r_2}{2} C_2$$

$$T_2 = \ddot{\theta}_1 [I_A + \frac{1}{2} m_2 (0.5 r_2^2 + 2r_1 C_2)] + \ddot{\theta}_2 [I_B]$$

$$- \frac{1}{2} m_2 r_1 r_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_2 r_1 r_2 \dot{C}_2 \dot{\theta}_2^2$$

$$+ \frac{1}{2} m_2 r_1 r_2 \dot{C}_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 g \frac{r_2}{2} C_2$$

$$\therefore T_1 = \ddot{\theta}_1 [5.1625 + 3 \cos \theta_2] + \ddot{\theta}_2 [0.7625 + 1.5 \cos \theta_2]$$

$$- \dot{\theta}_1 \dot{\theta}_2 [3 \sin \theta_2] - \dot{\theta}_2^2 [3 \sin \theta_2]$$

$$+ 6.5g \cos \theta_1 + 1.5g \cos(\theta_1 + \theta_2)$$

$$\therefore T_2 = \ddot{\theta}_1 [0.7625 + 1.5 \cos \theta_2] + \ddot{\theta}_2 [0.2]$$

$$- \cancel{\dot{\theta}_1 \dot{\theta}_2 [3 \sin \theta_2]} + \dot{\theta}_1^2 [1.5 \sin \theta_2] + 1.5g \cos(\theta_1 + \theta_2)$$

where $g = \text{acceleration due to gravity}$
 $\approx 9.81 \text{ m/s}^2$

8b

$$T_1 = \ddot{\theta}_1 [8.1625 + 3\cos\theta_2] + \ddot{\theta}_2 [0.7625 + 1.5\cos\theta_2]$$

$$\rightarrow -\dot{\theta}_1 \dot{\theta}_2 [3\sin\theta_2] - \dot{\theta}_2^2 [3\sin\theta_2]$$

$$+ 6 \cdot 8g \cos\theta_1 + 1 \cdot 8g \cos(\theta_1 + \theta_2)$$

$$\therefore \ddot{\theta}_1 = \frac{\left(T_1 - \ddot{\theta}_2 [0.7625 + 1.5\cos\theta_2] + \dot{\theta}_1 \dot{\theta}_2 [3\sin\theta_2] \right.}{8.1625 + 3\cos\theta_2} \\ \left. + \dot{\theta}_2^2 [3\sin\theta_2] - 6 \cdot 8g \cos\theta_1 + 1 \cdot 8g \cos(\theta_1 + \theta_2) \right)$$

→ ①

$$T_2 = \dot{\theta}_1 [0.7625 + 1.5\cos\theta_2] + \ddot{\theta}_2 [0.2]$$

$$+ \dot{\theta}_1^2 [1.5\sin\theta_2] + 1.5g \cos(\theta_1 + \theta_2)$$

$$\therefore \ddot{\theta}_2 = \frac{\left(T_2 - \dot{\theta}_1 [0.7625 + 1.5\cos\theta_2] - \dot{\theta}_1^2 [1.5\sin\theta_2] \right.}{0.2} \\ \left. - 1.5g \cos(\theta_1 + \theta_2) \right)$$

→ ②

9.7.ii Part b Code

```

1 %%Code to simultaneously solve for Angular acceleration 1 and 2, without reference to the other
2
3 close all;clear;clc;
4
5 % Torque 1&2
6 syms T1
7 syms T2
8 % Angles 1&2
9 syms theta1
10 syms theta2
11 % Angular Velocity 1&2
12 syms omega1
13 syms omega2
14 % Angular Acceleration 1&2
15 syms alpha1
16 syms alpha2
17
18 % Acceleration due to Gravity
19 g = 9.8;
20
21 % alpha1 = (T1 - alpha2*(0.7625 + 1.5*cos(theta2)) + omega1*omega2*(3*sin(theta2)) + omega2*3*sin(
22 %     ↪ theta2) - 6.5*g*cos(theta1) + 1.5*g*cos(theta1 + theta2))/(5.1625 + 3*cos(theta2));
23 % alpha2 = (T2 - alpha1*(0.7625 + 1.5*cos(theta2) - omega1^2*(1.5*sin(theta2))) - 1.5*g*cos(theta1 +
24 %     ↪ theta2))/0.2;
25 s = solve(alpha1 == (T1 - alpha2*(0.7625 + 1.5*cos(theta2)) + omega1*omega2*(3*sin(theta2)) + omega2*
26 %     ↪ *3*sin(theta2) - 6.5*g*cos(theta1) + 1.5*g*cos(theta1 + theta2))/(5.1625 + 3*cos(theta2)),
27 %     alpha2 == (T2 - alpha1*(0.7625 + 1.5*cos(theta2) - omega1^2*(1.5*sin(theta2))) - 1.5*g*cos(
28 %         ↪ theta1 + theta2))/0.2 ,alpha1,alpha2);
29
30 alpha1 = s.alpha1
31 alpha2 = s.alpha2
32
33 %% Function code for simulink - uses the equations found above to determine angular acceleration
34
35 % function [alpha1,alpha2] = fcn(omega1,theta1,T1,T2,theta2,omega2)
36 %#codegen
37
38 % alpha1 = (2*(3200*T1 - 12200*T2 + 226611*cos(theta1 + theta2) - 204048*cos(theta1) - 24000*T2*cos(
39 %     ↪ theta2) + 9600*omega2*sin(theta2) + 353160*cos(theta1 + theta2)*cos(theta2) + 9600*omega1*
40 %     ↪ omega2*sin(theta2)))/(5*(7320*omega1^2*sin(theta2) - 10800*cos(theta2) - 14400*cos(theta2)^2
41 %     ↪ + 14400*omega1^2*cos(theta2)*sin(theta2) + 2887));
42
43 % alpha2 = -(2*(12200*T1 - 82600*T2 + 1394982*cos(theta1 + theta2) - 777933*cos(theta1) - 1530360*
44 %     ↪ cos(theta1)*cos(theta2) + 24000*T1*cos(theta2) - 48000*T2*cos(theta2) + 36600*omega2*sin(
45 %     ↪ theta2) + 1059480*cos(theta1 + theta2)*cos(theta2) + 36600*omega1*omega2*sin(theta2) - 24000*
46 %     ↪ T1*omega1^2*sin(theta2) + 72000*omega2*cos(theta2)*sin(theta2) - 353160*omega1^2*cos(theta1 +
47 %     ↪ theta2)*sin(theta2) - 72000*omega1^2*omega2*sin(theta2)^2 - 72000*omega1^3*omega2*sin(theta2
48 %     ↪ )^2 + 1530360*omega1^2*cos(theta1)*sin(theta2) + 72000*omega1*omega2*cos(theta2)*sin(theta2))
49 %     ↪ )/(5*(7320*omega1^2*sin(theta2) - 10800*cos(theta2) - 14400*cos(theta2)^2 + 14400*omega1^2*
50 %     ↪ cos(theta2)*sin(theta2) + 2887));

```